POWER IMBALANCE PREDICTION IN TURKISH ENERGY MARKET

by
HASAN DEMIRTAŞ

Submitted to the Sabancı Graduate Business School
in partial fulfilment of
the requirements for the degree of Master of Science in Business Analytics

Sabancı University
Aug 2020

POWER IMBALANCE PREDICTION IN TURKISH ENERGY MARKET

Approved by:

Prof. Can Akkan ....
(Thesis Supervisor)

Assoc. Prof. Abdullah Daşcı ....

Assoc. Prof. Enes Eryarsoy ....

Date of Approval: Sep 7, 2020

HASAN DEMİRTAŞ 2020 ©

All Rights Reserved

ABSTRACT

POWER IMBALANCE PREDICTION IN TURKISH ENERGY MARKET

HASAN DEMIRTAŞ

BUSINESS ANALYTICS M.A. THESIS, SEP 2020

Thesis Supervisor: Prof. CAN AKKAN

Keywords: electricity load imbalance prediction, intraday market, balancing power market, predictive analytics, Turkish energy market, energy trade

There are potential trading opportunities in predicting energy imbalance in energy markets. The energy imbalance in this study is the hourly energy difference between the final planned production and the real-time consumption at the energy delivery hour. We name it as net loading. From the perspective of an energy trade company (TradeCo), being able to predict the net loading can help to make profitable trades in the intraday market (IM). From the perspective of a generation company (GenCo), being able to predict the net loading can help to optimize price offers it gives to TSO in the balancing power market (BPM). Therefore, being able to predict net loading can provide a competitive edge in the energy market. In this study, net loading is tried to be numerically predicted for (T+1) up to (T+32) hours where T is the prediction hour. Net loading follows an autoregressive pattern and therefore, the developed models are tested against a naïve model that uses the closest available past net loading value as the prediction. The naïve model works performs better than random guess for (T+1) up (T+3). Our champion model beats the naïve model for (T+1) up to (T+32). We have used 15 different machine learning models and tried to improve them in 3 modeling stages. Among the machine learning models, the voting ensemble model at the modeling stage 3 gives the best results. The year 2020 data is used as the main test data and 2018, 2019 data is used for modeling.

ÖZET

TÜRKİYE ENERJİ PİYASASINDA GÜÇ DENGESİZLİK TAHMİNİ

HASAN DEMIRTAŞ

İŞ ANALİTİĞİ YÜKSEK LİSANS TEZİ, MAYIS 2020

Tez Danışmanı: Prof. Dr. Can Akkan

Anahtar Kelimeler: elektrik yük dengesizlik tahmini, güniçi piyasa, dengesizlik güç piyasası, gözetimli öğrenme, Türkiye Enerji Piyasası, enerji ticareti

Enerji piyasalarında enerji dengesizliğini tahmin ederek sağlanan potansiyel ticaret fırsatları vardır. Bu çalışmadaki enerji dengesizliği, planlanan nihai üretim ile enerji teslim saatindeki gerçek zamanlı tüketim arasındaki saatlik enerji farkıdır. Bunu net dengesizlik olarak adlandırıyoruz. Bir enerji ticaret şirketinin (TradeCo) bakış açısından, net dengesizliği tahmin edebilmek, gün içi piyasasında (IM) karlı ticaretler yapmaya yardımcı olabilir. Bir enerji üretim şirketi (GenCo) perspektifinden, net dengesizliği tahmin edebilmek, dengeleme gücü piyasasında (BPM) TSO'ya verdiği fiyat tekliflerini optimize etmeye yardımcı olabilir. Bu nedenle, net dengesizliği tahmin edebilmek enerji piyasasında rekabet avantajı sağlayabilir. Bu çalışmada, T tahmin saati olarak alınarak (T + 1)'den (T + 32)'e kadar net dengesizlik sayısal olarak tahmin edilmeye çalışılmıştır. Net dengesizlik, otoregresif bir davranış sergilediği için geliştirilen modelleri mevcut en yakın geçmiş net dengesizlik değerini tahmin olarak kullanan naif bir modele karşı test etmekteyiz. Naif model sadece (T + 1)'den (T + 3)'e kadar rastgele tahminden daha iyi performans göstermektedir. Şampiyon modelimiz (T + 1)'den (T + 32)'e kadar tüm saatler için naif modelden daha iyi tahminleme yapmaktadır. Bu çalışmada 15 farklı makine öğrenimi modeli kullandık ve bunları 3 modelleme aşamasında geliştirmeye çalıştık. Makine öğrenimi modelleri arasında, modelleme aşaması 3'teki oylama topluluk modeli en iyi sonuçları vermektedir. Test verisi olarak 2020 yılı verisi, modellemede ise 2018, 2019 verisi kullanılmıştır.

ACKNOWLEDGEMENTS

I would like to acknowledge everyone who played a role in my academic accomplishments especially Sabancı professors who helped me to gain my knowledge in the area. Special thanks to my thesis advisor Prof. Can Akkan for his support and patience during the process and great Sabancı administrative staff for their help during my education. Thank you all for your unwavering support.

To my family and friends

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1. INTRODUCTION

The objective of this research is to check whether it is possible to predict the usage of the amount of back-up energy when needed i.e. load imbalance amount balanced through balancing power market (BPM) which we will name and call as net imbalance during this study. The back-up energy is previously contracted in BPM with a kind of optioning method meaning transmission system operator (TSO) is free to use or not at the delivery time.

The Turkish energy market is always dominated by the demand at the delivery moment. That is why the generation is arranged in real-time to meet the real-time consumption. That is how the need for a BPM emerged as a perfect matching of consumption and production is not possible. A day before the delivery time in the day-ahead market (DAM), the initial trading contracts are made. Then during the delivery day in the intraday market (IM), the secondary trading is made to balance the deviations from DAM agreements. IM works as a correction mechanism for DAM. The remaining deviation after IM trading is balanced by TSO thanks to the usage of back-up energy options of BPM whose agreements were realized a day before the delivery time. This study can be regarded as an error correction study for TSO's matching model between consumption and generation since it tries to predict the load imbalance. There are three types of Turkish energy markets mentioned in this study:

The first one is DAM. Even though DAM is not in the scope of the study, it is mentioned since both BPM and IM exist to compensate for the deviations from DAM arrangements. DAM and BPM agreements are determined the day before the delivery day whereas IM agreements are determined on the delivery day and the day before as shown in Figure 1.1. The energy trading door for DAM closes at 12.00 on the day before the delivery day. DAM is the primary auction market for power trading. It arranges the hourly energy buy-sell activities for the following day which is the delivery day. The delivery of electricity is based on the contracts made between sellers and buyers. Buyers put their best efforts to estimate the power consumption of their portfolio and sellers try to sell their potential energy generation with a conditional price scheme. Each party states how much they are willing to buy and sell at each price level. Bids are submitted to market operator (MO) by buyers and sellers. Then MO releases the day-ahead prices for each hour. DAM allows the supply side to adjust their price levels depending on their variable costs. DAM also enables market participants to balance their portfolios. This leads to a general fall in the generation and consumption imbalances in the portfolios.

BPM is the second market covered in this study. TSOs gain the ability to balance the supply demand financially in real-time thanks to BPM. It also helps power generation companies (GenCo) to get additional profit by either increasing their load (loading) or by decreasing their load (de-loading). Just after DAM clearing results are published, a GenCo submits its hourly loading and de-loading order bids to TSO for BPM. These orders are options from TSO point of view, whereas liabilities from the points of GenCos which is why they are called orders. Loading orders are usually offered with prices higher than day-ahead prices, and de-loading orders are offered with prices lower than of day-ahead prices to guarantee some profit margin. Making a profit when there is under-supply is straightforward as selling back-up energy at a higher price level than the regular energy price is profitable. Making a profit when there is over-supply is somewhat not intuitive to get immediately. The generation company buys back the energy it sold before with a lower price than its sell price, thus the company makes money for the amount of energy it buys back without producing energy since TSO pays money to the GenCo to lower its production. TSO keeps BPM offers until the delivery time and either accepts or rejects the bids just before it in case an energy imbalance occurs. If the bids approved are via a loading or a de-loading order, the GenCo is obliged to fulfil the request of TSO as stated before. The door for giving balancing energy options to TSO for BPM closes at 16.00 on the day before the delivery day as shown in Figure 1.1.

IM is the third and the last market covered. During the delivery day and the day before, the market participants trade energy. IM trading occurs since the consumption trend predicted in DAM is almost never realized perfectly. Additionally, at the generation side, the non-stable energy production by the wind and the solar power plants in addition to the unplanned malfunctions or incidents of big GenCos are also the factors for IM trades. The delivery day concept is different for IM than DAM and BPM as it is possible to trade as soon as the door opens for (T+1) up to (T+24). The energy trading door for IM opens at 18.00 as shown in Figure 1.1.

In this study, the goal is to forecast net imbalance on the delivery time for both IM and BPM. Trading companies can utilize the future net imbalance predictions starting from the next hour as IM trading is done during the active day. As the process shown in Figure 1.1, generation companies can utilize (T+9) net imbalance prediction as BPM offers are given to TSO a day before at 16.00 which requires to get the predictions at 15.00 for the next day meaning even predicting 00.00 of the next day requires predicting for (T+9), thus it is a harder task. This is why predictions are more beneficial for IM. For this study, an energy TradeCo is partnered with which is why predicting net imbalance for IM and providing insight for the potential IM energy price is considered the primary objective.

An example of energy trading is given in Figure 1.2 and Figure 1.3. Four TradeCos try to balance their portfolios at 23.00 for the next 3 hours, 00.00, 01.00, 02.00. For simplicity, TradeCos do not trade at any hour other than at 23.00 even though it is possible to trade any hour in IM. The energy needs of TradeCos are shown separately for all 3 following hours. Again, for simplicity TradeCos try to balance their portfolios among themselves first and then if there is a remaining need they trade with GenCos. Note that GenCos used for trading in IM are independent of the GenCos that give options in BPM. For 00.00 TradeCos can balance their portfolios among themselves. For 01.00 TradeCos need GenCos to balance their portfolios since their trading among them cannot provide balance as a total of their shortage and excess energy amounts are not the same. For 02.00 a TradeCo does not want to balance its portfolio at IM and wants to balance it at BPM and use the price decided by TSO at the delivery hour. After the IM operations are finished the energy imbalance at the delivery time is balanced by options given by GenCos for BPM. Even though the portfolios are balanced for 00.00 and 01.00 hours in IM, imbalances emerge in BPM since energy demand and production can change until the last moment. Three GenCos give options to TSO at 16.00 for the next day, (T+9) up to (T+32) for BPM. The options are the same for all hours for simplicity. At delivery hour 00.00 there is 5 MWh energy need. That means only the GenCoA can earn revenue by producing the extra energy since it gave the most economic option, which is $5 for each MWh and its 5 MWh capacity is enough to meet the need. The resulting revenue it makes is 5*5 = $25. GenCoB and GenCoC lose their chance for this hour due to their non-competitive prices. If they knew the net imbalance for this hour, they could have given lower prices and generate revenue at this hour. At delivery hour 01.00, there is 10 MWh energy need meaning GenCoA capacity is not enough to meet the need. As a result, GenCoB also sells energy and makes money. GenCoB's $6 price per MWh is higher so it makes generates revenue. The revenue gain difference between GenCoB and GenCoA is the opportunity cost for GenCoA. If GenCoA had known the net imbalance, it could have set its price at a higher level and make more money. At this hour GenCoC again could not make any money due to its non-competitive $7 price per MWh. At delivery hour

02.00, there is a need to reduce the amount of energy produced by 15 MWh. All the energy generation reduction capacity needs to be used since the total energy generation reduction capacity provided by GenCos is 15 MWh. GenCoC makes the most money since its $7 price per MWh is the highest and other GenCos lose money due to the opportunity cost. Again, if the net imbalance had been predicted correctly for this hour, GenCoA and GenCoB could have made more money.

Figure 1.1 Operations in Turkish Energy Markets

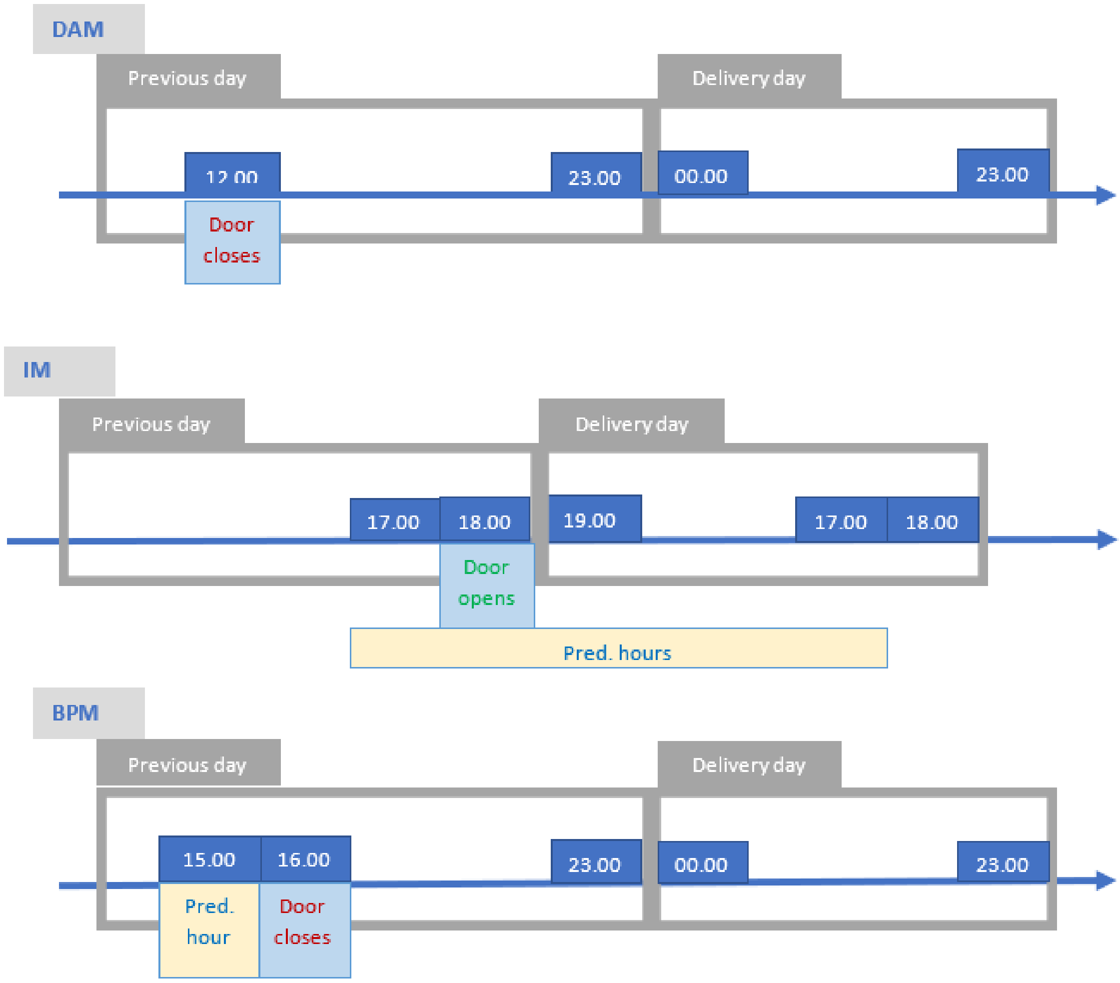
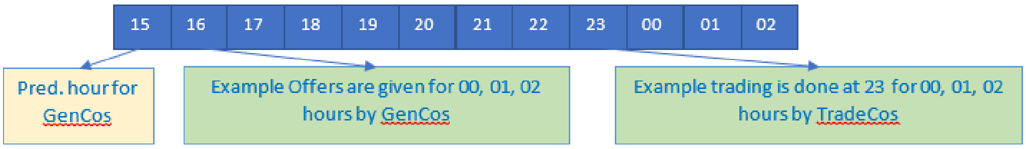


Figure 1.2 An Explanatory Trading Example, IM Part



IM

Need

	For 00-hour	For 01-hour	For 02-hour
TradeCoA	+10 MW-h	+20 MW-h	-20 MW-h
TradeCoB	+20 MW-h	+30 MW-h	-30 MW-h
TradeCoC	-10 MW-h	-10 MW-h	+20 MW-h
TradeCoD	-20 MW-h	-20 MW-h	+15 MW-h

Trade

	For 00-hour	For 01-hour	For 02-hour
TradeCoA	Buys 10 MW-h	Buys 20 MW-h	Sells 20 MW-h
TradeCoB	Buys 20 MW-h	Buys 30 MW-h	Sells 30 MW-h
TradeCoC	Sells 10 MW-h	Sells 10 MW-h	Buys 20 MW-h
TradeCoD	Sells 20 MW-h	Sells 20 MW-h	Buys 15 MW-h

Scenarios

	For 00-hour	For 01-hour	For 02-hour
Action	▪ Trade companies traded among themselves ▪ no external trading involving GenCos is required	▪ Trade companies traded among themselves ▪ external trading also involved as GenCos are required for missing total 20 MW-h energy	▪ Assuming TradeCoB did not want to sell its 15 MW-h energy need in IM. ▪ That is why this imbalance is left to BPM
Balanced	Yes	Yes	No

Figure 1.3 An Explanatory Trading Example, BPM Part

BPM

Offers

	Production Increase Option Price Offer	Production Decrease Option Price Offer
GenCo1	\$5 per MW-h up to 5 MW-h per hour	\$5 per MW-h up to 5 MW-h per hour
GenCo2	\$6 per MW-h up to 5 MW-h per hour	\$6 per MW-h up to 5 MW-h per hour
GenCo3	\$7 per MW-h up to 5 MW-h per hour	\$7 per MW-h up to 5 MW-h per hour

Need

	At 00-hour	At 01-hour	At 02-hour
Need	+5 MW-h	+10 MW-h	-15 MW-h
Reason	Energy balance changed at delivery time even though it was balanced in IM	Energy balance changed at delivery time even though it was balanced in IM	Energy was not balanced in IM, so it is balanced in BPM

Delivery

	Trade at 00-hour	At 01-hour	At 02-hour
GenCo1	Sells 5 MW-h & earns \$25	Sells 5 MW-h & earns \$25	Reduce gen. 5 MW-h & earns \$25
GenCo2	Cannot sell	Sells 5 MW-h & earns \$30	Reduce gen. 5 MW-h & earns \$30
GenCo3	Cannot sell	Cannot sell	Reduce gen. 5 MW-h & earns \$35

Energy price prediction methods are classified into seven main categories; simulation, multi-agent, statistical, computational intelligence, machine learning, hybrid intelligent, and combining forecast. The machine learning category is chosen for this study since the literature is dominated by machine learning models and they have better forecasting performance lately. Actual data from the Turkish power market is used to test the performances of the algorithms.

2. THE ELECTRICITY SECTOR IN TURKEY

Turkey's electricity sector was a state monopoly with the generation, transmission, distribution, and trading functions under the same umbrella before it went through liberalization transformations like the rest of the world. These transformations led to the emergence of different types of electricity markets.

2.1 Liberalization of the Electricity Sector

The electricity sector due to its heavy infrastructural composition was a natural monopoly like other utilities such as gas, telecommunication, water, and sewage services. A barrier to entry in an industry is defined as the initial cost the potential entrant to industry must bear which established players do not experience any more (Lindsay & Stigler, 1969).

The main factors that bring out entry barriers are listed as

1.1 sunk costs,

1.2 social and environmental obligations and regulatory requirements,

1.3 the economies of scale,

1.4 the economies of scope (Poudineh, 2019).

The electricity sector consisted of a unified public company as stated before and separated into multiple companies. For the old end to end unified non-liberated sector, the main entry barrier factor was the sunk cost due to infrastructure investments. It was not possible to bear the cost of a secondary electricity network.

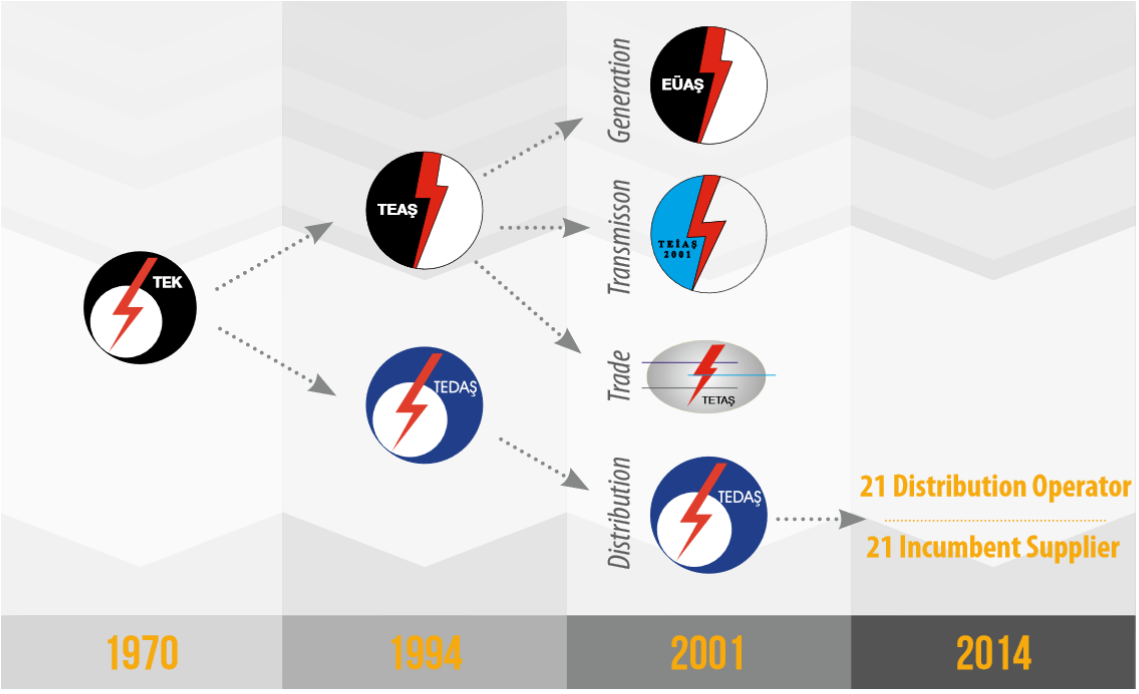
The second important factor was social and environmental obligations and regulatory requirements. Even in a hypothetically ideal world with no cost problems where investing in a secondary redundant electricity network was feasible, there was no space for that kind of a network in the cities. Besides implementing such a network was also non-environmental as it would waste resources of the world.

Thirdly, It is stated the economy of scale concept focuses on the reduction of cost average when there is a higher level of production of one good whereas the economy of scope concept focuses on the reduction of the average total cost of production of a variety of goods (Nickolas, 2019). An electricity supplier who wanted to enter the business might have needed a certain number of customers before its business was profitable.

Lastly, the main business of an electricity supplier is to produce energy and manage to sell the energy to its customers, however, there are side activities that are subject to retail competition such as metering, billing, the credit assessment, receivables collection, and outage reporting which were very hard to compete against the existing company.

It is stated there are two main functional integrations of companies, vertical and horizontal (Dundar & Utas, 2020). The unification of enterprises that are on the same field is called horizontal integration whereas the unification of enterprises that are at the different stages of the business like producing, distributing, and serving the goods is called vertical integration. The sector first started to abandon the integrated structure mainly with vertical unbundling, then continued the liberalization with horizontal unbundling as shown in Figure 2.1.

Figure 2.1 History of Electricity Sector Unbundling in Turkey



Source: (ELDER, 2020)

2.1.1 Vertical Unbundling: Separation of generation, transmission and distribution

Historically, integrations occurred before the unbundling concept emerged as a need to be able to centrally regulate the electricity sector in Turkey. According to the transmission system operator in Turkey, (TEIAS, 2020b), the electricity production started with 2 kW power in Tarsus with a dynamo power unit connected to a watermill in 1902. Twelve years later, the first electric power plant was constructed which was Silahtaraga Power Plant, the current campus of Istanbul Bilgi University (Santral Campus). From those initial times to the beginnings of the 1970s, the investor corporations in the energy sector were governmental financial institutions Etibank and Iller Bankası besides municipalities. Turkish Electricity Authority was established by gathering production, transmission, distribution, and retail services under its umbrella. The corporation remained as Turkish Electricity Corporation (TEK) until it was separated into the Turkish Electricity Generation and Transmission Company (TEAS) and Turkish Electricity Distribution Company (TEDAS) in 1994 meaning the distribution and retail functions were still under TEDAS umbrella. Then TEAS also got separated to Electricity Generation Company (EUAS), TEIAS and Turkish Electricity Trading and Contracting Company (TETAS) in 2001 which could be considered as the completion of the vertical unbundling process.

2.1.2 Horizontal Unbundling: Ownership unbundling, privatization of the generation and distribution companies

Horizontal and vertical unbundling was realized at different paces in different parts of the world. It is stated the European Commission propose considering energy regulation of unbundling of electricity and gas transmission network companies as the preferred form of organization of transmission ownership, with an alternative option of an independent system operator (Pollitt, Davies$*, Price*, Haucap, Mulder**, Shestalova \& Zwart$, 2007). Some countries (e.g. the Netherlands) are in the process of extending electricity and gas distribution networks ownership unbundling even further emulating New Zealand where the creation of standalone electricity distribution network companies was completed in 1999.

It is stated that even though the construction of energy plants was allowed in 1982, it was not possible to reduce the dominance of the state in the energy sector until the 2000s in Turkey (Uluatam, 2011). It is claimed Bereket Energy constructed the first private hydroelectric power plant of Turkey (AYDEM, 2020), Bereket HPP in 1997. It means the private enterprise entry is delayed around 15 years for hydroelectric power plants following the regulatory allowance. After the vertical unbundling of the state-owned electricity companies, the new goal was to change ownership of the state-owned status of the companies by privatizing the generation companies and the distribution companies. On the generation side, the plants were privatized separately. At the distribution side, the country was divided into 21 distribution regions then the trading and contracting companies were privatized jointly with the condition of complete separation of these two functions in the following years. The unbundling legislation did not allow the regions to merge to prevent re-integrations even though some of them are owned by the same conglomerates.

2.1.3 Deregulation: Separation of retail from distribution and allowance

of retail electricity providers

It is noted the privatization of electricity distribution and retail companies in Turkey started with the distribution region of Aydın, Denizli, Muğla provinces i.e. Aydem EDAS in 2008 (EMO, 2012). At the beginning of 2012, 13 of 21 distribution regions were already privatized and the remaining 8 were in the privatization process. These developments were followed by the legislation (EMRA, 2012) which states the retail and distribution functions in the distribution regions were going to be decomposed to different companies. It is pointed out the privatization of the EDAS companies was finished in 2013 (ELDER, 2020). Although the privatization contracts were signed in 2013, the practical separation of retail and distribution followed it with some lag which is shown in Figure 2.2. Turkish Electricity Trading and Contracting Company (TETAS) in Figure 2.1 was closed in 2018 which was not functional after deregulation according to (LegalGazette, 2018).

It is stated that independent retail companies were allowed with the separation of distribution and retail (LegalGazette, 2013). It was the starting point of a half liberal energy market since it only allowed the high electricity consumers as eligible customers (free agents), thus freed them to buy from retail companies. It is stated establishment of Energy Exchange Istanbul occurred in 2015 which was followed by the start of DAM the same year (EXIST, 2020). It is stated the eligible customer lower limit is 1400 kWh a year in 2020, meaning around monthly ∼\$12 (∼80TL), which is a quite low limit and it shows the electricity market liberalization is close to the stage of removing the eligible customer lower limit (EMRA, 2020a).

Figure 2.2 History of Deregulation

Source: (EXIST, 2020)

2.2 Electricity Sector Key Statistics

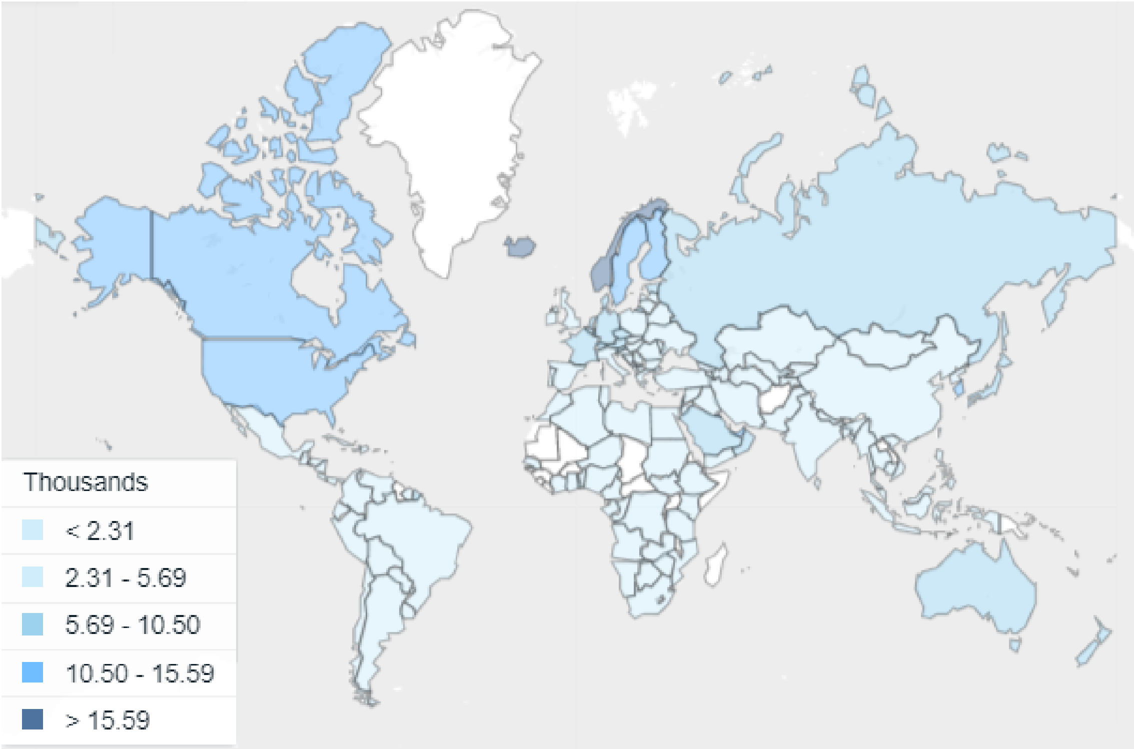
It is showed there is a strong correlation between electric power consumption and the economic development state of counties (WorldBank, 2020d). It is also showed the energy demand per capita did not increase in developed countries in terms of kg of oil equivalent, however it increased dramatically in the fast-developing countries

(WorldBank, 2020b). From 1971 to 2014, the demand per capita increase in Turkey with 2.9 times is above India with 2.4 times and below China with 4.8 times. The Arab world with 4.8 times also had a dramatic change like China. The net GDP per capita increase can be found by dividing GDP increase to dollar inflation. According to (World Bank, GDP per Capita (Current US$) - United States, Turkey) from 1971 to 2014 GDP in the USA increased from $5609 to $62886 which means roughly 11 times the gross increase. It is remarked the dollar inflation calculated by multiplying yearly inflations is roughly a total of 600% for the 1971 - 2014 period (WorldBank, 2020c). The GDP per capita roughly increased 2 times which means the value, or the amount of the products increased 2 times also. This increase is not seen in electric demand in terms of kg of oil equivalent per capita which indicates the increase in energy efficiency or the technological advances in production made it possible to produce 2 times the value with the same amount of energy. GDP per capita in Turkey increased from $455 to $12095 roughly 30 times that is nearly 3 times of USA which explains the relative increase in demand to 3 times the USA's.

When the rapid increase in the energy demand per capita is paired with the increase in population from 35.7 million to 77.6 million from 1971 to 2014 according to (WorldBank, 2020a), the energy consumption (kg of oil equivalent) of Turkey increased more than 6 times. Although the percentage of the energy obtained from renewables over the overall production declined over the years with the decrease of hydro plants' share in Turkey according to (WorldBank, 2020e), the increase in the share of new renewables induce a more dynamic energy production since wind and solar productions are quite weather dependent. It is stated the maximum demand was 46.1 MW and the minimum demand was 18.2 MW meaning 2.5 ratio of max to min in 2018 (TEIAS, 2020a). Along with the increase in overall demand and the introduction of weather-dependent renewables, the dramatic difference between the maximum and minimum requires well-managed forecasts and the retail markets to offer optimum prices to the customers. It is reported the energy consumption was 303,674 GWh in 2019 (TEIAS, 2019). It is envisioned the energy demand as 613 386 GWh meaning doubling the current demand in 2039 (TEIAS, 2020a). It is provided many statistics about the electric sector as it stated the number of consumers in 2019 was 43 million with a 3 percent increase compared to the previous year (EMRA, 2019). It also reported the energy usage shares in percentage regarding the usage purposes as shown in Figure 2.5 and the production shares among private and public companies in Figure 2.6. The sector provides a significant number of jobs and investment in the economy of Turkey as the total number of personnel working in EDAS companies was around 57000 and 32000 of them were employed via subcontractor companies of the EDAS companies. Besides, the total investment in

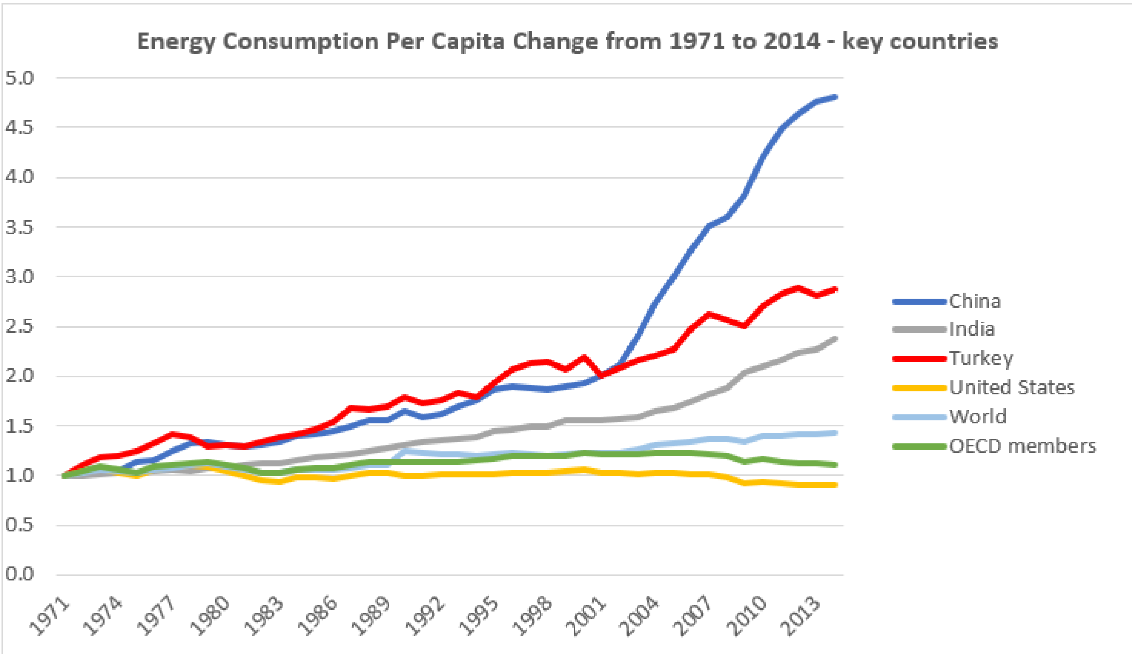
2019 towards the transmission system was $0.48 billion and towards the distribution system was $1.28 billion.

Figure 2.3 Energy Usage (kg of oil equivalent per capita)



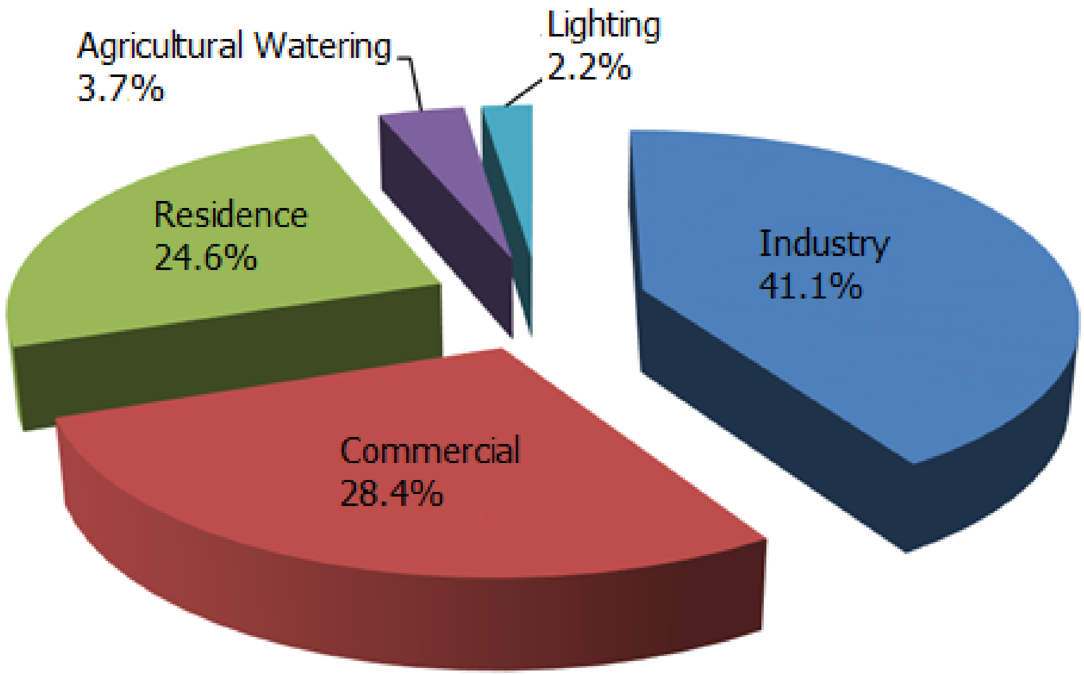
Source: (WorldBank, 2020c)

Figure 2.4 Energy Consumption Change – Key Countries



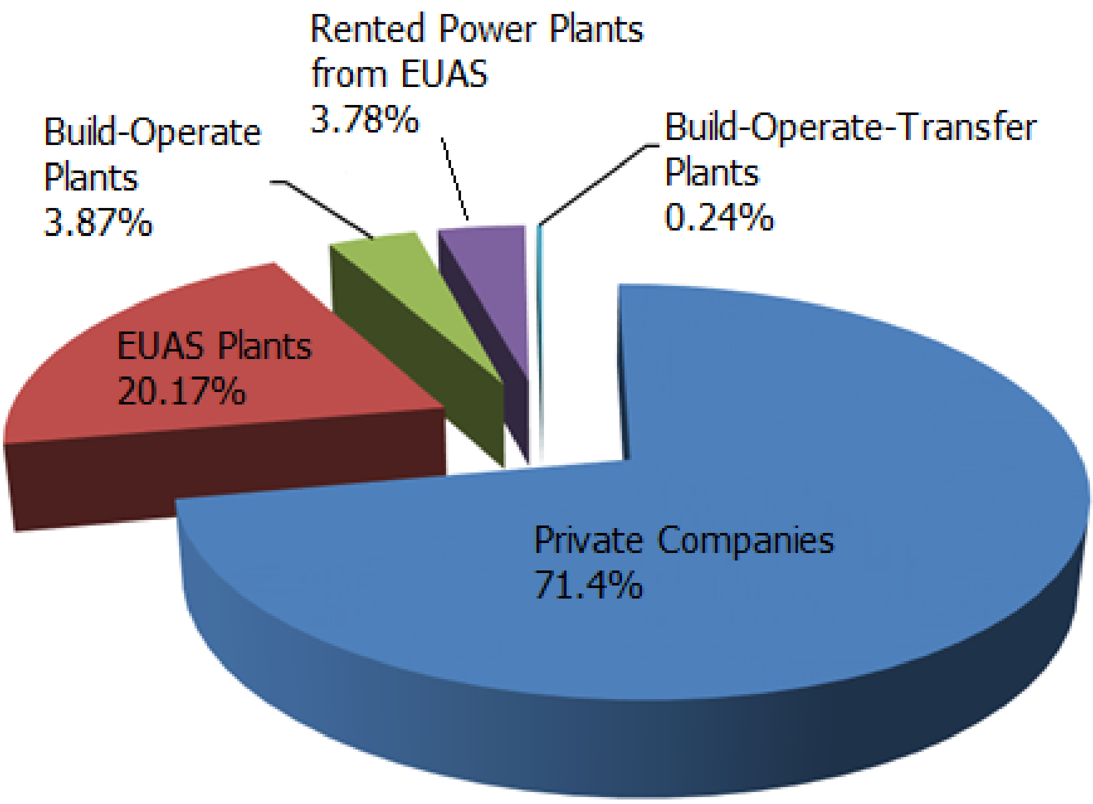
Source: (WorldBank, 2020b)

Figure 2.5 Energy Consumption Shares according to Consumer Types



Source: (EMRA, 2019)

Figure 2.6 Energy Production Shares Among Private and Public Companies



Source: (EMRA, 2019)

2.3 Power Trading

As it is stated the short-term power trading in Turkey is performed in three different types of markets which are DAM, IM, BPM (EMRA, 2019). The price determination formulas are open to the public whoever wants to investigate deeper, however, they are too complicated and beyond the scope of this thesis work to cover. That is why the objectives, principles, and operations of the three main markets are explained in detail without touching on the price formulations.

2.3.1 Day-ahead Market

DAM is defined as an organized wholesale electricity market established for electricity energy buying and selling based on the settlement period to be delivered after a day and operated by MO, Energy Exchange Istanbul (EMRA). It consists of activities carried out to balance supply and demand in the system and balance market contracts and production and/or consumption plans for the delivery day (EMRA, 2020b).

DAM objectives are:

- Enabling market participants to balance their production and/or consumption needs and their contractual obligations the day before.
- Determining the electrical energy reference price.
- Helping TEIAS, the transmission system operator (TSO), for a balanced system from the day-ahead.
- Helping TEIAS to perform constraint management from the day-ahead.
- In addition to bilateral agreements, market participants create the opportunity to buy and sell energy for the next day.

The general principals of DAM:

- DAM transactions are carried out daily, on an hourly basis. Each day consists of hourly time slots starting at 00:00 and ending at 00:00 the next day.
- The transactions in DAM correspond to constant supply or demand commit-

ments over the relevant period meaning an average consumption amount is assumed to be realized and generation companies produce a stable amount of power during that period to meet the demand. The deviations from these ideal assumptions are balanced thanks to IM and BPM.

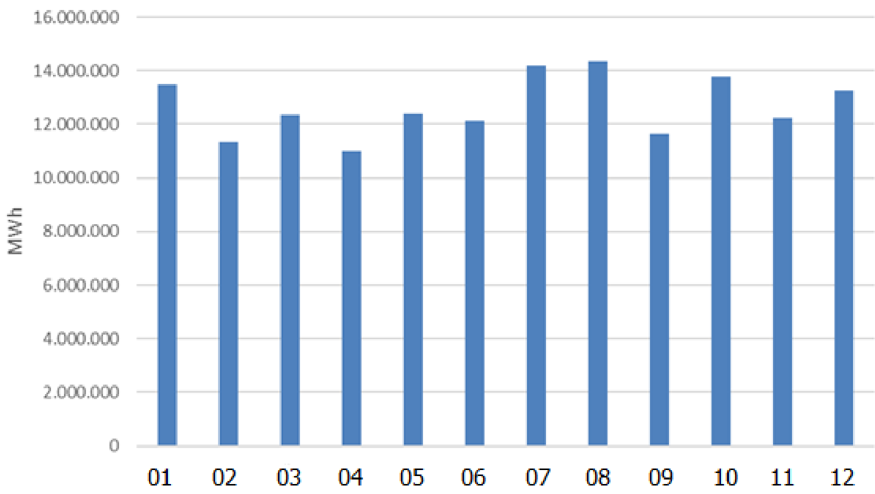
- In DAM, all offers are used for a certain day, and a certain period within that day.

The operations of DAM:

- Between 12:00 - 13:00 every day, MO calculates the market clearing price, equilibrium monetary value determined by the bid-ask process of buyers and sellers, for each hour of the next day and each bid region.
- Every day at 13:00; MO notifies the market participants that participate in DAM commercial transaction confirmation, which includes the purchase and sales amounts of each market participant in DAM. In other words, the market players are informed about the acceptance of the appropriate purchase-sale offers and rejection of their non-economical offers considering the equilibrium price.
- Every day between 13:00 - 13:30; Market participants participating in DAM check the commercial transaction confirmations notified to them by MO and report their objections regarding commercial transaction approvals to MO when necessary.
- Every day between 13:30 - 14:00; MO evaluates the objections and informs the relevant market participants about the results of their objections.

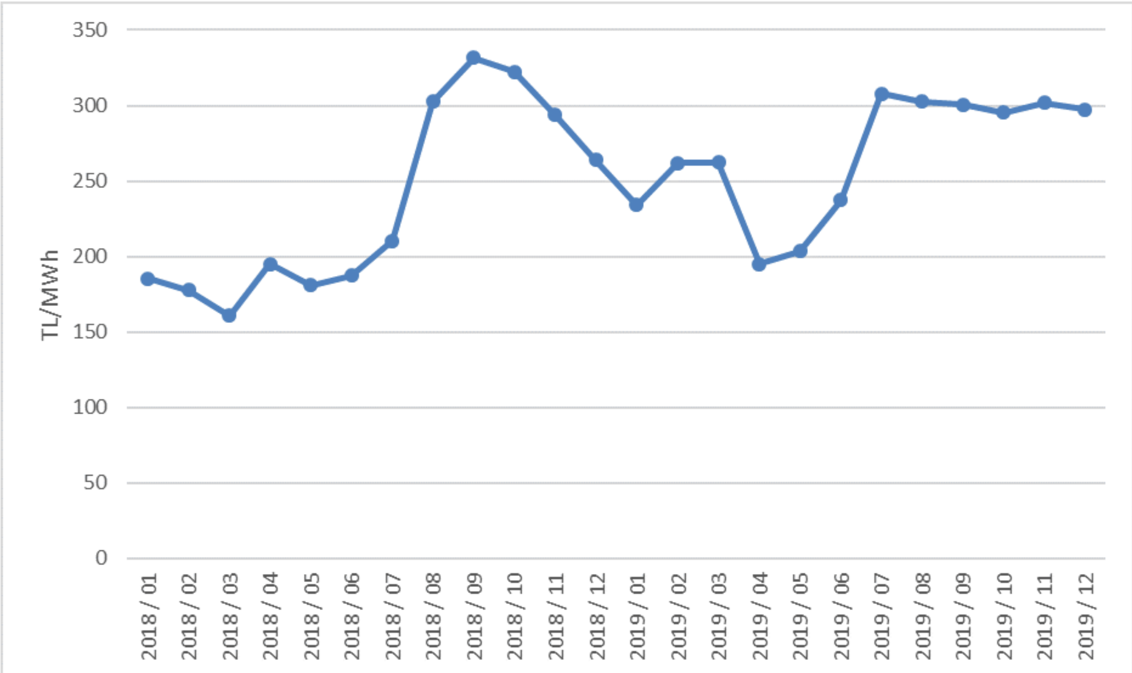
It is mentioned the history of DAM as it points the first step taken in line with the goal of transitioning from the single buyer, single seller model to a free and competitive electricity market model was to switch to the monthly 3-time settlement system on July 1, 2006 (EMRA, 2019). The next step was DAM Planning system, which became operational on December 1, 2009. These transition periods were very important for the electricity market to be stronger and more dynamic. The experiences gained by the parties involved in the operation of the market, the experiences gained in each transition period and the developments they envisaged were transferred to new market models. December 1, 2011 date was a milestone for the Turkey Electricity Market as it was the launch date of DAM.

Figure 2.7 Monthly DAM Matching Energy Amounts for 2019



Source: (EMRA, 2019)

Figure 2.8 Weighted Average DAM Clearing Prices for 2018, 2019



Source: (EMRA, 2019)

2.3.2 Balancing-power Market

It is defined BPM as the organized wholesale electricity market operated by TSO, where the purchase and sale of the spare capacity obtained with the output power change that can be realized in fifteen minutes to serve the purpose of balancing the supply and demand in real-time (EMRA, 2020b).

BPM objectives:

- Balancing active electrical energy supply and demand in real-time.
- Real-time balancing, ensuring that electrical energy is available to consumers in an adequate, quality, continuous, and cost-effective manner.

BPM objectives:

- BPM offers are given daily, on an hourly basis. Each day consists of hourly time slots starting at 00:00 and ending at 00:00 the next day.
- All offers submitted to BPM are valid for a certain balancing unit, a certain offer region, a certain day, and a certain time period within that day.
- In proposals submitted to BPM, it is essential to propose all the technically capable capacity of the relevant balancing unit in line with the structure of the proposal submitted.
- Within the scope of BPM, BPM commitment orders can be given by TEIAS at any time from the finalization of the day-ahead production/consumption schedule and the end of the physical delivery time.

The operations of BPM:

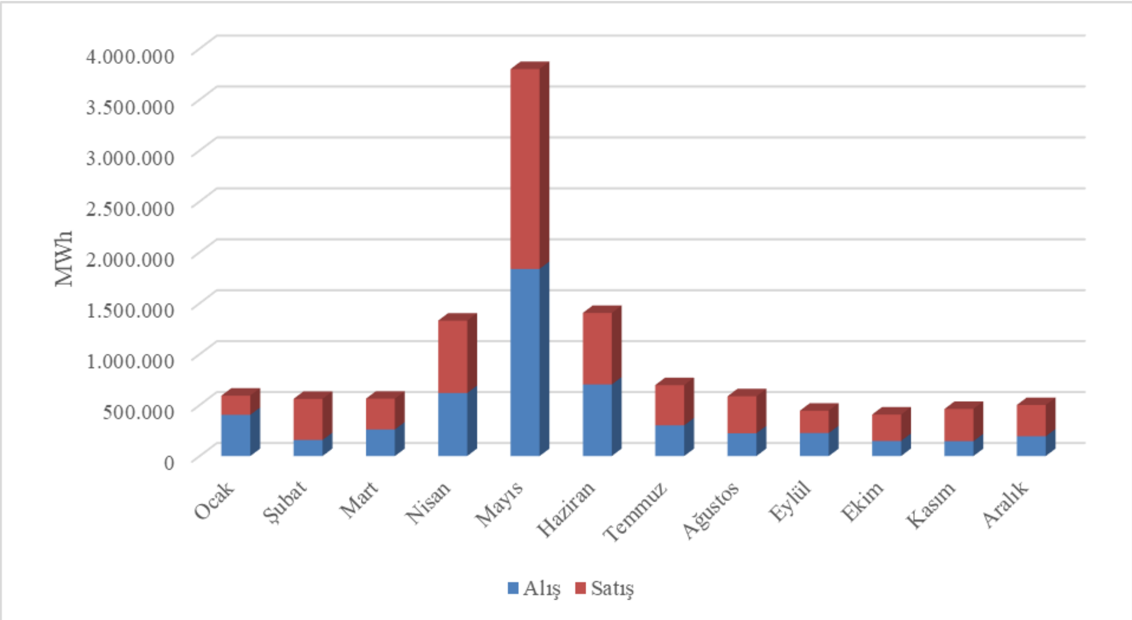
- Until 16:00 every day, each market participant participating in BPM will have final day-ahead production/consumption programs that include hourly production or consumption values for all settlement mediation-traction units registered in his name and notifies TSO about the up-regulation and down-regulation offers regarding BPM.
- Until 17:00 every day, TSO checks the final day-ahead production/consumption program notifications and offers for bids plus loads and determines whether there are any material errors in the notifications. TSO gets in touch with the relevant market participant regarding the erroneous notifications and makes necessary corrections until 17:00.
- The up-regulation and down-regulation offer submitted within the context of BPM are sorted by TSO in the price order for each offer region and each hour.
- As of 17:00 every day, taking the load offered by TSO within the context of BPM in order to eliminate the energy deficit or surplus occurring in the system

related to the relevant day or foreseen the future, to create the capacity for removing system constraints and/or providing ancillary service. Load shedding bids are evaluated and instructions regarding the bids approved are informed to the relevant market participants. Notifications regarding the termination of the instructions are made to the relevant market participants.

- System marginal prices determined in BPM for each hour are determined by TSO within four hours following the relevant time and announced to the market participants.

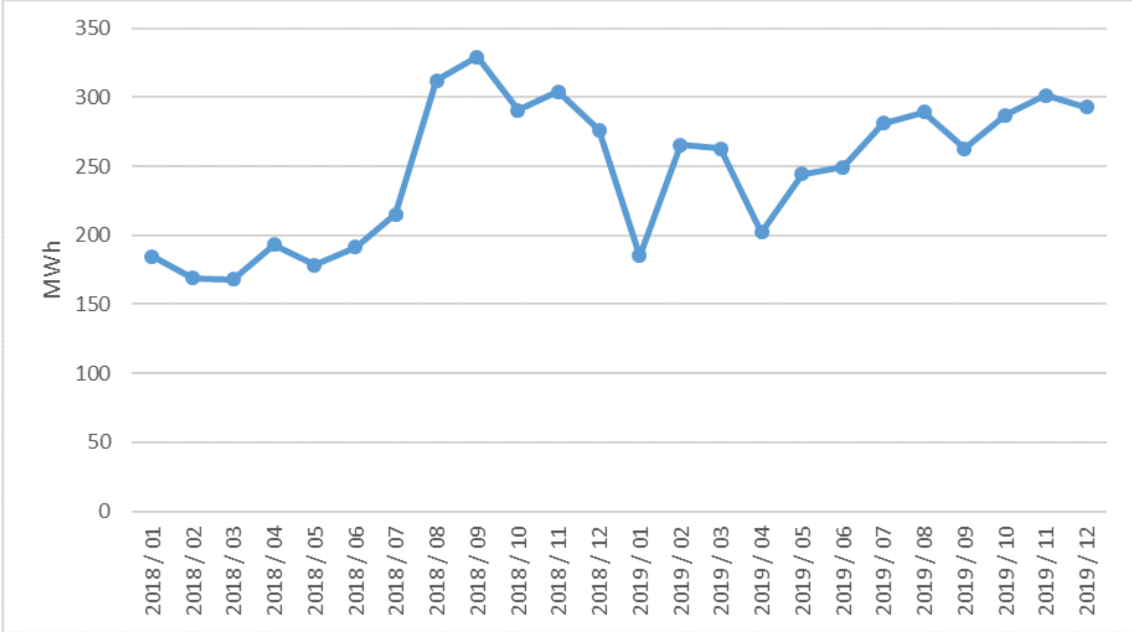
Monthly Volumes in BPM in 2019 are shown in Figure 2.9 and BPM Monthly Weighted Average Prices in 2018 and 2019 are shown in Figure 2.10.

Figure 2.9 Monthly Volumes in BPM in 2019



Source: (EMRA, 2019)

Figure 2.10 BPM Monthly Weighted Average Prices in 2018, 2019



Source: (EMRA, 2019)

2.3.3 Intra-day Market

It is defined IM as an organized wholesale electricity market where electricity trading is done until the closing of IM (EMRA, 2020b). It consists of activities carried out with the aim to make trading possible during the delivery day. Its activities are shaped by production and/or consumption plans made by DAM throughout the day and deviations from them. The BPM participants' contractual commitments effective on the prices in the market, thus it is interested in the terms of BPM participants' contractual commitments. Main responsible is MO.

Intra-day market objectives:

- Enabling market participants to balance contractual commitments and production and/or consumption plans.
- Ensuring the reduction of energy imbalance amounts.
- Providing a balanced system prior to real-time balancing to TSO.
- Creating energy trading opportunities to market participants, in addition to the bilateral agreements and trading in DAM.

The general principals of intra-day market:

- The operations can be either hourly or in blocks. The next day hourly contracts are opened at 6pm the day before. The transactions in IM can take place at any time until IM door closes.
- IM door closing time is one hour before physical delivery.

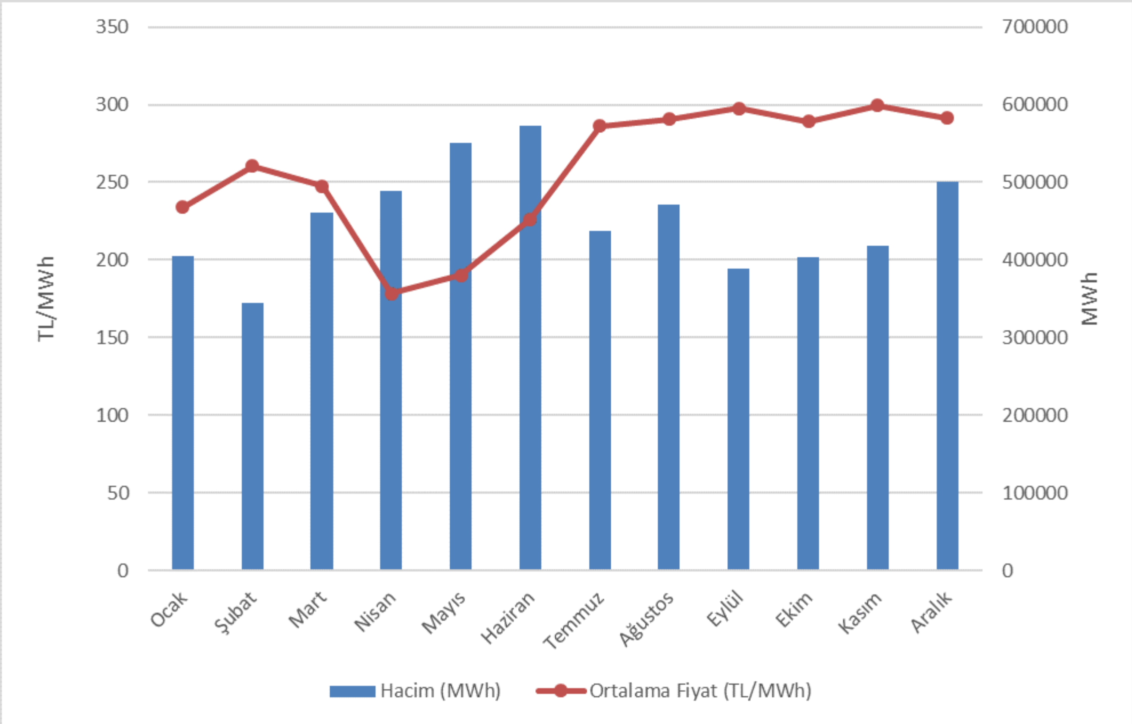
The operations of IM:

- IM participants report IM offers to the MO every day starting from 18:00 until IM door closing time for the next day (delivery day). It means intraday arrangements via IM for delivery day's earlies hour 00:00 can be made at most 6 hours before the consumption.
- IM offers can be updated, cancelled, or suspended by the relevant market participant until the validity period of the related contract expires unless it matches. In other words, a seller can increase or decrease the price that it told until a buyer buys energy at the seller's offered price. The same is valid for a buyer as it can change the price level it buys until a seller provides energy at that price. The system settles at the latest update on the proposal, considering time information.
- IM participants check the commercial transaction confirmations notified to them following the matching of the offers and notify their objections to the Market Operator.

It is stated IM has become operational on July 1, 2015. With IM (EMRA, 2019), which was brought in addition to DAM, BPM, which were already operating, real-time trading opportunities were provided, and market participants were given the opportunity to balance their portfolios in the short term.

Monthly IM matching energy amounts and prices for 2019 are shown in Figure 2.11. Even though it seems like for the April, May and June months, monthly volume and average price in IM seems slightly negatively correlated compared to the other months, there is no well-defined business-related explanation for this weak correlation according to the business owner in the partner TradeCo.

Figure 2.11 Monthly Volume and Average Price in IM in 2019



Source: (EMRA, 2019)

3. RESEARCH OBJECTIVES

In the Turkish electricity market, the participants deal with penalties that arise from both overproduction and underproduction through BPM. Therefore, to trade optimally it is not enough to know the expected power generation or power consumption values separately. The market players should also position themselves against the energy difference between consumption and production which we call as net imbalance to minimize expected balancing costs. The realized net imbalances are expected to be representative of the intra-day prices of the following hours. Focusing on this pointed relation is the potential extension of this study as we only predict net imbalance and do not use the results for a second prediction for IM prices as a stacked machine learning model fashion.

The energy production and demand values for the next day are sent to MO (EPIAS) until 12:30. MO accepts energy production offers from the cheapest to the more expensive production offer until the energy demand prediction is met, which is also called merit-order. The certain results are announced at 14:00. The deviations from the predictions are inevitable in energy consumption. That is balanced by TEIAS accepting the energy production decrease or increase offers from the power generation companies. The balancing plan is arranged to cost minimum as the offers are listed from cheapest to the most expensive one to be used when needed for both increasing the generation or reducing the generation. The offers are sent to TEIAS until 16:00.

The first objective of this work is to check whether it is possible or not to predict the imbalance during the day. During the day, a TradeCo keeps track of its customers' consumption. When the consumption is higher than the energy it purchased previously in DAM, a need for meeting the energy shortage arises. The missing energy is purchased in IM. If the company cannot find or choose not to find the missing energy, the missing energy is supplied at the price dictated by TEIAS which can be considered as a penalty price since it is the energy produced without any planning. When the consumption is lower than the energy the TradeCo purchased previously in DAM, this time the company tries to sell the excess energy in IM. If it cannot sell by itself, the selling price is the price TEIAS dictates since TEIAS is the ultimate buyer and seller. By predicting the net imbalance (imbalance in TEIAS prediction) during the day with the previously announced net imbalance values, the target is to help the company to position against the intra-day buy-sell events. Two main actions are possible with a successful prediction. The first one is to minimize the penalty price sell-buy action due to an imbalance in customers' consumption and energy provided by the TradeCo. The second one is optimizing intra-day energy trading. The companies have more idea about the potential system net imbalance as the delivery hour gets closer. If it is possible to predict the imbalance (T+1) up to (T+24) hours before, the energy can be purchased when it is cheaper and can be sold when it is more expensive. Let t denote the delivery hour (recall that T denotes the current hour). By predicting over-demand at (T+8), thus energy production deficiency, TradeCos can buy X amount of energy at the market price at t–8, then sell the purchased excessive energy at the market price at t–1 time with a P profit, the company can make X * P amount of money in 7 hours.

The second objective of this work is to check whether it is possible or not to predict the imbalance for the next day before sending buy/sell options to TEIAS at 4 p.m. Assuming the energy company's production cost is C and the company's profit is P for regular energy generation and sell activity, the regular generation offer for DAM would be C + P. The company only accepts to reduce the energy it produces for BPM if the profit coming from the energy reduction offer is bigger than its regular profit P which is P + P, assuming no regulatory forced action. That is why GenCos give energy reduction offers as options to TSO for BPM with such prices that help them make more money by not generating the amount of energy given in the option rather than generating it. On a balanced day, offering a large P can result in no profit from BPM as the offer would not be realized at the delivery time, since it will not be needed until the cheaper options run out. On the other hand, during an extremely unbalanced day, a big P value brings profit since the cheaper options run out and the energy is sold even though it is more expensive due to the need. The offers are noticed at 6 p.m., which means the analysis needs to be completed previously. Assuming a 1-hour operational buffer at 5 p.m. the analysis needs to be completed. The real-time energy imbalance is announced with an hour delay which means the imbalance at 4 p.m. is available. The prediction for the next day is at least 8 hours ahead. The target is to predict 8-32 hours ahead for the next day. Any prediction better than random is plausible for this case.

4. POWER MARKET FORECASTING LITERATURE

One of the most similar works to ours is performed in the Polish market (Popławski, Dudek & Łyp, 2015). It is claimed their prediction method which they called "a similarity-based method; fuzzy estimator of the regression function" beat machine learning methods random forest and neural network when applied for the Polish balancing market's 15-minute balance prediction periods of the following day. The best method for predicting the reserve capacities for the next day (day-ahead) is found as LASSO with penalized quantile regression in the Austrian balancing market using public data (Essl, Ortner, Haas & Hettegger, 2017). Their study utilized quarter-hourly values of load-, generation-, wind, and photovoltaic-forecasts for the year 2015 a total of 53 variables. Besides machine learning models, stochastic models are also used with the same purpose in the Norwegian market. It is found for the short term forecast (1-hour ahead), Seasonal Autoregressive Integrated Moving Average (SARIMA) model was the best whereas, for the day-ahead forecast (12-36 hours ahead), CROST (an autoregressive model for unevenly spaced time series found by Croston) was the best model in balancing market volume forecast (Klæboe, Eriksrud & Fleten, 2013). At the price side, they forecasted balancing market premium and found using a naïve approach, the balancing market price from the last hour, was the best for the short-term forecast (1-hour ahead). For the day-ahead forecast (12-36 hours ahead), the performances of the models were not satisfactory. As an alternative method it is tried to model the balancing energy demand as a mathematical function of market-related variables which are the gradient of load, a arbitrage incentive, a technical incentive, and a varying general market position, a non-predictable event risk which can be considered to use a business-related method rather than a data science approach (Möller, Rachev & Fabozzi, 2011). In another work, energy price predictions in the German power exchange market is focused on (Uniejewski, Marcjasz & Weron, 2019). According to them, the most important feature for IM was the price at the previous hour. This is an expected result since both energy production and consumption show auto-regressive behaviour due to their nature. A second notable finding of the study was the performance of the naïve model, the price from the last hour, over some of the machine learning models.

It is claimed they could achieve to train a neural network for an intraday hourly load forecast with 1.5% MAPE with 89 days data and electricity consumption and temperature-based 7 features in the Bosnian energy sector (Becirovic & Cosovic, 2016). The significance of the study was the number of data points which 2136 instances for winter (89 days) and 2189 instances for summer (91 days).

If desired, it is possible to find short-term load forecasting studies in the 1990s. Forecast the half-hourly electric load of the power system of Kuwait with a neural network model is tried (AlFuhaid, El-Sayed & Mahmoud, 1997). They claimed the analysis as significant as it decreased both the average absolute forecasting error and the maximum absolute error. Various load forecasting studies in different parts of the world were performed. In a study for east asia, day-ahead load in Hong Kong is predicted (Chow & Leung, 1996) whereas day-ahead load in Crete is also predicted in another early study (Kiartzis, Zoumas, Theocharis, Bakirtzis & Petridis, 1997). The studies at those years use day-ahead and short-term forecast concepts together since there was no concept at those days which can be contradictory with the current market literature as is often called as spot market and short-term forecast phrase is used for .

Lastly, two novel studies in the Turkish electricity market are covered. Predicting intra-day electricity prices is tried and it is found that gated recurrent unit (GRU) and long short-term memory (LSTM) neural network models perform best with the data from Jan 2017 to Feb 2019 (Oksuz & Ugurlu, 2019). The same group continued their research in this area as they tried to check whether modelling knowledge transfer between different power markets is possible. It is found it was possible to utilize the transfer learning concept of neural network by putting a pre-training step with the data of other countries in DAM (Gunduz, Ugurlu & Öksüz, 2020). The markets in the study were Belgium, Germany, France, Norway, and Turkey. As expected, the model performance increases more significantly when less data is available for the training.

In Figure 4.1, the key points of the selected works are listed.

Figure 4.1 Power Market Forecasting Literature Key Points

	Title	Findings	Data
Popławski et al. (2015)	Forecasting methods for balancing energy market in Poland	They claim their prediction method performs better than well-known ML models for BPM	Polish Power Market
Essl et al. (2017)	Machine Learning Analysis for a Flexibility Energy Approach towards Renewable Energy Integration with Dynamic Forecasting of Electricity Balancing Power	Lasso is best for predicting the energy imbalance for DAM	Austrian Power Market
Klæboe et al. (2013)	Benchmarking time series-based forecasting models for electricity balancing market prices	Naïve model is the best for predicting (T+1) price by using balancing prices for IM	Norwegian Power Market
Möller et al. (2011)	Balancing energy strategies in electricity portfolio management	Mathematical formulation with business related features are used for prediction rather than statistical learning methods for DAM	German Power Market
Uniejewski et al. (2019)	Understanding Intraday Electricity Markets: Variable Selection and Very Short-Term Price Forecasting Using LASSO	The most important feature for IM was the price at the previous hour and naïve model beats some ML models	German Power Market
Becirovic & Cosovic (2016)	Machine Learning Techniques for Short-Term Load Forecasting	They claim the results were significant even with 90 days (2136 hours) data	Bosnian Power Market
Oksuz & Ugurlu (2019)	Neural Network Based Model Comparison for Intraday Electricity Price Forecasting	They used neural networks, GRU and LSTM, to predict IM prices	Turkish Power Market
Gunduz et al. (2020)	Transfer Learning for Electricity Price Forecasting	Knowledge transfer is possible between markets with transfer learning (by pre-training neural networks) for DAM when the training data is scarce	Belgium, Germany, France, Norway, and Turkey Power Market

5. THEORETICAL BACKGROUND FOR FORECASTING WITH MACHINE LEARNING

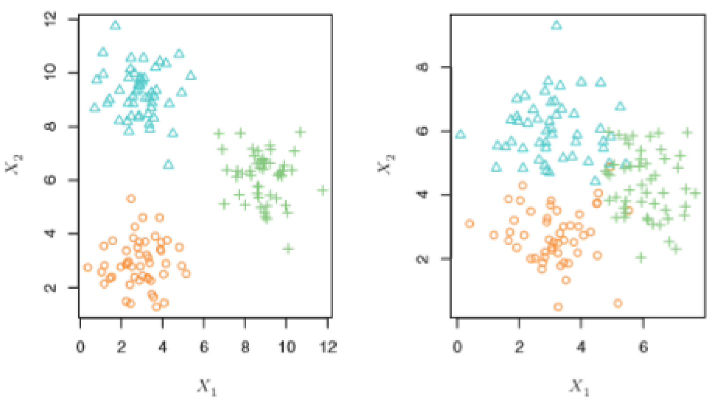
5.1 Main Machine Learning Concepts

Machine learning, data mining, and statistical learning are very similar concepts to find valuable information in data. The method is to use a part of the historic data for model training and a part of it for validating the trained model. The trained and validated model is tested by data the model has not seen before which is generally separated from modelling data by time. After the test results are successful. The successfully trained, validated and tested model is used to provide valuable information for the business party, in this case, the energy TradeCos.

Supervised learning is the methodology of identifying the similarities between data points directed by the purpose of predicting a target feature. For example, a salary prediction supervised model decides to use features like education, gender, and profession if and only if they can explain the salary. It is the main method that we use for this study.

There is no obvious target for model training in unsupervised learning. It is indicated unsupervised learning is a methodology in which for every observation i = 1,...,n, we observe a vector of measurements x_i but no associated response y_i (James, Witten, Hastie & Tibshirani, 2013). It is not possible to fit a simple linear regression model since there is no response variable to predict. Working blindly without the lead of a response variable is called unsupervised because of the absence of supervision of a response variable. One method is checking whether the observations can be grouped (clustered) as relatively distinct groups as shown in Figure 5.1. A clustering task with two variables and a goal to represent them in three groups is visualized in that figure. The task is easier when the data points are easily separable like in the left illustration, and it gets relatively complicated when the data points are overlapping like in the right one.

Figure 5.1 Clustering



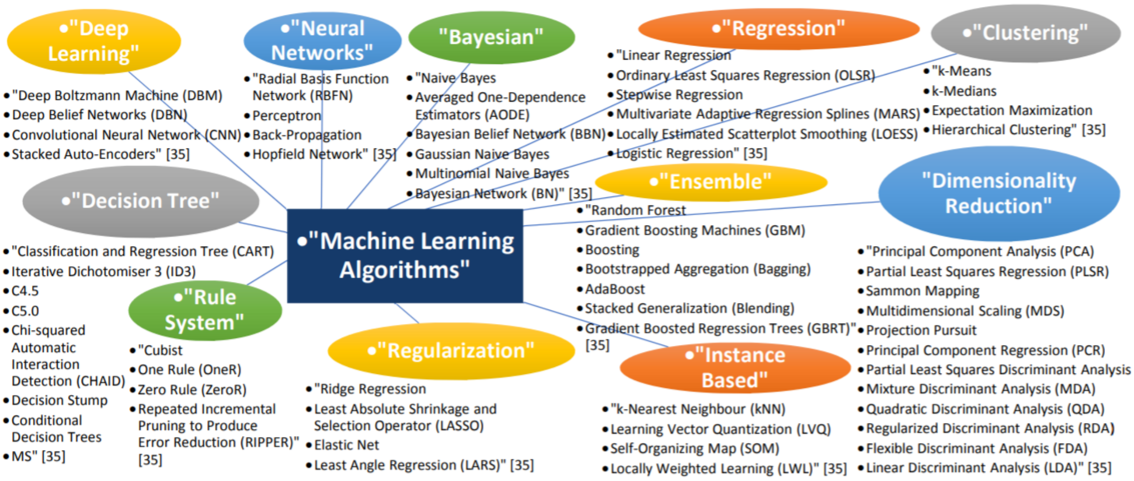
Source: (James et al., 2013)

Evaluation metrics are the numeric values to understand the success of the models.

- Common performance metrics used for a categorical target are below:
  - accuracy; $\frac{\text{sum of number of correctly predicted true values and false values}}{\text{total number of values}}$
  - precision; $\frac{\text{number of correct true predicted values}}{\text{total number of true predicted values}}$
  - recall; $\frac{\text{number of correct true predicted values}}{\text{total number of true values}}$
  - f-score; $\frac{2}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}}$
- Common performance metrics used for a numeric target are below where y = actual, ŷ = predicted, n = data amount (rows), k = number of features (cols):
  - MAE; Mean absolute error, $\frac{1}{n}\sum|y-\overline{y}|$ uses the average error amount
  - MAPE; Mean absolute percentage error, $\frac{1}{n}\sum\frac{|y-\overline{y}|}{y}$ uses the average error amount
  - RMSE; Root mean square error, $\sqrt{\frac{1}{n}}\sum|y-\overline{y}|^2$
  - R-square; Coefficient of determination, $\frac{MSE(mean)-MSE(model)}{MSE(mean)}$, explained error percentage thanks to the model where $MSE=\frac{1}{n}\sum|y-\overline{y}|^2$
  - Adjusted R-square; $1-\left(\frac{n-1}{n-k-1}\right)*(1-R^2)$, puts a basic penalty for additional features, i.e. among the models with n & 2n features and the same R-square values, the one with the less features (n) is preferable since the same performance is obtained with less variables

Figure 5.2 gives a compact visualization for most of the main machine learning algorithms.

Figure 5.2 Machine Learning Methods at a Glance



Source: (Essl et al., 2017)

5.2 Supervised learning methods

In this study, there are two parts that many machine learning algorithms are used. The first one is feature selection for regression, the second one is predicting the system balance as a numeric value. Multiple feature selections algorithms can be used together with a mechanism called voting. The working mechanism of voting is counting the number of methods that indicate whether the variables are important or not, then keeping the variables above a certain vote threshold. The reason many feature selection algorithms are chosen with a voting method is to prevent overfitting to choices of a single algorithm thus preventing bias. For feature selection purposes, four different main methods and thirteen different sub-methods are used. At the prediction part, mainly linear models and tree-based models are used. Unless otherwise is stated, Scikit-learn is used for all the methods whenever the base python is not enough for the desired operation (Pedregosa, Varoquaux, Gramfort, Michel, Thirion, Grisel, Blondel, Prettenhofer, Weiss, Dubourg, Vanderplas, Passos, Cournapeau, Brucher, Perrot & Duchesnay, 2011). Table 5.1 shows the usage of the algorithms:

Table 5.1 Machine Learning Methods

Linear Regression	**Correlation**
Lasso	SelectKBest: f_regression (anova-f)
Elastic net	SelectKBest: mutual_info_regression
Knn Regression	RFE: SGDRegressor
Random Forest	RFE: ElasticNetCV
Extra Trees	RFE: LassoLarsCV
Adaboost	RFE: OrthogonalMatchingPursuitCV
Gradient Boosting	RFE: AdaBoostRegressor
XGBoost	RFE: GradientBoostingRegressor
LightGBM	RFE: ExtraTreesRegressor
CatBoost	SelectFromModel: RandomForestRegressor
Naïve method (latest available self-value)	SelectFromModel: RidgeCV
Stacking ensemble	SelectFromModel: LGBMRegressor
Voting ensemble	

5.2.1 Linear Models

A linear model tries to relate the dependent variable, target, the independent variables with a mathematically linear relation:

$$\overline{y} = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2$$

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \epsilon$$

$$y - \overline{y} = \epsilon$$

Note that any function which can be reduced to the above formula with transformations is a linear model. For instance, y = $\beta_0 + \beta_1 * x_1^2$ is again a linear model since the function can be reduced to linear by assigning $z = x_1^2 => y = \beta_0 + \beta_1 * z$. It tries to minimize the difference between the prediction and real data, error, by minimizing the sum of squared errors. The purpose of using squares is penalizing the big errors more compared to small errors.

There are many linear models. The ones used as part of this thesis research are briefly explained below:

5.2.1.1 Ridge regression

In statistics, a less complex model with the same performance is preferable. Ordinary linear regression does not have an inbuilt method to omit the features which only provide noise if the R-square value is even slightly improved with the contribution of noise variables. Ridge tries to address that problem by adding a penalty to the error term. It is stated in particular, the ridge regression coefficient estimates are the values that minimize $RSS + \sum_{j=1}^{p} \beta_j^2$ where $\lambda \geqslant 0$ is a tuning parameter, to be determined separately (James et al., 2013). As with least squares, ridge regression seeks coefficient estimates that fit the data well, by making the RSS small. However, the second term, $\lambda * \sum_j \beta_j^2$, is called a shrinkage penalty, is small when $\beta_1 \ldots, \beta_p$ are close to zero, and so it has the effect of shrinking the estimates of β_j towards zero

5.2.1.2 Lasso regression

Lasso is another method to penalize unnecessary inputs. It is stated that lasso and ridge regression have similar formulations (James et al., 2013). The only difference is that the β_j^2 term in the ridge regression penalty has been replaced by $|\beta_j|$ in the lasso penalty. In statistical parlance, the lasso uses an L1 penalty instead of an L2 penalty. $\lambda * \sum_j |\beta_j|$ is the resulting penalty.

5.2.1.3 Elastic net regression

Elastic net uses a combined penalty of L1 and L2 together. The weights of the L1 and L2 penalties sum to 1 which means they are inversely related. Elastic-net penalty is introduced as a different compromise between ridge and lasso (Zou & Hastie, 2005). Its equation, $\lambda * \sum_{j=1}^{p} (\alpha * \beta_j^2 + (1 - \alpha) * |\beta_j|$, shows how ridge and lasso are combined with the α term.

5.2.2 Tree based methods

A tree is created by splitting data according to conditions. Splitting starts at the root where all the data points are together and continue with the successive conditions. The way tree methods split data is similar to a tree shape. That is why they are called it. The splitting is based on information gain. Assuming completely balanced binary data with fifty percent share in each category, the splits try to increase the odds ratio at the new data segments created with splits. The tree algorithm would prefer a split option with the resulting two data segments consist of '(80% CategoryA, 20% CategoryB) and (20% CategoryA, 80% CategoryB)' over a split with the resulting two segments consist of '(60% CategoryA, 40% CategoryB) and (40% CategoryA, 60% CategoryB)' as the odd ratios for the first option are (4/1 , 1/4) and for the second option are (6/4 , 4/6). The target is to create final data fragments with the best odds ratios. Note that, generally the data is not balanced, and splitting data into two equal sizes at nodes is not optimal. That is why methods as entropy gain and Gini are developed and used which we will not cover here.

Decision tree methods can be used both for regression and classification (James et al., 2013). They segment the predictor space into several simple regions. The mean or the mode of the training observations in the region to which it belongs is used to predict a data point. The rules used to divide the predictor space can be summarized in a tree representation.

5.2.2.1 Single Classification and Regression Tree

Using trees for a single algorithm created the CART concept (Gordon, Breiman, Friedman, Olshen & Stone, 1984). Metrics and methods for a computer to use decision trees are also defined by the creators of CART. Their work later helped the evolution of more complex tree algorithms. The main advantage of using a single tree is the interpretability of the model.

5.2.2.2 Bagging Regression

Random forest algorithm is established over bagging, that is why it is introduced before explaining random forest. Bagging is introduced as an improvement to CART idea (Breiman, 1996). Bagging is using multiple decision trees based on randomly taken samples from the same dataset. It is mentioned averaging a set of observations reduces variance (James et al., 2013). Hence a natural way to reduce the variance and hence increase the prediction accuracy of a statistical learning method is to take many training sets from the population, build a separate prediction model using each training set, and average the resulting predictions. Using separate prediction results, $\hat{f}^1(x), \hat{f}^2(x), ..., \hat{f}^B(x)$ (x)), trained on B separate training sets and average them is logical to obtain a single low-variance statistical learning model. In reality, data is not redundant to use them separately for each training. The solution is to create separate datasets by using the available with the methodology called bootstrapping as B different bootstrapped training data sets are created by taking repeated samples from the (single) training data set. The result of averaging prediction results can be denoted as $\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$.

5.2.2.3 Random Forest Regression

The most dominant feature in decision trees is in the first parts of the tree. This causes dominant features to suppress other features. The idea to overcome this problem is using randomly selected features (typically the square-root of the original number of features) for each decision tree. Random forest algorithm is introduced as an enhancement of tree bagging (Breiman, 2001). This process is called as decorrelating the trees and assert this process makes the combined prediction outcomes of resulting trees less variable and hence steadier (James et al., 2013).

5.2.2.4 Extremely Randomized Trees Regression

It is stated the key differences of the algorithm with other tree-based ensemble methods splitting nodes by choosing cut-points fully at random and using the whole learning sample rather than a bootstrap replica to grow the trees (Geurts, Ernst & Wehenkel, 2006).

5.2.2.5 Boosting Regression

The idea of creating additional models to correct the errors of the previous models by fitting to the errors of the previous models is introduced (Kearns, 1988). Later, It is mentioned the concept of a weak learner as a produced hypothesis achieving slightly better performance than a random guess (Schapire, 1990). It is stated boosting uses stacking models back to back regarding the previous error by first creating a tree than creating another tree that tries to fit the residuals of the previous one in a stage-wise fashion until the predefined number of trees are reached. Note that boosting does not involve bootstrap sampling (James et al., 2013).

Adaboost A version of boosting called adaptive boosting which tries to create successive trees (weak learners) by sampling the wrongly predicted instances more and correctly predicted instances less is introduced. It also gives an adaptive coefficient to the trees' weights in the final model (strong learner) regarding their performance (Freund & Schapire, 1997).

Gradient Boosting Gradient boosting is introduced as an idea to use the gradient descent method in boosting as his method views function estimation/approximation from numerical optimization in function space rather than parameter optimization perspective (Friedman, 2000). The developed connection between the general boosting idea, stagewise additive expansions, and steepest-descent minimization is named a general gradient descent boosting paradigm. A month later a modification is added to the algorithm as stochastic gradient boosting by using subsamples of the training data for each learning iteration is introduced. Software implemented versions of gradient boosting are generally the later version (Friedman, 2002). Gradient Boosting and Adaboost were the champion algorithms before Xgboost, LightGBM and Catboost started become more popular.

XGBoost It is one of the most popular algorithms used in machine learning competitions. The algorithm first came as an R package with a 4-page article like a user guide, then with the success of the algorithm, the creators have published their methodology in a more conventional paper format. It is stated choosing the split point in a tree with a basic exact greedy algorithm takes too much time and the solution is to use a second-order gradient for approximately best-split point candidates (Chen & Guestrin, 2016). The algorithm also focuses on the sparsity problem by setting a default direction for each node.

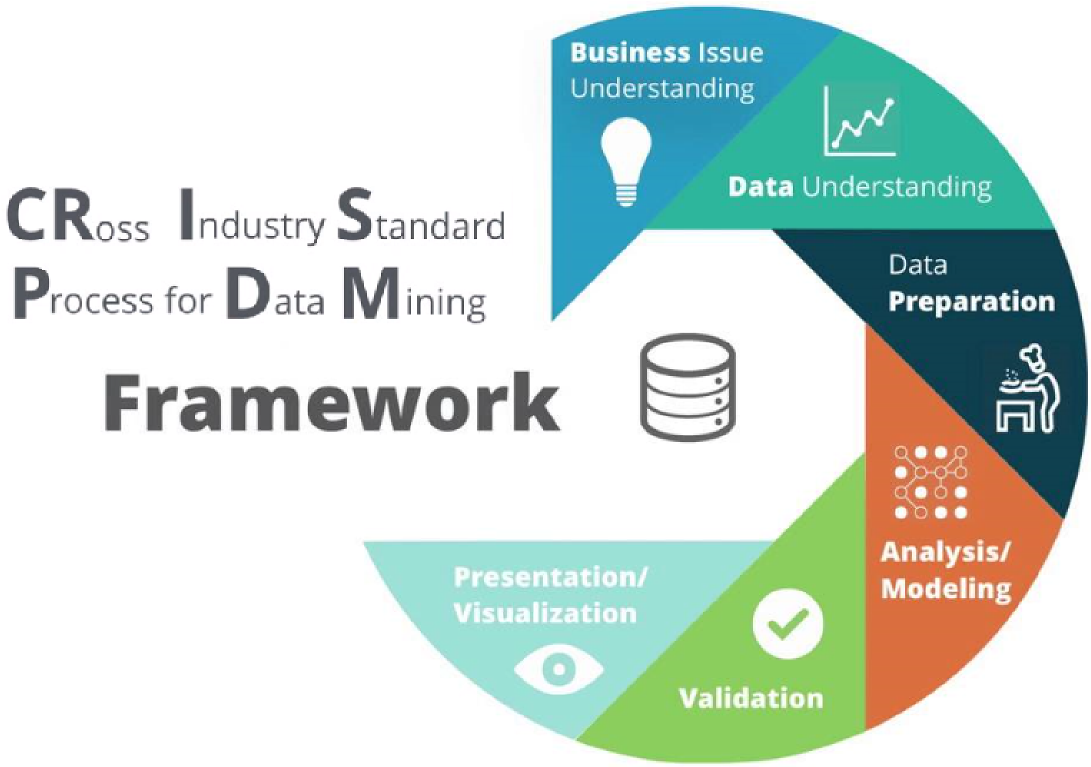
LightGBM Light Gradient Boosting Machine, was created by researchers of Microsoft Company with their claim improvement over XGBoost. It is state the algorithm's key difference is the way it creates splits with Gradient-based One-side Sampling (GOSS) and Exclusive Feature Bundling (EFB) methods as they named them (Ke, Meng, Finley, Wang, Chen, Ma, Ye & Liu, 2017). GOSS remarks that data points with greater gradients are more significant to decide splits. Data points with small gradients are already minimized, so data points with larger gradients should be the focus as the information gains achieved by splitting at that point are higher. Note that small gradient points are still kept and GOSS performs a random sampling of them and puts a constant weight value to keep the original data distribution while the focus is placed on the large gradient points. EFB prioritizes the exclusive features (features only rarely take non-zero values at the same time) as such features can be bundled or combined effectively, which reduces the width in a dataset.

CatBoost Yandex Company's researchers introduced the CatBoost algorithm. It is stated they introduce a new boosting scheme which fights biases with a dynamic boosting they call ordered boosting which helps to reduce overfitting and improves the quality of the model (Ostroumova, Gusev, Vorobev, Dorogush & Gulin, 2018). Catboost also provides support for categorical features inherently which means it does not require one hot encoding process the boosting algorithms that do not support.

6. ANALYSIS

During the analysis CRoss Industry Standard Process for Data Mining (CRISP-DM) Framework in Figure 6.1 is followed. It is one of the first frameworks known with the purpose of standardizing the stages of data science projects. It starts with understanding the problem, then continues with understanding the available data. After the data is understood, desired data is prepared by using the available data. Analysis and modelling stages follow the data preparation. In the end the results are checked by validation and the findings are visualized for them to be understood easily and then they are presented to the business owner.

Figure 6.1 Machine Learning Methods at a Glance



Source: (Medium.com, 2020)

6.1 Main Machine Learning Concepts

The business issue understanding stage is done by consulting to the experts and reading the related legislation and the history of the Turkish energy market to understand the dynamics of it which are explained in detail in Chapter 2. The business issue is to help market participants to optimize their profit by knowing the energy imbalance at the delivery time which we call net imbalance in this study, which is explained in detail in Chapter 3.

In a competitive market like the energy market, any previous knowledge about the prices that will be realized or the indicators that lead the prices are valuable. By predicting net imbalance, the market participants can have a better understanding of the market prices, thus can optimize their operations accordingly. The two possible benefits for predicting net imbalance value are to have a competitive edge in IM for energy s and the energy generation companies that give production commitment offers in case net imbalance is different from zero which is a perfect balance scenario that rarely happens. In IM, predicting the upcoming 24 hours is valuable since when the market is opened the closest hour that the trading can be made is the next hour, and the farthest hour for it is 24 hours later. BPM, the closest hour is after 9 hours since the commitment offers are given to TSO at 16.00 which means prediction is possible at 15.00. The next day starts at 00.00 meaning the closest prediction is ahead 9 hours for BPM. The last hour for the offers is 23.00 of the next day meaning, the farthest hour for prediction is 32 hours ahead. Predicting net imbalance for IM is the primary objective since it is the market that energy trading occurs. Besides, it is a lot easier compared to predicting net imbalance for BPM since the prediction hours are not as far as they are for BPM. Predicting net imbalance for BPM is the secondary objective since it is related to energy generation companies rather than energy s. Besides, it is challenging to predict net imbalance for distant hours.

6.2 Data Understanding Stage

The data understanding stage is done by checking the hourly available net imbalance value due to the dynamics of the Turkish energy market and checking the possible data that can be used to predict it. The data is formed around the hourly net imbalance values between 15.08.2015 and 05.05.2020. The data tested for predicting net imbalance can be grouped under four different categories. The raw data consists of 15120 instances with 18 features. Among these 18 features, 9 of them are time-based, 3 of them are price-based, 4 of them are forecast based, 1 of them is failures in the production plants and 1 is the target variable.

The time-based data are:

The operations of IM:

- Year
- Day of the year
- Month of the year
- Day of the month
- Day of the week
- Hour
- Night flag
- Daylight flag
- Twilight_dusk flag

The price-based data are:

- Difference between system marginal price (SMP) and market clearing price (MCP) of the 6 hours before the delivery time (LatestHour_SMP_MCP)
- PriceFeature1
- PriceFeature2

The forecast-based data are:

- TEIAS demand forecast error of the previous hour (TEIASDemForecastError)
- ForecastedFeature1
- ForecastedFeature2
- ForecastedFeature3

The failure-based data is:

- FailureFeature1

Figure 6.2 Machine Learning Methods at a Glance

MODELING	TEST	BACK-TEST
• Includes train and validation data • No seperate validation operation is done • Some models separate train and validation data inside and use automatically	• It is randomly seperated from modeling data before any operation to prevent bias • Good for markets that do not change in time	• Main testing data since it is like a simulation of the performance of the ML work after the deployment • Robust in markets that change in time

Figure 6.3 Machine Learning Methods at a Glance

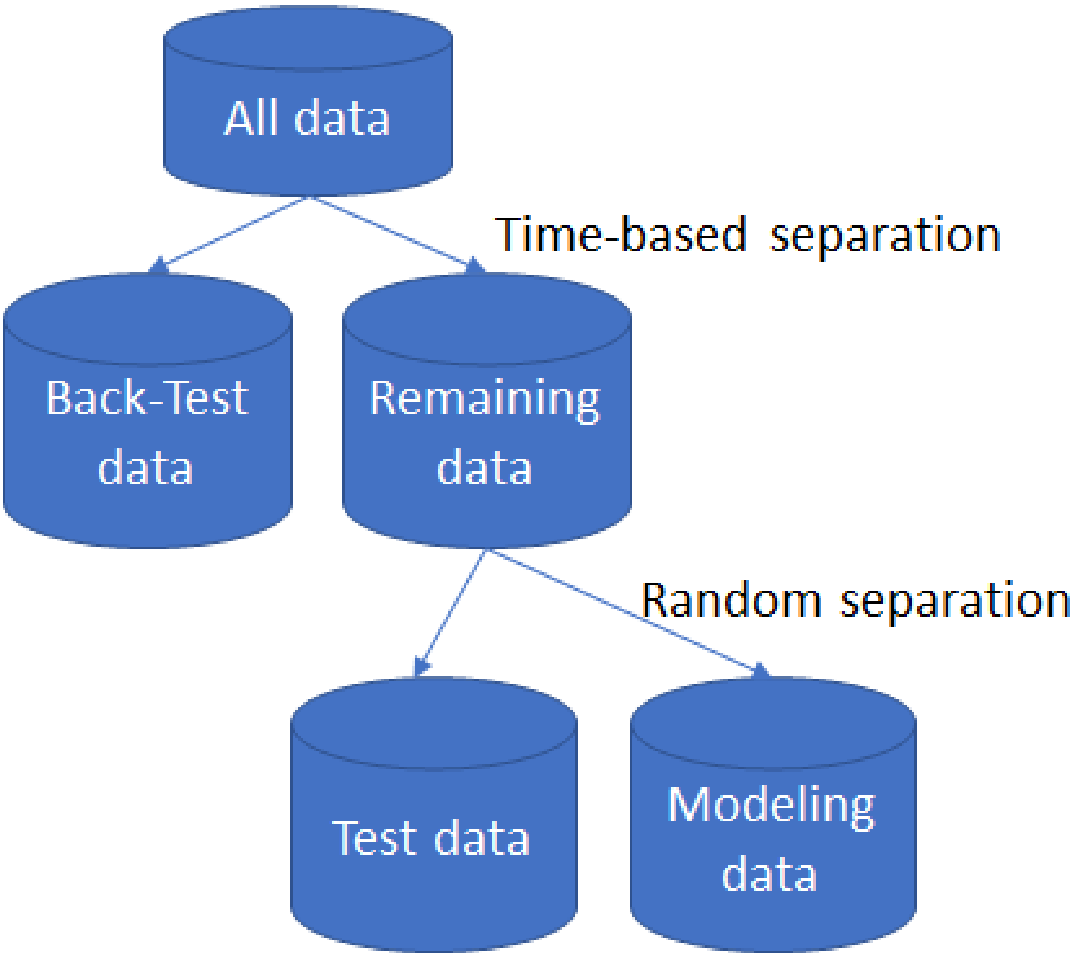


Table 6.1 Descriptive Statistics of (T+1) Target Feature

	Modeling Data	**Test Data**	**Back-Test Data**
count	9052	2405	3024
mean	0.46	0.40	-0.04
std	1.11	1.19	1.02
min	-4.68	-4.51	-5.20
25%	-0.10	-0.22	-0.63
50%	0.31	0.25	0.00
75%	1.12	1.08	0.52
max	5.62	5.86	3.94

Descriptive Statistics of the target feature, Output_f1_NetLoading, are shown in 6.1. Note that these statistics are similar for (T+1) up to (T+32) since they are simply the iterated versions of each other.

6.3 Data Preparation Stage

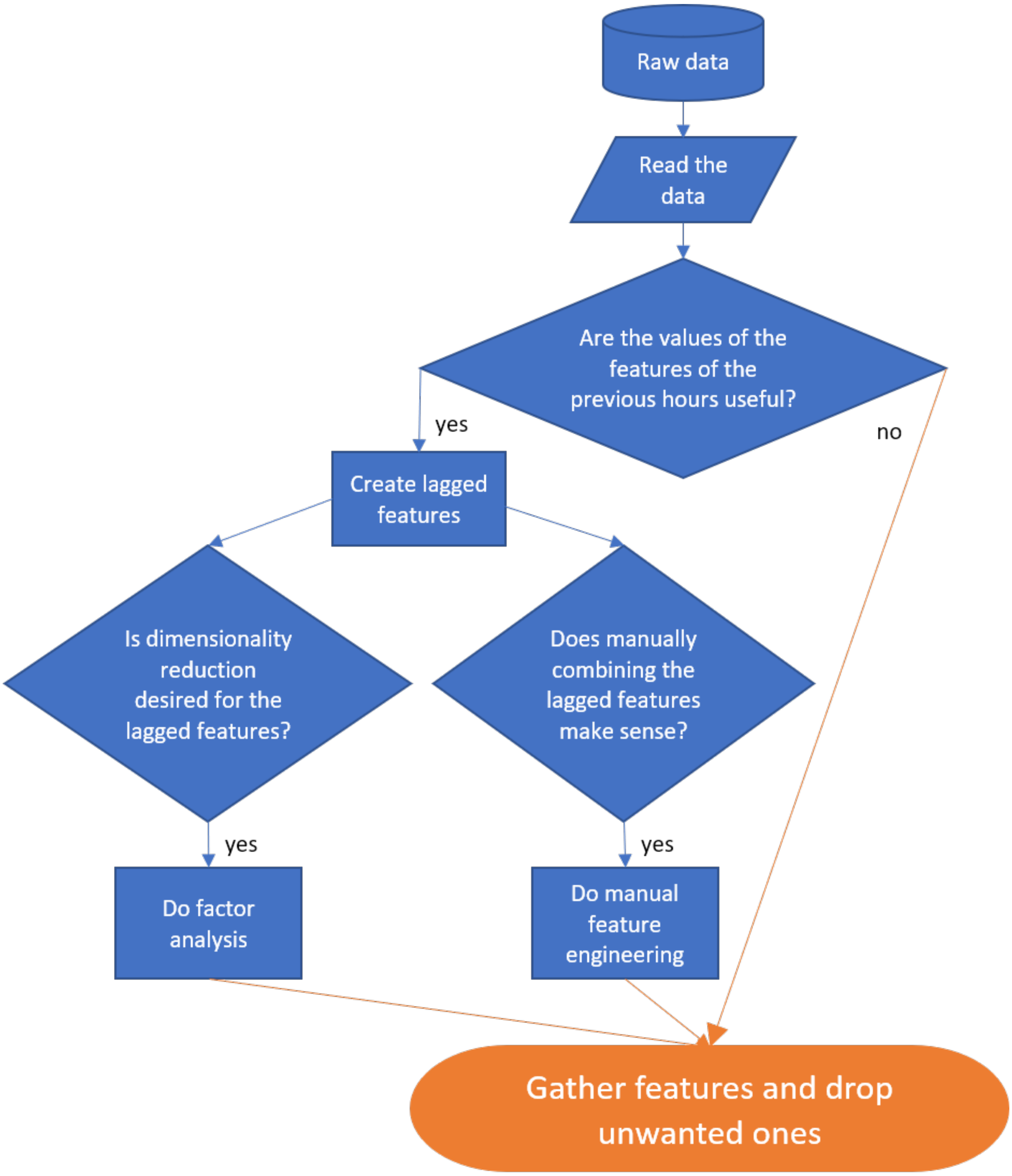
The data preparation stage is done using publicly available data and the resulting generated raw data consists of hourly values of net imbalance between 15.08.2015 and 05.05.2020. This raw data consists of 15120 instances with 18 features. The gathered data is enriched with feature engineering by creating new features from the previous 24-hour values of the target feature (here net imbalance) and two of the input features. Then, manual feature engineering is performed to create combined features from the existing features. After that, additional new features are created by factor analysis from some of the features in case the combinations of the features are more representative than their base features.

The data preparation steps are covered in detail in the later parts of this stage however the steps do not include data acquisition from multiple resources like the web and databases. Those steps are done by the business owner, so are not included in this study.

The part we can consider for this research the data preparation is the feature engineering part which is listed below. This part also includes the separation of time-independent test data which we also call back-test (back-test) since it is like a simulation and maintenance of real-time model evaluation in a deployed machine learning model. Then, modeling data (training-validation) and testing data are separated with random selection. The reason to separate the test data this early is to be sure there is no bias in test data due since both of them do not participate in any of the feature generations based on the features, outlier removal or feature selection operations.

We will mainly use the back-test data as the primary test data since it is completely independent of the train data as it is from another time interval. The test data is used as the secondary test data. It indicates the possible prediction success in a market whose dynamics do not change with time. All the operations mentioned after this point are done via Python programming language and its integrated development environment (IDE), Spyder. The laptop used has Intel Core i5-8300H 2.3 GHz processor and 16 GB RAM. Its operating system is Windows 10. The operations are done first showed in Flowchart 1 and then explained in details after it.

Figure 6.4 Data Preparation Stage



6.3.1 Step 1: Lagged data creations

- Lagged data creation operations on net imbalance:
  - Target is named as Output_f1_NetLoading to represent the net imbalance value that is tried to be predicted for the 1 hour ahead, future 1 (f1).

Lagged features of net imbalance are created by taking this output value as a base for the previous 24 hours. Alternatively stated, the net imbalance value of the previous hour is named as NetLoading_lag2 since it is 2 hours before the net imbalance value that is tried to be predicted. For Output_f1_NetLoading target, input features from NetLoading_lag2 to NetLoading_lag24 are created. NetLoading_lag1 is not available since there is an hour gap between the net imbalance to be announced by TSO.

  - Since we predict (T+1) up to (T+32), 32 data frames are created for each hour tried to be predicted. The targets are named as Output_fX_NetLoading to represent the exact amount of the distant hour that is wanted to be predicted. The lagged inputs are created by taking Output_fX_NetLoading as a reference similar to the above data creation. The 32 hours ahead target is named as Output_f32_NetLoading and the input features from NetLoading_lag33 to NetLoading_lag55 are created.

- Lagged data creation operations on LatestHour_SMP_MCP:
  - The closest time the SMP_MCP difference is available is 6 hours before the delivery time. That is why for Output_f1_NetLoading target, input features from LatestHour_SMP_MCP_lag6 to LatestHour_SMP_MCP_lag24 are created. Same feature creations are done for all 32 data frames in their respective lags.

- Lagged data creation operations on TEIASDemForecastError:
  - The closest time the TEIASDemForecastError is available is 2 hours before the delivery time. The business owner advised us to get only the 2, 3, 24 hours lagged versions of the variables. TEIASDemForecastError_lag2, TEIASDemForecastError_lag3, TEIASDemForecastError_lag24 features are created. The same feature creations are done for all 32 data frames in their respective lags. The shape of the resulting data is (15120, 62).

6.3.2 Step 2: Back-test, test, modelling data seperations:

- The first 72 hours of the data points are dropped since creating lagged features results in NA values in these first 3 days due to not having the previous values for the oldest data points.

- All of the 2020 data is kept for back-test. It makes 3024 data points.
- The test data and the modeling data are split in 0.2 to 0.8 ratios resulting in 2405 data points for the test and 9619 data points for the modeling.

6.3.3 Step 3: Feature creation with factor analysis

"(Madnani, 2020)'s factor_analyzer" python library is used for the following operations.

- Factor analysis applied to modeling data:
  - For Output_f1_NetLoading, the features from NetLoading_lag3 to NetLoading_lag24 are used to create additional features since the created features can be more representative compared to the features create them. NetLoading_lag2 is not included for the factor analysis since it is alone is the most related feature with the target, so including adding any part of it to the newly created features is not desired. The same feature creations are done for all 32 data frames in their respective lags.
  - For Output_f1_NetLoading, the features from LatestHour_SMP_MCP_lag6 to LatestHour_SMP_MCP_lag24 are used to create additional features. The same feature creations are done for all 32 data frames in their respective lags.
- Factor analysis applied to back-test and test data:
  - - The transformations saved while applying factor analysis to the modeling data are applied to back-test and test data. . This step is performed especially by first applying the factor analysis and learning factor analysis parameters from modeling data rather than mixing modeling and test/back-test data together. This prevented any bias in testing success as we did not feed test data insight to modeling by mistake. This operation is emphasized especially since it is a common mistake to process modeling data and test data together thus feeding information about test data to modeling.

6.3.4 Step 4: Manual feature creations

- NetLoading_lag_c3-7 is created by summing values of the features from NetLoading_lag3 to NetLoading_lag7. Same is done for hours between 8-13, 14-19, 20-24. Also pair additions are done for 3-4, 4-5, 5-6.
- ii. Pairwise subtractions are done for 3-2, 4-3, 5-4, 6-5. Net imbalance values from 10 to 23 are dropped. The idea behind all these feature creations is to give the feature selection step as much as logical options to select from as the change between the hours in the successive hours can be a logical indicator of the net imbalance trend.

6.3.5 Step 5: gathering the created data

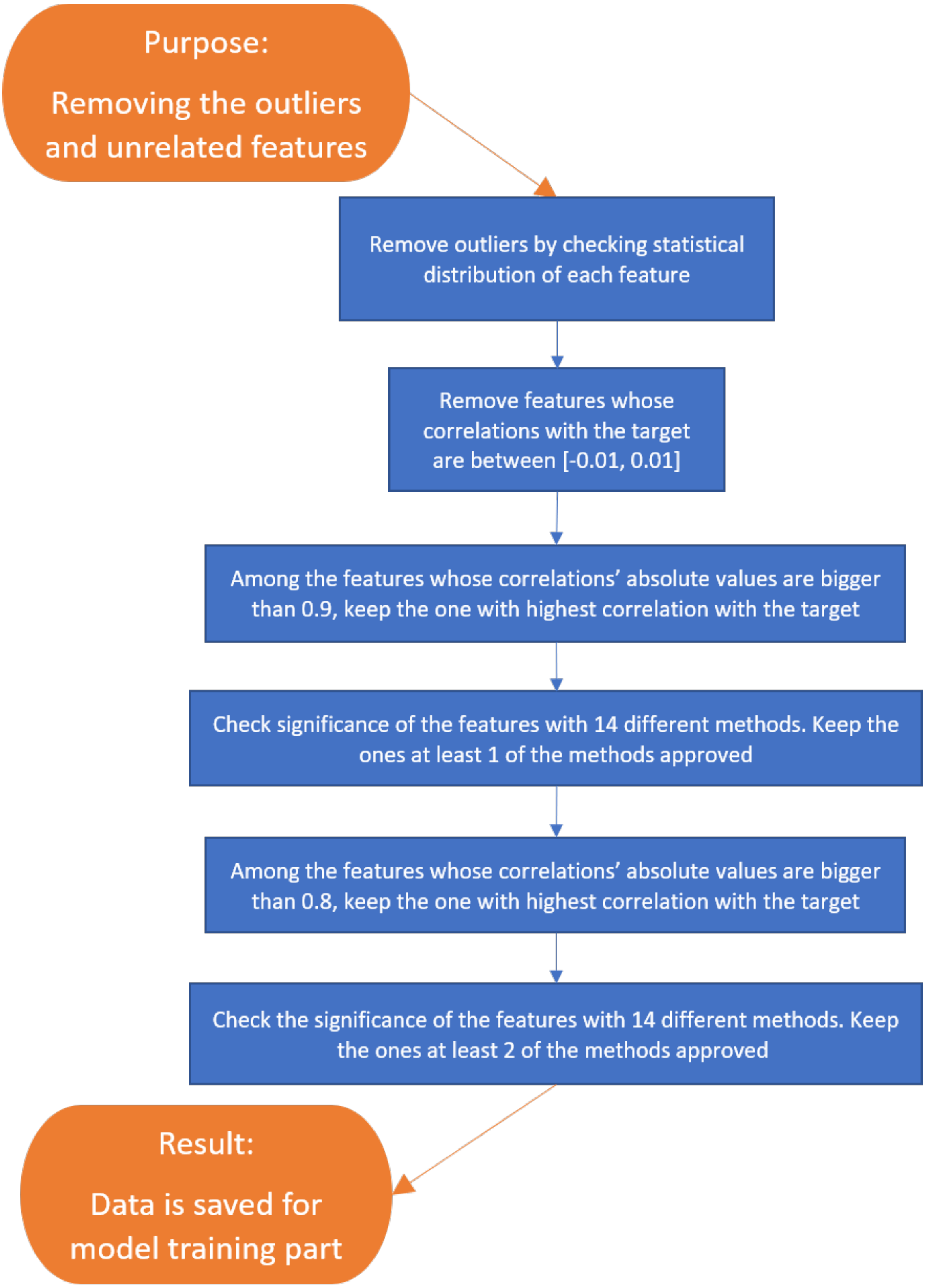
- All the created features are added to the data frames except the intentionally dropped ones. The resulting shapes of the data frames for modeling, test, and back-test parts are (9619;70), (2405;70), and (3024;70) respectively. All the 32-hour data groups have the same compositions meaning we have 3 * 32 = 96 total data frames at this point.

6.4 Analysis Stage

In this stage, the data with a big pool of features created in the data preparation stage are analyzed and the data is tried to be made ready for modeling stages. That is why first the data is cleaned from outlier values. Second, the number of highly correlated inputs are reduced. Last, the big pool of the features created in the data preparation step is needed to be reduced.

The operations are done are first showed in Flowchart 2 and then explained in detail after it.

Figure 6.5 Analysis Stage



6.4.1 Step 1: Outliers are eliminated

- The outliers for each feature are dropped in if the sigma distance to the average of mean and median is greater than 4.5 sigma.
- The point of taking the average of mean and median instead of using only one of them is trying to benefit from both of their strengths. Taking the median value as a reference and measuring the distance between the data point and the median value is strong to prevent the effect of extreme values if the data is skewed, however, it may cause to remove real data points with a little bit extreme values to be removed. That is why the average of mean and median are used together with the goal of being on the safe side against skewed feature shapes while not removing a little bit of extreme data.
- The point of taking a conservative 4.5 sigma is because clearing the outliers in this study only aims to drop the shock values in the system or potential data errors. Other than that all the data is real, so it is not desired to drop data points even if they are a little bit extreme. That is why the standard 3 sigma distance outlier filter approach is not chosen. Besides, we have 69 input features at this stage, and we are dropping data points for each feature. If we have used 3 sigma distance which means dropping 0.3% when the data is in perfect normal distribution shape and a worst-case scenario happens where there is no intersection of the outlier data points for 69 input features, 0.3% * 69 = 20.7 of the data would be gone. Since the real-world data is not perfectly normally distributed it could be even higher.
- When outlier elimination with 3 sigma distance is applied to the modeling data with Output_f1_NetLoading target variable, the modeling data drops from 9619 instances to 6234 instances. Losing 35% of the modeling data was not preferred so 4.5 sigma distance is used and the resulting data frame has 9052 instances meaning dropping only 6% of the instances which is completely okay. From this point on the number of data instances for each target hour change since the outlier elimination operation is done regarding the internal dynamics of each data frame. That is why, after this point data shapes will be given for the only Output_f1_NetLoading to prevent a potential crowd of 32 data frames just by giving data shape of the closest ahead hour.

6.4.2 Step 2: Feature elimination with correlations phase 1

- Inputs with [-0.01, 0.01] correlation with the target are dropped.

- Inputs whose absolute correlation value among themselves are more than 0.9 are handled by keeping the one which has the highest correlation with the target and dropping the others. 0.9 is a very conservative threshold compared to the common practice of using 0.7 or 0.5 thresholds in risk operations in the banking sector. This conservative threshold is selected to drop features gradually to be on the safe side. The resulting shape of the modeling data is (9052;57). Note that after this point the shapes of the data for (T+1) up to (T+32) vary according to the operations done based on their statistical characteristics. Here the shape of the data for target variable Output_f1_NetLoading (T+1) is given for simplicity instead of writing all 32 of them.

6.4.3 Step 3: Feature selection phase 1

Each of the below methods is asked to select 20 most important features among the remaining 57 features. This 20 is selected as a common-sense since a higher threshold causes to the elimination of almost no features and a lower threshold results in a dramatic decrease in the number of features instead of gradual feature elimination approach which is tried to be followed in this study

- Statistics-based methods:
  - Based on correlation
  - Based on F statistic with sklearn.feature_selection.f_regression() function
  - Based on mutual information (MI) with sklearn.feature_selection.mutual_info_regression()
- Recursive function elimination: It is applied with sklearn.feature_selection.RFE() function. It selects features with the desired machine learning in a backward feature selection way. It is used via the following machine learning models:
  - stochastic gradient regression model with linear_model.SGDRegressor() function
  - elastic net regression with linear_model.ElasticNetCV() function

  - lasso lars regression with linear_model.LassoLarsCV() function
  - orthogonal matching persuit regression with OrthogonalMatchingPursuitCV() function
  - adaboost regression with sklearn.ensemble.AdaBoostRegressor() function
  - gradient boosting regression with sklearn.ensemble. GradientBoostingRegressor() function
  - extra trees regression with sklearn.ensemble.ExtraTreesRegressor() function

- Selecting features based on importance weights: It is applied with sklearn.feature_selection.SelectFromModel. It can only be applied to the machine learning models with feature importance attribute. It is performed with the following machine learning models:
  - Random forest regression with sklearn.ensemble.RandomForestRegressor() function
  - Ridge regression with sklearn.linear_model.RidgeCV() function
  - Light gradient boosting regression with lightgbm.LGBMRegressor() function

- The features which can not get approval from any of these methods are dropped. This approval by any method is the lowest possible threshold. It is again selected to drop the features gradually to not miss any important feature. The resulting shape of the modeling data is (9052;45):

6.4.4 Step 4: Feature elimination with correlations phase 2

Inputs whose absolute correlation value among themselves are more than 0.8 are handled by keeping the one which has the highest correlation with the target and dropping the others. 0.8 is the most conservative correlation keep in the applications I have experienced. It is kept this high to leave as many feature options to the feature selection parts while preventing overwhelming them with unnecessarily correlated features. The resulting shape of the modeling data is (9052;33).

6.4.5 Step 5: Feature selection phase 2

- Each of the methods is asked to select 15 most important feature among the remaining 33 features:
- The methods used in feature selection phase 1 are also used here. The features which can not get approval from at least 2 methods are dropped. The resulting shape of the modeling data is (9052;30):

6.5 Modeling Stage

In this stage, the 11 different machine learning models are used for modeling purposes. A naïve model which assumes the target is the last available net imbalance as the prediction is used. In other words; for the prediction of next hour target, Output_f1_NetLoading is assumed to be equal to NetLoading_lag2. This model is used as the baseline for prediction success. Additionally, these 11 models are used together thanks to stacking and voting ensemble methods. Thus, 14 models are used.

The modeling stage is performed in three modeling phases. In the first modeling phase, the machine learning models with non-optimized features and a fixed number of features are used. In the second modeling phase, the machine learning models with optimized features by random search and a fixed number of features are used. In the third modeling phase, the machine learning models with optimized features by random search and optimized number of features are used.

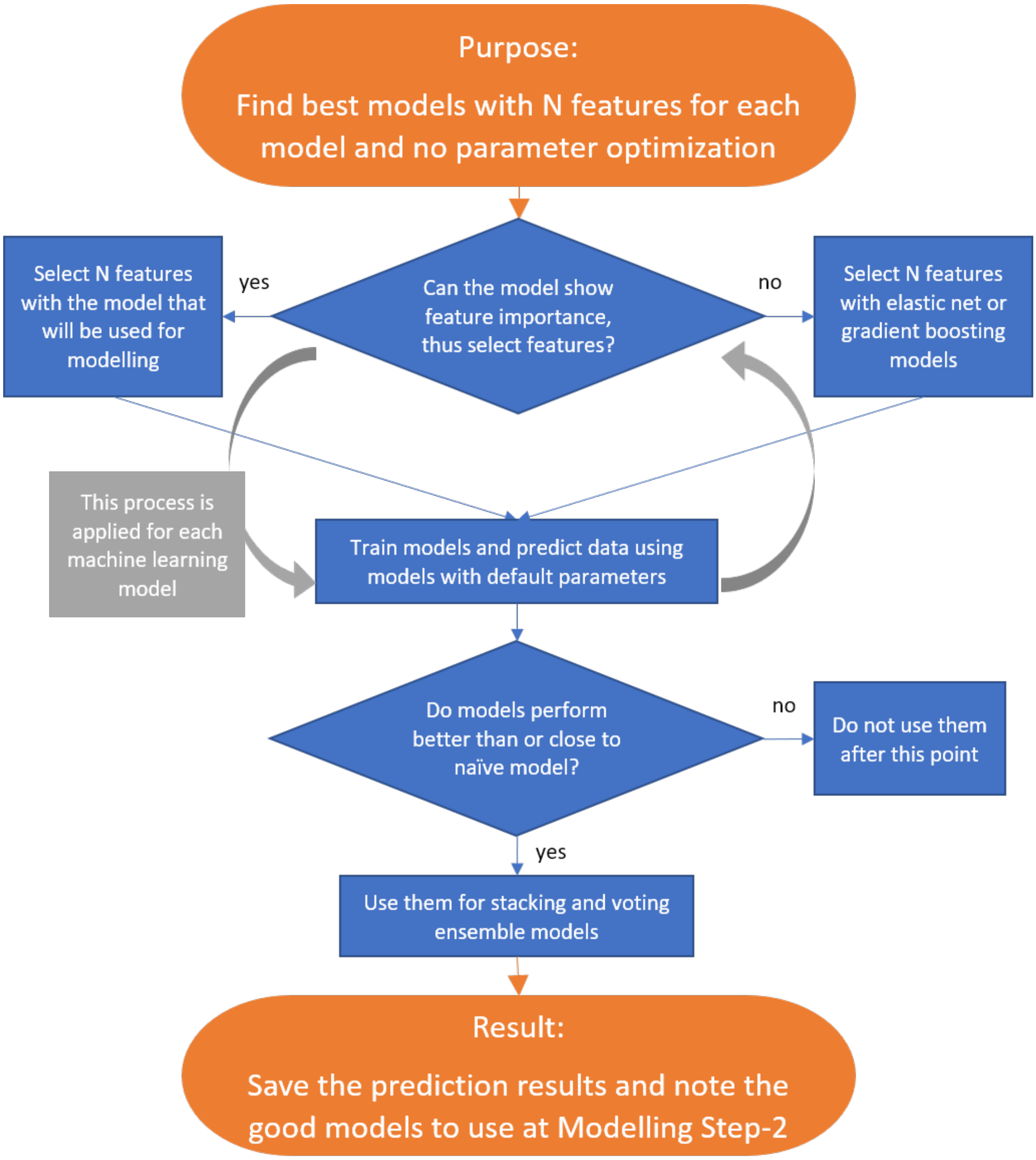
In all the three modeling phases, five steps are followed. The first step is applying standard scaling to the input variables. The second step is the final feature selection before training the models. The third step is training machine learning models with the selected features. The fourth step is checking the performances of the trained models by comparing them to the naïve model, then deciding not to use the bad performing machine learning models for the two ensemble methods stacking and voting. The fifth step is to apply stacking and voting ensemble models with the remaining machine learning models.

Note that, in all the phases these five steps are followed for predicting the next 32 hours in a loop approach meaning net imbalance values for the next 32 hours are predicted in each modeling phase. The operations are done are first showed in flowcharts and then explained in details after them.

6.5.1 Step 1: Standardization

- A standardization of input features is done over the remaining 30 features by sklearn.preprocessing.StandardScaler() function which applies z transformation.
- This operation is done on the modeling data and the learned transformation (Scaler_ss) is applied to back-test and test data. As stated before, to prevent any bias all the data manipulation related operations are performed on modeling data and then copied operations are applied to back-test and test data.

Figure 6.6 Modeling Stage Phase 1; N Features and no Parameter Optimization



6.5.2 Step 2: Feature selection phase for modeling phases 1-2; feature selection results in pre-decided fix number of features

- In this step, the features standardized are used and the shape of the modeling data change from (9052;30) to (9052;8). In this modeling stage, we wanted to see the model performances with a fixed number of features. 8 is chosen after trying some numbers such as 6,8,10 since it seemed like a good choice. It may look like a bold shot, however, it is not that important since for the modeling phase 3 we will let the models chose the best number of features.

- The selection is done with sklearn.feature_selection.RFE() using a single machine learning model. It is tried to be used the same model for feature selection with the respective model in the modeling phase. As an example; while creating a feature selector Selector=RFE(estimator=regressor_fsel) for a modelling regressor reg=regressor_model.fit(X=X_tr,y=y_tr), regressor_fsel=LinearRegression() and regressor_model=LinearRegression() are used together. The purpose of this is to select features that the model itself think is the best.
- Back-test and test data are filtered according to the selected features by the modeling data.

6.5.3 Step 3: Modeling phase 1; with 8 features and default model parameters

- 11 different machine learning models are trained:
  - Linear regression
  - Lasso
  - Elastic net
  - K nearest neigborhood (KNN)
  - Random forest
  - Extra trees
  - Ada boosting
  - Gradient boosting
  - Xg boosting
  - Light gradient boosting
  - Cat boosting

- A naïve model mentioned before which gets the closest net imbalance as the predicted value is added for baseline comparison of modeling success of the other models.
- 7 of the 11 models perform better than or close to naïve model in back-test data. That is why they are chosen for stacking and voting ensemble models. These models are.
  - Linear regression
  - Lasso
  - Elastic net
  - Extra trees
  - Gradient boosting
  - Light gradient boosting
  - Cat boosting
- 7 good performing models are chosen to be used in stacking and voting ensemble models. The stacking model is trained with sklearn.ensemble.StackingRegressor() function and voting model is trained with sklearn.ensemble.VotingRegressor() function.
- Coefficient of determination (R-square regression score function) results are obtained with sklearn.metrics.r2_score(y_true, y_pred) function , MAE results are obtained with sklearn.metrics.mean_absolute_error(y_true, y_pred) function and MAPE is calculated as shown in Figure 6.7.
- R-square, MAE and MAPE performance results are shown in tables from 6.2 up to 6.7. Results for back-test data and test data are labeled from a1 to a3 and b1 to b3 respectively to distinguish them easier.

Figure 6.7 MAPE Calculation

```python
def percentage_error(actual, predicted):
    res = np.empty(actual.shape)
    for j in range(actual.shape[0]):
        if actual[j] != 0:
            res[j] = (actual[j] - predicted[j]) / actual[j]
        else:
            res[j] = predicted[j] / np.mean(actual)
    return res

def mean_absolute_percentage_error(y_true, y_pred):
    return np.mean(np.abs(percentage_error(np.asarray(y_true), np.asarray(y_pred)))) * 100
```

Table 6.2 Modeling phase 1 – a1. BACK-TEST Data Performance - R-square

	linear	**lasso**	**elastic**	knn	rf	et	ab	gb	xgb	lgbm	catb	naive	stack	voting
nl_f_01	0.585	0.549	0.594	0.382	0.522	0.547	0.302	0.564	0.42	0.497	0.527	0.53	0.569	0.598
nl_f_02	0.429	0.401	0.453	0.243	0.352	0.411	0.259	0.426	0.18	0.334	0.36	0.318	0.418	0.457
nl_f_03	0.286	0.298	0.342	0.028	0.207	0.289	0.17	0.284	0.037	0.016	0.226	0.124	0.297	0.339
nl_f_04	0.216	0.235	0.27	-0.05	0.138	0.169	0.036	0.168	-0.09	-0.01	0.019	-0.04	0.133	0.219
nl_f_05	0.169	0.186	0.221	-0.1	0.088	0.108	-0.06	0.13	-0.19	-0.07	0.01	-0.18	0.104	0.175
nl_f_06	0.123	0.154	0.189	-0.09	0.063	0.082	-0.19	0.083	-0.34	-0.05	0.002	-0.3	0.062	0.15
nl_f_07	0.135	0.135	0.169	-0.1	0.03	0.065	-0.14	0.029	-0.32	-0.09	-0.05	-0.38	0.04	0.132
nl_f_08	0.111	0.121	0.155	-0.13	-0.01	0.024	-0.34	0.037	-0.61	-0.06	-0.09	-0.44	-0	0.128
nl_f_09	0.095	0.106	0.139	-0.18	-0.03	-0.03	-0.23	0.047	-0.26	-0.15	-0.13	-0.49	-0.04	0.09
nl_f_10	0.081	0.099	0.127	-0.18	-0.08	-0.04	-0.19	0.009	-0.47	-0.07	-0.18	-0.54	-0.07	0.111
nl_f_11	0.067	0.096	0.122	-0.15	-0.1	-0.03	-0.14	0.009	-0.24	-0.08	-0.21	-0.58	-0.09	0.106
nl_f_12	0.053	0.094	0.124	-0.16	-0.07	-0.02	-0.31	-0	-0.35	-0.02	-0.14	-0.61	-0.05	0.108
nl_f_13	0.046	0.091	0.119	-0.22	-0.05	-0.02	-0.35	0.021	-0.53	-0.08	-0.13	-0.61	-0.03	0.108
nl_f_14	0.044	0.092	0.118	-0.21	-0.04	0.003	-0.39	0.029	-0.41	-0.01	-0.09	-0.63	-0.01	0.115
nl_f_15	0.048	0.088	0.114	-0.25	-0.04	-0.01	-0.4	0.016	-0.36	-0.06	-0.14	-0.65	-0	0.111
nl_f_16	0.043	0.09	0.112	-0.27	-0.05	-0.02	-0.31	0.017	-0.26	-0.09	-0.11	-0.64	-0.01	0.111
nl_f_17	0.03	0.1	0.121	-0.26	-0.03	0.001	-0.39	0.01	-0.53	-0.16	-0.11	-0.62	-0.01	0.108
nl_f_18	0.047	0.095	0.119	-0.26	-0.06	-0.01	-0.24	0.006	-0.49	-0.12	-0.1	-0.58	-0	0.11
nl_f_19	0.015	0.103	0.122	-0.27	-0.08	-0	-0.2	0.01	-0.52	-0.11	-0.11	-0.53	-0.01	0.109
nl_f_20	0.021	0.105	0.122	-0.25	-0.13	-0.03	-0.39	-0.03	-0.37	-0.07	-0.11	-0.48	-0.02	0.105
nl_f_21	0.034	0.106	0.125	-0.23	-0.16	-0.06	-0.25	-0.04	-0.48	-0.08	-0.13	-0.45	-0.04	0.104
nl_f_22	0.016	0.105	0.123	-0.25	-0.14	-0.08	-0.27	-0.04	-0.62	-0.24	-0.15	-0.43	-0.05	0.11
nl_f_23	0.037	0.103	0.124	-0.2	-0.1	-0.02	-0.37	-0.04	-0.55	-0.13	-0.13	-0.41	-0.04	0.117
nl_f_24	0.015	0.093	0.107	-0.19	-0.09	-0.14	-0.26	-0.03	-0.51	-0.2	-0.14	-0.51	-0.05	0.104
nl_f_25	-0.04	0.084	0.09	-0.23	-0.1	-0.04	-0.15	0.003	-0.26	-0.2	-0.15	-0.58	-0.06	0.117
nl_f_26	-0.04	0.077	0.084	-0.21	-0.11	-0.07	-0.32	-0.01	-0.85	-0.22	-0.17	-0.65	-0.05	0.113
nl_f_27	-0.06	0.071	0.08	-0.27	-0.13	-0.07	-0.2	0.003	-0.43	-0.19	-0.15	-0.72	-0.05	0.114
nl_f_28	-0.06	0.069	0.078	-0.24	-0.14	-0.09	-0.24	-0.03	-0.18	-0.24	-0.2	-0.77	-0.07	0.096
nl_f_29	-0.06	0.069	0.075	-0.24	-0.11	-0.08	-0.28	-0.05	-0.46	-0.15	-0.22	-0.83	-0.09	0.095
nl_f_30	-0.06	0.068	0.073	-0.2	-0.12	-0.11	-0.35	-0.03	-0.73	-0.23	-0.24	-0.88	-0.12	0.101
nl_f_31	-0.06	0.069	0.073	-0.26	-0.12	-0.07	-0.31	-0.03	-0.51	-0.14	-0.2	-0.9	-0.12	0.1
nl_f_32	-0.04	0.071	0.074	-0.2	-0.12	-0.08	-0.13	-0.03	-0.53	-0.23	-0.14	-0.92	-0.1	0.105

Table 6.3 Modeling phase 1 – a2. BACK-TEST Data Performance - MAE

	linear	lasso	elastic	knn	rf	et	ab	gb	xgb	lgbm	catb	naive	stack	voting
nl_f_01	0.488	0.517	0.482	0.606	0.527	0.511	0.663	0.5	0.593	0.543	0.522	0.509	0.499	0.481
nl_f_02	0.574	0.596	0.561	0.675	0.622	0.592	0.678	0.59	0.717	0.636	0.612	0.62	0.586	0.568
nl_f_03	0.647	0.641	0.617	0.766	0.686	0.654	0.709	0.654	0.768	0.78	0.675	0.715	0.647	0.627
nl_f_04	0.683	0.665	0.651	0.806	0.727	0.714	0.776	0.71	0.831	0.79	0.768	0.784	0.724	0.687
nl_f_05	0.702	0.686	0.672	0.82	0.748	0.741	0.813	0.722	0.877	0.807	0.773	0.837	0.737	0.704
nl_f_06	0.719	0.699	0.684	0.824	0.755	0.748	0.859	0.739	0.919	0.793	0.782	0.878	0.757	0.714
nl_f_07	0.714	0.707	0.692	0.812	0.771	0.754	0.843	0.767	0.907	0.82	0.807	0.908	0.766	0.726
nl_f_08	0.724	0.712	0.698	0.837	0.79	0.773	0.934	0.767	1.014	0.803	0.827	0.928	0.785	0.732
nl_f_09	0.73	0.717	0.704	0.858	0.796	0.797	0.878	0.762	0.884	0.848	0.835	0.943	0.8	0.745
nl_f_10	0.736	0.718	0.707	0.861	0.817	0.803	0.862	0.781	0.959	0.811	0.855	0.955	0.812	0.73
nl_f_11	0.741	0.718	0.708	0.846	0.824	0.793	0.839	0.777	0.885	0.812	0.878	0.971	0.824	0.732
nl_f_12	0.743	0.72	0.709	0.846	0.817	0.786	0.908	0.78	0.923	0.8	0.841	0.983	0.798	0.724
nl_f_13	0.746	0.722	0.71	0.863	0.802	0.789	0.923	0.772	0.986	0.815	0.841	0.987	0.794	0.724
nl_f_14	0.748	0.722	0.711	0.854	0.801	0.776	0.936	0.766	0.936	0.781	0.82	0.994	0.784	0.72
nl_f_15	0.748	0.725	0.714	0.865	0.796	0.78	0.942	0.771	0.923	0.815	0.844	1.003	0.783	0.724
nl_f_16	0.749	0.724	0.715	0.872	0.793	0.79	0.923	0.771	0.895	0.822	0.833	1.002	0.786	0.723
nl_f_17	0.751	0.719	0.709	0.87	0.786	0.775	0.94	0.775	0.964	0.849	0.826	0.994	0.782	0.724
nl_f_18	0.745	0.72	0.71	0.885	0.803	0.786	0.882	0.776	0.978	0.831	0.827	0.976	0.786	0.724
nl_f_19	0.76	0.718	0.709	0.889	0.81	0.785	0.862	0.783	0.975	0.835	0.831	0.955	0.79	0.725
nl_f_20	0.754	0.715	0.708	0.876	0.827	0.791	0.94	0.787	0.922	0.814	0.834	0.935	0.794	0.726
nl_f_21	0.75	0.715	0.707	0.865	0.839	0.802	0.884	0.792	0.957	0.823	0.843	0.921	0.806	0.729
nl_f_22	0.757	0.716	0.708	0.869	0.831	0.813	0.901	0.792	0.98	0.885	0.845	0.913	0.807	0.727
nl_f_23	0.748	0.716	0.707	0.858	0.82	0.799	0.939	0.796	0.968	0.849	0.852	0.906	0.808	0.724
nl_f_24	0.762	0.719	0.717	0.863	0.828	0.85	0.895	0.805	0.974	0.871	0.852	0.951	0.816	0.729
nl_f_25	0.786	0.72	0.724	0.866	0.825	0.811	0.853	0.79	0.889	0.873	0.849	0.984	0.813	0.726
nl_f_26	0.79	0.722	0.727	0.858	0.835	0.818	0.916	0.794	1.072	0.888	0.868	1.01	0.816	0.728
nl_f_27	0.803	0.724	0.728	0.883	0.848	0.817	0.872	0.79	0.958	0.864	0.852	1.036	0.811	0.73
nl_f_28	0.799	0.724	0.729	0.875	0.849	0.827	0.882	0.804	0.868	0.893	0.869	1.054	0.819	0.736
nl_f_29	0.802	0.724	0.729	0.874	0.839	0.819	0.906	0.809	0.956	0.852	0.88	1.069	0.826	0.737
nl_f_30	0.801	0.725	0.73	0.863	0.834	0.832	0.939	0.801	1.037	0.884	0.893	1.08	0.842	0.738
nl_f_31	0.8	0.724	0.73	0.893	0.843	0.819	0.913	0.8	0.958	0.843	0.87	1.079	0.838	0.736
nl_f_32	0.789	0.724	0.73	0.863	0.838	0.822	0.841	0.798	0.971	0.88	0.846	1.081	0.826	0.735

Table 6.4 Modeling phase 1 – a3. BACK-TEST Data Performance - MAPE

	linear	lasso	elastic	knn	rf	et	ab	gb	xgb	lgbm	catb	naive	stack	voting
nl_f_01	289	270	249	400	318	346	409	307	473	385	368	368	313	282
nl_f_02	322	294	282	444	366	364	458	368	523	412	405	454	359	327
nl_f_03	369	284	285	517	387	371	389	412	519	553	418	523	366	347
nl_f_04	424	301	325	493	442	441	498	441	516	480	480	674	422	383
nl_f_05	405	286	312	567	470	433	493	442	527	534	497	722	445	391
nl_f_06	346	256	279	552	446	427	490	445	641	489	498	728	473	388
nl_f_07	309	255	250	499	450	451	471	443	566	502	516	706	490	390
nl_f_08	329	230	241	474	406	407	629	418	643	437	469	717	445	359
nl_f_09	327	232	236	498	396	420	508	415	510	480	510	711	473	377
nl_f_10	327	237	250	529	416	434	482	468	609	394	550	686	487	358
nl_f_11	332	239	254	480	495	412	467	457	516	471	520	678	480	344
nl_f_12	346	244	258	509	527	437	516	482	558	465	601	617	527	370
nl_f_13	369	252	274	633	474	399	606	502	682	505	650	628	599	390
nl_f_14	386	253	280	591	619	456	662	509	588	511	599	579	556	393
nl_f_15	381	251	277	628	456	382	636	495	461	521	578	594	548	380
nl_f_16	382	246	278	589	473	431	629	498	539	502	643	604	526	391
nl_f_17	364	242	268	592	561	456	678	505	742	510	621	622	541	399
nl_f_18	361	247	270	659	485	416	630	499	605	489	659	562	527	391
nl_f_19	379	245	272	543	492	400	611	522	703	535	674	564	545	395
nl_f_20	375	232	266	612	612	422	711	484	580	470	657	572	601	388
nl_f_21	407	236	274	715	702	485	656	565	712	539	691	561	644	429
nl_f_22	408	237	276	794	555	531	610	553	607	584	695	576	647	422
nl_f_23	395	230	264	717	504	487	681	538	632	556	709	609	602	401
nl_f_24	408	224	276	696	479	519	573	513	542	595	600	613	551	389
nl_f_25	412	218	274	509	505	463	538	505	529	471	564	630	527	358
nl_f_26	411	220	273	539	537	462	659	492	847	596	654	614	554	362
nl_f_27	418	221	269	589	566	448	596	529	643	564	545	720	479	348
nl_f_28	409	222	271	597	583	487	629	541	527	581	551	797	473	358
nl_f_29	403	222	269	524	578	478	642	513	602	565	590	822	488	361
nl_f_30	403	222	269	512	544	486	721	537	726	606	648	857	591	394
nl_f_31	406	223	270	526	557	541	630	563	567	514	625	840	548	390
nl_f_32	389	223	271	499	582	484	520	550	597	637	594	830	561	375

Table 6.5 Modeling phase 1 – b1. TEST Data Performance - R-square

	linear	**lasso**	**elastic**	**knn**	**rf**	**et**	**ab**	**gb**	**xgb**	**lgbm**	**catb**	**naive**	**stack**	**voting**
nl_f_01	0.615	0.553	0.587	0.609	0.617	0.629	0.503	0.632	0.624	0.659	0.656	0.496	0.652	0.637
nl_f_02	0.491	0.428	0.471	0.546	0.546	0.576	0.351	0.524	0.601	0.578	0.58	0.3	0.582	0.534
nl_f_03	0.413	0.348	0.391	0.509	0.5	0.546	0.293	0.461	0.538	0.549	0.542	0.159	0.547	0.474
nl_f_04	0.362	0.302	0.339	0.458	0.481	0.569	0.22	0.41	0.553	0.541	0.49	0.021	0.518	0.422
nl_f_05	0.327	0.268	0.303	0.434	0.518	0.567	0.169	0.377	0.549	0.508	0.471	-0.08	0.502	0.392
nl_f_06	0.28	0.243	0.276	0.42	0.531	0.571	0.136	0.366	0.535	0.502	0.543	-0.14	0.574	0.406
nl_f_07	0.251	0.218	0.25	0.414	0.53	0.578	0.163	0.352	0.558	0.513	0.547	-0.24	0.579	0.395
nl_f_08	0.264	0.205	0.232	0.396	0.545	0.585	0.164	0.36	0.576	0.502	0.555	-0.3	0.603	0.395
nl_f_09	0.253	0.192	0.219	0.399	0.543	0.664	0.147	0.337	0.577	0.528	0.532	-0.36	0.564	0.373
nl_f_10	0.231	0.183	0.211	0.372	0.573	0.655	0.153	0.355	0.568	0.499	0.542	-0.37	0.584	0.38
nl_f_11	0.221	0.178	0.206	0.344	0.565	0.657	0.151	0.341	0.578	0.511	0.569	-0.44	0.626	0.386
nl_f_12	0.233	0.178	0.206	0.333	0.576	0.642	0.153	0.345	0.543	0.531	0.57	-0.44	0.64	0.387
nl_f_13	0.23	0.178	0.211	0.429	0.581	0.662	0.17	0.335	0.55	0.531	0.568	-0.46	0.635	0.385
nl_f_14	0.231	0.176	0.21	0.407	0.562	0.644	0.161	0.341	0.546	0.511	0.56	-0.45	0.631	0.384
nl_f_15	0.233	0.173	0.206	0.383	0.582	0.648	0.167	0.338	0.501	0.521	0.566	-0.47	0.635	0.384
nl_f_16	0.235	0.171	0.206	0.396	0.58	0.648	0.163	0.341	0.557	0.519	0.562	-0.47	0.625	0.384
nl_f_17	0.243	0.178	0.214	0.376	0.518	0.574	0.174	0.335	0.58	0.521	0.527	-0.44	0.569	0.372
nl_f_18	0.24	0.173	0.21	0.358	0.557	0.619	0.159	0.345	0.516	0.519	0.553	-0.46	0.603	0.382
nl_f_19	0.244	0.175	0.213	0.373	0.552	0.608	0.171	0.337	0.508	0.523	0.562	-0.49	0.605	0.386
nl_f_20	0.248	0.177	0.215	0.322	0.512	0.59	0.162	0.346	0.474	0.518	0.552	-0.46	0.581	0.386
nl_f_21	0.245	0.174	0.213	0.284	0.509	0.587	0.167	0.341	0.537	0.515	0.561	-0.44	0.583	0.385
nl_f_22	0.248	0.174	0.213	0.396	0.533	0.585	0.158	0.352	0.53	0.526	0.55	-0.4	0.582	0.385
nl_f_23	0.246	0.171	0.209	0.295	0.537	0.583	0.146	0.342	0.552	0.503	0.551	-0.36	0.583	0.383
nl_f_24	0.226	0.155	0.194	0.28	0.526	0.617	0.162	0.321	0.543	0.506	0.499	-0.43	0.535	0.361
nl_f_25	0.216	0.148	0.184	0.333	0.536	0.602	0.153	0.341	0.554	0.52	0.553	-0.49	0.593	0.374
nl_f_26	0.208	0.144	0.176	0.287	0.533	0.62	0.146	0.339	0.528	0.517	0.545	-0.55	0.593	0.371
nl_f_27	0.19	0.14	0.168	0.365	0.549	0.625	0.133	0.351	0.588	0.517	0.555	-0.66	0.606	0.371
nl_f_28	0.187	0.138	0.166	0.294	0.558	0.644	0.139	0.344	0.597	0.516	0.58	-0.71	0.638	0.379
nl_f_29	0.182	0.137	0.164	0.301	0.552	0.636	0.159	0.349	0.568	0.515	0.585	-0.7	0.644	0.384
nl_f_30	0.183	0.138	0.163	0.283	0.556	0.64	0.135	0.34	0.56	0.529	0.571	-0.74	0.622	0.378
nl_f_31	0.186	0.139	0.165	0.322	0.559	0.634	0.135	0.333	0.581	0.524	0.576	-0.74	0.62	0.38
nl_f_32	0.19	0.14	0.166	0.349	0.561	0.643	0.137	0.329	0.574	0.542	0.574	-0.77	0.623	0.378

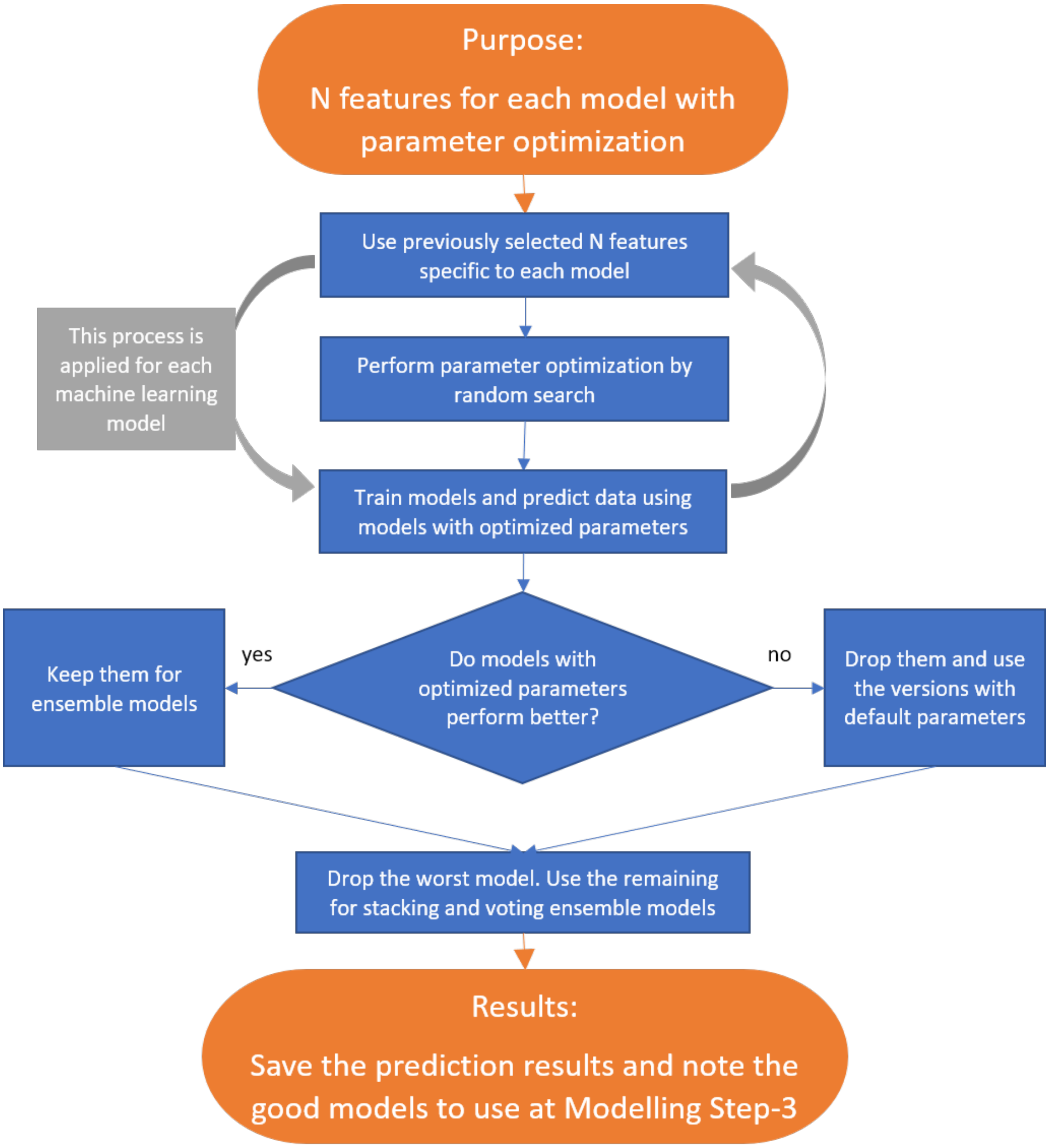
Table 6.6 Modeling phase 1 – b2. TEST Data Performance - MAE

	linear	**lasso**	**elastic**	**knn**	**rf**	**et**	**ab**	**gb**	**xgb**	**lgbm**	**catb**	**naive**	**stack**	**voting**
nl_f_01	0.537	0.571	0.548	0.523	0.526	0.509	0.636	0.519	0.518	0.5	0.507	0.583	0.507	0.516
nl_f_02	0.623	0.653	0.627	0.57	0.574	0.547	0.739	0.597	0.548	0.565	0.562	0.699	0.554	0.588
nl_f_03	0.672	0.705	0.682	0.596	0.603	0.573	0.76	0.645	0.587	0.591	0.593	0.772	0.579	0.632
nl_f_04	0.707	0.736	0.716	0.635	0.615	0.559	0.814	0.682	0.581	0.597	0.633	0.849	0.605	0.67
nl_f_05	0.729	0.759	0.739	0.645	0.593	0.559	0.857	0.702	0.585	0.622	0.646	0.902	0.617	0.69
nl_f_06	0.758	0.775	0.757	0.655	0.584	0.552	0.874	0.712	0.585	0.622	0.597	0.934	0.562	0.684
nl_f_07	0.773	0.788	0.771	0.653	0.582	0.547	0.85	0.721	0.583	0.619	0.593	0.978	0.554	0.693
nl_f_08	0.771	0.795	0.779	0.677	0.579	0.544	0.857	0.716	0.556	0.616	0.588	0.998	0.538	0.694
nl_f_09	0.775	0.803	0.787	0.674	0.581	0.485	0.862	0.729	0.584	0.611	0.608	1.027	0.574	0.707
nl_f_10	0.782	0.809	0.792	0.683	0.556	0.486	0.858	0.727	0.571	0.624	0.604	1.028	0.555	0.708
nl_f_11	0.793	0.813	0.797	0.712	0.561	0.489	0.861	0.732	0.56	0.615	0.577	1.06	0.515	0.704
nl_f_12	0.788	0.813	0.797	0.708	0.551	0.493	0.858	0.727	0.585	0.61	0.573	1.062	0.5	0.701
nl_f_13	0.793	0.814	0.797	0.648	0.546	0.475	0.848	0.731	0.581	0.609	0.575	1.07	0.503	0.703
nl_f_14	0.792	0.815	0.797	0.665	0.562	0.492	0.856	0.731	0.587	0.614	0.581	1.071	0.507	0.703
nl_f_15	0.791	0.817	0.799	0.681	0.543	0.488	0.853	0.732	0.603	0.612	0.577	1.079	0.501	0.703
nl_f_16	0.79	0.818	0.8	0.675	0.545	0.492	0.846	0.729	0.578	0.606	0.577	1.083	0.511	0.702
nl_f_17	0.786	0.816	0.797	0.69	0.593	0.548	0.852	0.73	0.566	0.613	0.605	1.07	0.557	0.71
nl_f_18	0.788	0.819	0.8	0.698	0.56	0.513	0.855	0.725	0.603	0.612	0.581	1.081	0.527	0.703
nl_f_19	0.784	0.817	0.798	0.688	0.563	0.516	0.844	0.728	0.607	0.609	0.576	1.086	0.528	0.7
nl_f_20	0.783	0.815	0.797	0.716	0.592	0.529	0.85	0.724	0.616	0.61	0.579	1.065	0.545	0.7
nl_f_21	0.787	0.817	0.799	0.747	0.599	0.533	0.856	0.729	0.583	0.605	0.572	1.066	0.543	0.702
nl_f_22	0.787	0.818	0.799	0.67	0.577	0.535	0.852	0.723	0.595	0.602	0.58	1.036	0.546	0.702
nl_f_23	0.788	0.82	0.801	0.734	0.574	0.533	0.863	0.727	0.58	0.605	0.58	1.015	0.542	0.701
nl_f_24	0.797	0.826	0.808	0.743	0.579	0.5	0.851	0.737	0.584	0.606	0.606	1.047	0.57	0.714
nl_f_25	0.797	0.83	0.812	0.705	0.579	0.515	0.85	0.73	0.571	0.601	0.586	1.07	0.538	0.706
nl_f_26	0.801	0.832	0.815	0.744	0.577	0.505	0.858	0.732	0.578	0.605	0.588	1.1	0.54	0.71
nl_f_27	0.81	0.835	0.82	0.691	0.572	0.504	0.871	0.729	0.551	0.607	0.587	1.139	0.535	0.712
nl_f_28	0.812	0.837	0.822	0.731	0.56	0.493	0.863	0.731	0.553	0.607	0.571	1.161	0.509	0.708
nl_f_29	0.813	0.837	0.823	0.715	0.569	0.5	0.86	0.73	0.565	0.609	0.569	1.176	0.506	0.707
nl_f_30	0.813	0.837	0.823	0.735	0.569	0.501	0.872	0.736	0.578	0.606	0.579	1.189	0.523	0.711
nl_f_31	0.813	0.836	0.823	0.715	0.567	0.503	0.863	0.737	0.56	0.613	0.576	1.189	0.525	0.709
nl_f_32	0.811	0.836	0.822	0.706	0.568	0.5	0.864	0.74	0.562	0.601	0.581	1.189	0.523	0.711

Table 6.7 Modeling phase 1 – b3. TEST Data Performance - MAPE

	linear	lasso	elastic	knn	rf	et	ab	gb	xgb	lgbm	catb	naive	stack	voting
nl_f_01	164	147	146	162	163	156	198	156	141	144	156	189	149	148
nl_f_02	183	171	170	163	183	162	248	175	163	156	168	227	157	165
nl_f_03	190	176	178	169	173	170	235	189	198	183	179	243	170	177
nl_f_04	214	195	200	207	179	171	310	210	192	174	199	289	188	195
nl_f_05	226	206	212	199	178	163	335	226	190	185	205	348	190	208
nl_f_06	225	205	213	219	170	164	319	219	165	202	184	343	171	201
nl_f_07	233	209	216	208	165	162	272	212	190	195	181	367	168	201
nl_f_08	227	205	212	245	166	161	269	207	167	185	187	357	166	197
nl_f_09	224	209	214	247	179	140	265	220	176	192	201	347	187	205
nl_f_10	234	210	216	233	166	137	255	221	185	186	194	363	174	204
nl_f_11	230	210	216	222	170	145	270	211	173	188	182	336	158	195
nl_f_12	224	210	214	194	159	149	267	208	184	177	181	335	154	194
nl_f_13	225	210	212	198	161	140	250	208	167	184	186	367	153	194
nl_f_14	225	208	211	210	168	145	280	211	183	183	188	352	158	195
nl_f_15	221	206	209	203	170	154	260	207	195	187	189	324	157	192
nl_f_16	217	204	207	181	156	145	253	200	188	176	179	319	147	189
nl_f_17	216	208	208	215	168	157	263	201	192	182	184	312	156	191
nl_f_18	216	207	209	209	162	153	264	198	197	179	177	342	151	188
nl_f_19	215	206	207	197	173	144	236	201	189	175	171	340	148	187
nl_f_20	212	205	206	199	168	145	260	198	196	192	175	332	155	187
nl_f_21	216	205	205	226	174	143	286	198	187	180	163	373	149	186
nl_f_22	214	205	204	183	178	148	247	199	209	187	170	360	152	187
nl_f_23	221	206	207	216	172	150	261	199	191	174	173	359	150	187
nl_f_24	224	206	210	222	175	146	249	206	172	179	184	378	178	192
nl_f_25	218	205	207	201	172	172	240	201	196	181	181	376	159	188
nl_f_26	218	205	208	234	175	158	252	206	186	180	182	362	166	191
nl_f_27	221	206	210	198	174	156	255	207	182	182	179	365	155	191
nl_f_28	226	207	212	202	161	161	252	209	184	178	189	339	162	193
nl_f_29	223	207	211	194	177	154	259	207	184	176	183	363	160	193
nl_f_30	222	207	211	231	169	154	269	205	199	180	187	358	160	193
nl_f_31	222	207	212	232	164	159	260	204	185	190	182	371	161	191
nl_f_32	222	207	212	199	168	155	274	207	183	179	192	381	157	193

Figure 6.8 Modeling Stage Phase 2; N Features and Optimized Model Parameters with Random Search



6.5.4 Step 4: Modeling phase 2; with 8 features and optimized model parameters with random search

- Features selected at step 2 are used.
- Among the remaining 7 models, 1 is linear regression which does

not need any parameter optimization. That is why parameter optimization with random search is performed for the models with sklearn.model_selection.RandomizedSearchCV() function.

- Among those 6 models, LightGBM is dropped since its performance according to R-square value is worse than the naïve model both in the modeling phase 1 and modeling phase 2. It was kept to see whether it will be improved with the random search, however, no improvement is observed. With the linear regression model remaining 6 good performing models are chosen to be used in stacking and voting ensemble models. Elasticnet and gradient boosting methods do not improve with parameter optimization so for ensemble models their default versions are used which are shown in Figure 6.9.

- R-square, MAE and MAPE performance results are shown in tables from 6.8 up to 6.13.

Figure 6.9 Models used in Voting Ensemble Regressor in Modeling Stage Phase 2

```
    estimators = [
     ('linear'     , LinearRegression()           )
    ,('lasso'      , l_e['lasso'][i]              )
    ,('elasticnet', ElasticNet(alpha=0.1)         )
    ,('et'         , l_e['et'   ][i]              )
    ,('gb'         , GradientBoostingRegressor() )
    ,('catb'       , l_e['catb' ][i]              )
    ]
    regressor_model = VotingRegressor(
         estimators      = estimators
        )
```

Table 6.8 Modeling phase 2 – a1. BACK-TEST Data Performance - R-square

	linear	lasso	elastic	et	gb	lgbm	catb	naive	stack	voting
nl_f_01	0.585	0.579	0.578	0.555	0.536	0.492	0.542	0.53	0.581	0.603
nl_f_02	0.429	0.453	0.421	0.43	0.343	0.318	0.296	0.318	0.394	0.463
nl_f_03	0.286	0.344	0.307	0.322	0.198	0.002	0.155	0.124	0.257	0.345
nl_f_04	0.216	0.257	0.224	0.247	0.008	0.018	-0.07	-0.04	0.066	0.238
nl_f_05	0.169	0.206	0.177	0.198	-0.03	-0.06	-0.11	-0.18	0.008	0.186
nl_f_06	0.123	0.175	0.147	0.166	-0.04	-0.06	-0.08	-0.3	-0.03	0.172
nl_f_07	0.135	0.153	0.124	0.155	-0.13	-0.1	-0.13	-0.38	-0.08	0.155
nl_f_08	0.111	0.131	0.123	0.145	-0.1	-0.1	-0.23	-0.44	-0.16	0.15
nl_f_09	0.095	0.115	0.107	0.119	-0.14	-0.09	-0.23	-0.49	-0.16	0.117
nl_f_10	0.081	0.098	0.09	0.116	-0.25	-0.11	-0.35	-0.54	-0.25	0.12
nl_f_11	0.067	0.092	0.084	0.11	-0.23	-0.07	-0.32	-0.58	-0.23	0.131
nl_f_12	0.053	0.092	0.089	0.11	-0.19	-0.04	-0.26	-0.61	-0.18	0.122
nl_f_13	0.046	0.103	0.064	0.106	-0.21	-0.1	-0.25	-0.61	-0.19	0.125
nl_f_14	0.044	0.103	0.061	0.113	-0.23	-0.04	-0.23	-0.63	-0.16	0.13
nl_f_15	0.048	0.077	0.056	0.104	-0.22	-0.06	-0.2	-0.65	-0.13	0.13
nl_f_16	0.043	0.088	0.049	0.098	-0.18	-0.12	-0.26	-0.64	-0.2	0.124
nl_f_17	0.03	0.111	0.065	0.099	-0.19	-0.15	-0.2	-0.62	-0.13	0.13
nl_f_18	0.047	0.078	0.056	0.095	-0.18	-0.12	-0.23	-0.58	-0.16	0.12
nl_f_19	0.015	0.082	0.054	0.099	-0.15	-0.08	-0.21	-0.53	-0.14	0.133
nl_f_20	0.021	0.102	0.056	0.095	-0.2	-0.16	-0.24	-0.48	-0.17	0.124
nl_f_21	0.034	0.106	0.054	0.097	-0.27	-0.14	-0.27	-0.45	-0.21	0.126
nl_f_22	0.016	0.104	0.053	0.09	-0.25	-0.27	-0.29	-0.43	-0.22	0.121
nl_f_23	0.037	0.107	0.055	0.079	-0.18	-0.17	-0.3	-0.41	-0.22	0.131
nl_f_24	0.015	0.07	0.028	0.065	-0.14	-0.2	-0.21	-0.51	-0.15	0.13
nl_f_25	-0.04	0.049	0.011	0.082	-0.22	-0.25	-0.29	-0.58	-0.2	0.129
nl_f_26	-0.04	0.045	0.009	0.089	-0.22	-0.17	-0.35	-0.65	-0.26	0.12
nl_f_27	-0.06	0.037	0.014	0.084	-0.34	-0.2	-0.29	-0.72	-0.22	0.124
nl_f_28	-0.06	0.038	0.031	0.078	-0.27	-0.21	-0.35	-0.77	-0.27	0.114
nl_f_29	-0.06	0.024	0.027	0.098	-0.3	-0.27	-0.34	-0.83	-0.25	0.112
nl_f_30	-0.06	0.025	0.018	0.099	-0.35	-0.17	-0.4	-0.88	-0.31	0.11
nl_f_31	-0.06	0.027	0.02	0.089	-0.3	-0.16	-0.36	-0.9	-0.28	0.111
nl_f_32	-0.04	0.025	0.019	0.1	-0.28	-0.27	-0.33	-0.92	-0.24	0.113

Table 6.9 Modeling phase 2 – a2. BACK-TEST Data Performance - MAE

	linear	lasso	elastic	et	gb	lgbm	catb	naive	stack	voting
nl_f_01	0.488	0.493	0.489	0.506	0.518	0.549	0.514	0.509	0.493	0.478
nl_f_02	0.574	0.561	0.577	0.573	0.621	0.646	0.645	0.62	0.6	0.563
nl_f_03	0.647	0.616	0.635	0.627	0.689	0.79	0.708	0.715	0.665	0.622
nl_f_04	0.683	0.662	0.679	0.66	0.767	0.782	0.798	0.784	0.747	0.676
nl_f_05	0.702	0.683	0.699	0.681	0.775	0.813	0.81	0.837	0.769	0.696
nl_f_06	0.719	0.697	0.711	0.694	0.792	0.791	0.813	0.878	0.795	0.702
nl_f_07	0.714	0.706	0.721	0.701	0.821	0.818	0.845	0.908	0.826	0.712
nl_f_08	0.724	0.716	0.72	0.707	0.835	0.824	0.876	0.928	0.852	0.718
nl_f_09	0.73	0.721	0.725	0.72	0.842	0.821	0.879	0.943	0.854	0.728
nl_f_10	0.736	0.726	0.73	0.72	0.898	0.823	0.92	0.955	0.889	0.722
nl_f_11	0.741	0.729	0.731	0.723	0.885	0.817	0.914	0.971	0.885	0.717
nl_f_12	0.743	0.727	0.729	0.722	0.859	0.797	0.886	0.983	0.86	0.716
nl_f_13	0.746	0.72	0.736	0.726	0.878	0.821	0.888	0.987	0.867	0.714
nl_f_14	0.748	0.721	0.739	0.721	0.885	0.79	0.877	0.994	0.852	0.711
nl_f_15	0.748	0.735	0.744	0.727	0.874	0.816	0.866	1.003	0.841	0.71
nl_f_16	0.749	0.731	0.747	0.729	0.867	0.837	0.892	1.002	0.869	0.713
nl_f_17	0.751	0.717	0.735	0.729	0.853	0.839	0.867	0.994	0.842	0.71
nl_f_18	0.745	0.734	0.741	0.731	0.866	0.828	0.869	0.976	0.846	0.715
nl_f_19	0.76	0.732	0.74	0.732	0.843	0.817	0.872	0.955	0.845	0.711
nl_f_20	0.754	0.721	0.738	0.732	0.87	0.853	0.881	0.935	0.855	0.713
nl_f_21	0.75	0.72	0.739	0.728	0.889	0.842	0.898	0.921	0.874	0.713
nl_f_22	0.757	0.722	0.739	0.731	0.885	0.896	0.905	0.913	0.879	0.717
nl_f_23	0.748	0.72	0.739	0.739	0.849	0.853	0.92	0.906	0.893	0.713
nl_f_24	0.762	0.741	0.755	0.741	0.85	0.873	0.884	0.951	0.859	0.719
nl_f_25	0.786	0.749	0.763	0.734	0.882	0.881	0.905	0.984	0.877	0.719
nl_f_26	0.79	0.752	0.766	0.733	0.873	0.865	0.932	1.01	0.902	0.723
nl_f_27	0.803	0.755	0.763	0.732	0.92	0.877	0.904	1.036	0.88	0.723
nl_f_28	0.799	0.754	0.757	0.74	0.898	0.887	0.921	1.054	0.894	0.725
nl_f_29	0.802	0.76	0.759	0.73	0.909	0.904	0.919	1.069	0.892	0.728
nl_f_30	0.801	0.76	0.763	0.731	0.925	0.861	0.939	1.08	0.912	0.728
nl_f_31	0.8	0.758	0.761	0.738	0.902	0.862	0.934	1.079	0.907	0.728
nl_f_32	0.789	0.76	0.761	0.731	0.892	0.893	0.911	1.081	0.884	0.728

Table 6.10 Modeling phase 2 – a3. BACK-TEST Data Performance - MAPE

	linear	lasso	elastic	et	gb	lgbm	catb	naive	stack	voting
nl_f_01	289	266	288	248	345	380	372	368	319	274
nl_f_02	322	322	338	274	421	407	462	454	389	312
nl_f_03	369	323	355	278	440	503	466	523	409	334
nl_f_04	424	385	416	271	496	490	498	674	437	375
nl_f_05	405	372	398	282	557	575	557	722	491	366
nl_f_06	346	346	369	259	511	473	554	728	521	351
nl_f_07	309	313	343	248	470	504	547	706	517	352
nl_f_08	329	316	329	242	539	468	552	717	524	340
nl_f_09	327	314	327	257	544	447	564	711	531	346
nl_f_10	327	305	317	248	650	516	630	686	590	332
nl_f_11	332	311	314	262	517	471	537	678	510	320
nl_f_12	346	315	316	264	608	460	699	617	658	343
nl_f_13	369	322	355	278	648	483	688	628	650	351
nl_f_14	386	330	368	271	669	537	702	579	658	361
nl_f_15	381	338	363	278	721	447	628	594	590	347
nl_f_16	382	328	365	277	673	495	702	604	663	359
nl_f_17	364	312	348	278	684	522	734	622	682	361
nl_f_18	361	331	353	297	762	553	660	562	618	346
nl_f_19	379	332	360	302	576	520	763	564	712	368
nl_f_20	375	317	353	282	700	566	743	572	692	345
nl_f_21	407	323	379	264	811	562	806	561	754	379
nl_f_22	408	327	376	267	680	538	803	576	747	384
nl_f_23	395	313	370	295	708	592	812	609	757	370
nl_f_24	408	340	390	292	544	583	555	613	515	314
nl_f_25	412	341	384	251	671	525	609	630	569	324
nl_f_26	411	345	382	252	585	654	714	614	663	335
nl_f_27	418	344	371	243	623	569	582	720	545	320
nl_f_28	409	344	350	285	543	530	596	797	559	334
nl_f_29	403	345	350	256	671	594	595	822	553	333
nl_f_30	403	345	351	260	737	517	702	857	660	351
nl_f_31	406	344	349	296	689	625	693	840	653	353
nl_f_32	389	350	351	258	665	678	635	830	596	342

Table 6.11 Modeling phase 2 – b1. TEST Data Performance - R-square

	linear	lasso	elastic	et	gb	lgbm	catb	naive	stack	voting
nl_f_01	0.615	0.576	0.612	0.551	0.64	0.649	0.656	0.496	0.644	0.626
nl_f_02	0.491	0.467	0.493	0.44	0.558	0.557	0.582	0.3	0.577	0.528
nl_f_03	0.413	0.391	0.417	0.367	0.498	0.489	0.548	0.159	0.535	0.464
nl_f_04	0.362	0.341	0.361	0.318	0.446	0.468	0.491	0.021	0.481	0.411
nl_f_05	0.327	0.309	0.326	0.283	0.428	0.437	0.476	-0.08	0.464	0.382
nl_f_06	0.28	0.284	0.299	0.263	0.504	0.456	0.589	-0.14	0.578	0.392
nl_f_07	0.251	0.262	0.272	0.246	0.504	0.458	0.585	-0.24	0.576	0.375
nl_f_08	0.264	0.245	0.246	0.242	0.499	0.447	0.598	-0.3	0.59	0.373
nl_f_09	0.253	0.233	0.233	0.226	0.488	0.458	0.559	-0.36	0.551	0.348
nl_f_10	0.231	0.225	0.226	0.22	0.515	0.441	0.586	-0.37	0.574	0.355
nl_f_11	0.221	0.218	0.221	0.215	0.535	0.448	0.605	-0.44	0.596	0.356
nl_f_12	0.233	0.217	0.221	0.211	0.539	0.449	0.61	-0.44	0.602	0.362
nl_f_13	0.23	0.219	0.233	0.213	0.523	0.447	0.606	-0.46	0.595	0.358
nl_f_14	0.231	0.219	0.233	0.21	0.525	0.44	0.603	-0.45	0.593	0.358
nl_f_15	0.233	0.218	0.23	0.207	0.525	0.46	0.606	-0.47	0.594	0.357
nl_f_16	0.235	0.214	0.231	0.204	0.53	0.44	0.603	-0.47	0.592	0.357
nl_f_17	0.243	0.224	0.241	0.201	0.506	0.447	0.574	-0.44	0.558	0.349
nl_f_18	0.24	0.221	0.238	0.199	0.54	0.436	0.588	-0.46	0.575	0.356
nl_f_19	0.244	0.225	0.243	0.204	0.526	0.455	0.597	-0.49	0.583	0.359
nl_f_20	0.248	0.224	0.245	0.2	0.526	0.45	0.587	-0.46	0.571	0.358
nl_f_21	0.245	0.218	0.241	0.217	0.521	0.448	0.592	-0.44	0.578	0.357
nl_f_22	0.248	0.219	0.243	0.217	0.534	0.456	0.592	-0.4	0.576	0.36
nl_f_23	0.246	0.216	0.237	0.2	0.519	0.419	0.578	-0.36	0.565	0.354
nl_f_24	0.226	0.202	0.223	0.205	0.474	0.434	0.536	-0.43	0.521	0.322
nl_f_25	0.216	0.195	0.213	0.199	0.509	0.462	0.591	-0.49	0.576	0.343
nl_f_26	0.208	0.194	0.206	0.207	0.525	0.451	0.584	-0.55	0.569	0.338
nl_f_27	0.19	0.184	0.198	0.204	0.514	0.458	0.589	-0.66	0.574	0.337
nl_f_28	0.187	0.183	0.183	0.178	0.556	0.452	0.617	-0.71	0.603	0.341
nl_f_29	0.182	0.177	0.183	0.211	0.569	0.454	0.614	-0.7	0.602	0.344
nl_f_30	0.183	0.178	0.18	0.213	0.535	0.445	0.605	-0.74	0.595	0.342
nl_f_31	0.186	0.18	0.183	0.187	0.538	0.451	0.6	-0.74	0.592	0.34
nl_f_32	0.19	0.181	0.184	0.213	0.535	0.477	0.607	-0.77	0.596	0.339

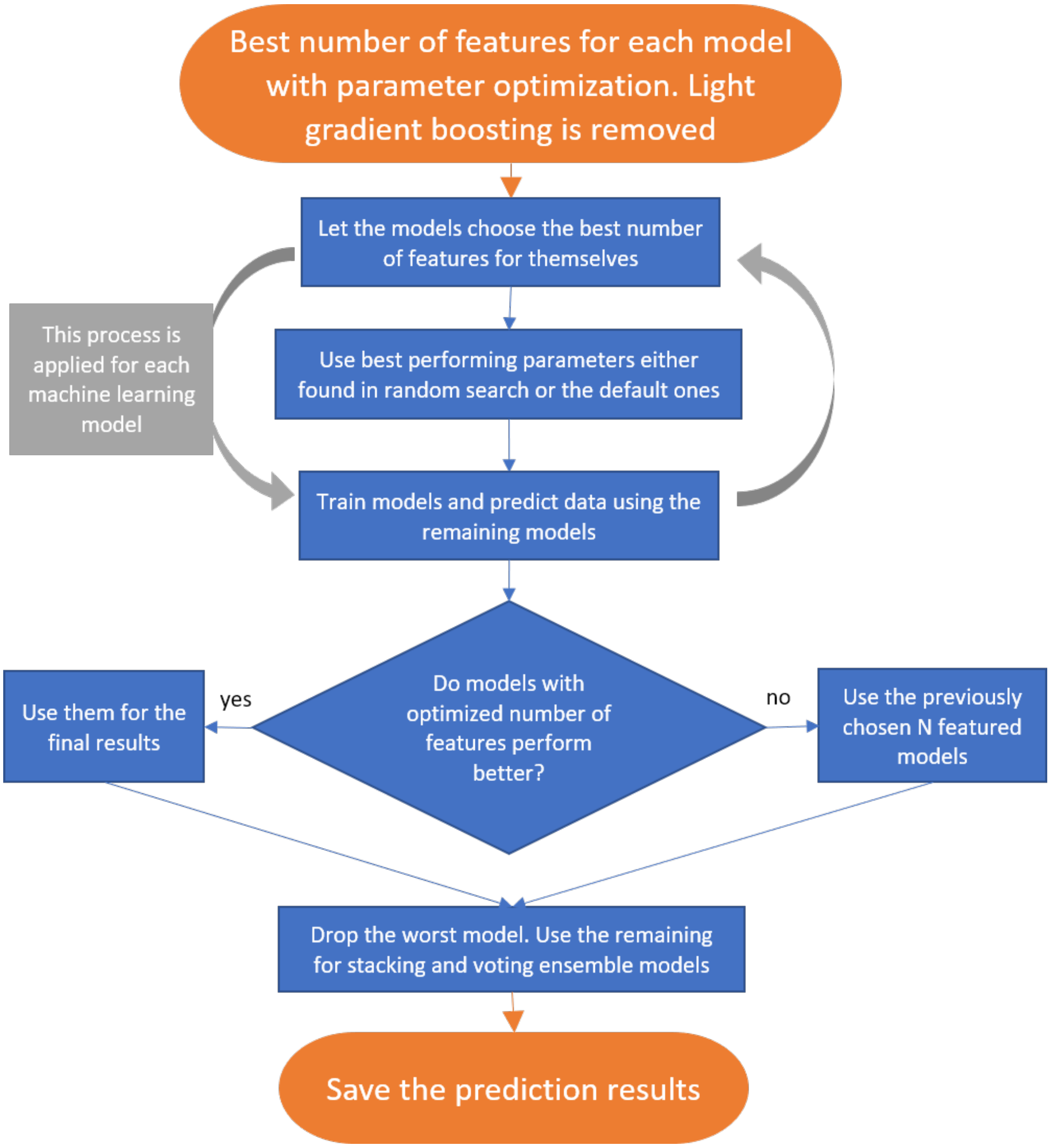
Table 6.12 Modeling phase 2 – b2. TEST Data Performance - MAE

	linear	lasso	elastic	et	gb	lgbm	catb	naive	stack	voting
nl_f_01	0.537	0.555	0.537	0.561	0.52	0.51	0.51	0.583	0.516	0.522
nl_f_02	0.623	0.628	0.617	0.632	0.585	0.583	0.567	0.699	0.567	0.589
nl_f_03	0.672	0.68	0.669	0.681	0.622	0.633	0.587	0.772	0.596	0.636
nl_f_04	0.707	0.714	0.707	0.712	0.656	0.649	0.633	0.849	0.639	0.674
nl_f_05	0.729	0.736	0.73	0.737	0.671	0.667	0.643	0.902	0.65	0.693
nl_f_06	0.758	0.752	0.748	0.749	0.62	0.657	0.557	0.934	0.567	0.691
nl_f_07	0.773	0.764	0.762	0.761	0.618	0.659	0.557	0.978	0.566	0.702
nl_f_08	0.771	0.772	0.772	0.765	0.62	0.661	0.549	0.998	0.555	0.705
nl_f_09	0.775	0.781	0.78	0.777	0.636	0.658	0.586	1.027	0.594	0.72
nl_f_10	0.782	0.786	0.785	0.781	0.628	0.666	0.573	1.028	0.582	0.72
nl_f_11	0.793	0.792	0.79	0.783	0.606	0.658	0.547	1.06	0.555	0.72
nl_f_12	0.788	0.793	0.79	0.784	0.594	0.66	0.541	1.062	0.547	0.716
nl_f_13	0.793	0.793	0.789	0.785	0.604	0.665	0.548	1.07	0.555	0.718
nl_f_14	0.792	0.793	0.788	0.784	0.603	0.668	0.548	1.071	0.555	0.718
nl_f_15	0.791	0.793	0.79	0.788	0.605	0.653	0.547	1.079	0.555	0.718
nl_f_16	0.79	0.794	0.79	0.788	0.603	0.663	0.544	1.083	0.553	0.718
nl_f_17	0.786	0.791	0.785	0.788	0.617	0.659	0.569	1.07	0.581	0.723
nl_f_18	0.788	0.794	0.789	0.791	0.594	0.665	0.554	1.081	0.564	0.719
nl_f_19	0.784	0.791	0.785	0.787	0.605	0.659	0.551	1.086	0.561	0.717
nl_f_20	0.783	0.792	0.783	0.788	0.599	0.653	0.553	1.065	0.564	0.718
nl_f_21	0.787	0.795	0.788	0.785	0.598	0.655	0.547	1.066	0.559	0.72
nl_f_22	0.787	0.795	0.787	0.785	0.594	0.653	0.553	1.036	0.565	0.718
nl_f_23	0.788	0.797	0.79	0.787	0.599	0.668	0.553	1.015	0.563	0.719
nl_f_24	0.797	0.803	0.797	0.792	0.63	0.666	0.578	1.047	0.591	0.733
nl_f_25	0.797	0.805	0.799	0.794	0.614	0.641	0.557	1.07	0.568	0.725
nl_f_26	0.801	0.806	0.802	0.794	0.606	0.655	0.557	1.1	0.567	0.727
nl_f_27	0.81	0.811	0.807	0.795	0.612	0.649	0.557	1.139	0.566	0.729
nl_f_28	0.812	0.813	0.813	0.803	0.591	0.654	0.54	1.161	0.551	0.73
nl_f_29	0.813	0.818	0.813	0.793	0.583	0.654	0.544	1.176	0.554	0.73
nl_f_30	0.813	0.817	0.815	0.793	0.603	0.662	0.552	1.189	0.56	0.73
nl_f_31	0.813	0.816	0.814	0.8	0.603	0.664	0.553	1.189	0.56	0.73
nl_f_32	0.811	0.815	0.814	0.792	0.601	0.649	0.55	1.189	0.559	0.732

Table 6.13 Modeling phase 2 – b3. TEST Data Performance - MAPE

	linear	lasso	elastic	et	gb	lgbm	catb	naive	stack	voting
nl_f_01	164	158	162	127	159	152	160	189	152	149
nl_f_02	183	183	184	142	180	178	176	227	161	162
nl_f_03	190	189	186	157	194	178	182	243	171	173
nl_f_04	214	214	212	168	204	191	207	289	193	193
nl_f_05	226	227	225	174	218	190	211	348	199	205
nl_f_06	225	227	227	169	204	201	178	343	172	198
nl_f_07	233	224	229	170	194	200	183	367	177	200
nl_f_08	227	223	225	169	201	208	188	357	181	194
nl_f_09	224	224	226	173	206	210	195	347	188	202
nl_f_10	234	226	227	172	202	200	199	363	192	201
nl_f_11	230	226	227	175	200	200	189	336	182	194
nl_f_12	224	222	224	175	193	200	190	335	183	191
nl_f_13	225	217	220	173	188	205	191	367	185	192
nl_f_14	225	215	219	172	184	201	184	352	177	190
nl_f_15	221	212	217	169	195	196	193	324	185	189
nl_f_16	217	216	216	167	181	197	166	319	162	183
nl_f_17	216	213	215	172	183	192	181	312	176	187
nl_f_18	216	212	216	169	176	203	180	342	173	185
nl_f_19	215	211	214	168	183	196	180	340	174	185
nl_f_20	212	213	213	167	177	193	180	332	174	186
nl_f_21	216	211	217	160	173	196	171	373	166	184
nl_f_22	214	212	213	159	188	202	175	360	169	184
nl_f_23	221	214	220	168	181	198	178	359	172	185
nl_f_24	224	219	224	159	199	195	190	378	183	194
nl_f_25	218	218	216	160	190	181	181	376	174	186
nl_f_26	218	217	217	159	187	193	188	362	179	188
nl_f_27	221	219	218	159	187	198	179	365	170	186
nl_f_28	226	219	219	167	190	196	185	339	178	189
nl_f_29	223	219	221	160	190	192	194	363	186	191
nl_f_30	222	220	220	158	193	193	190	358	181	188
nl_f_31	222	220	221	168	192	193	189	371	183	189
nl_f_32	222	222	221	160	196	195	191	381	183	191

Figure 6.10 Modeling Stage Phase 3; Optimized Number of Features and Optimized Model Parameters with Random Search



6.5.5 Step 5: Feature selection phase for modeling phase 3; feature selection results in variable number of features

As mentioned in step 2, in this step algorithms are free to choose the best number of variables. This way, the models can fit to the number of features they consider best for performance. Note that feature selection number is restarted for it to be coherent with the names of the modeling phases

6.5.6 Step 6: Modeling phase 3; with optimized number of features and optimized model parameters with random search

- Features selected at step 5 are used.
- Among the remaining 6 models, 1 is linear regression which does not need any parameter optimization. That is why parameter optimization with random search is performed for the models with the function sklearn.model_selection.RandomizedSearchCV().
- iii. Among those 5 models, catboost is dropped by hoping to get a better voting ensemble result. Even though the performance of catboost for modeling stage 3 is below the naïve model, its previous performances were above the naïve model. We could just keep it. Model based feature selection is performed only for individual models, however separate a variable number of features can not be used for every model in the voting regression model. That is why feature selection operations are done specifically for every single model not for the two ensemble models stacking and voting and then the models are trained. It is seen that the individual models do not beat the previous champion, phase 2 voting. That is why the worst-performing algorithm is dropped for the ensemble models with the hope of beating the phase 2 champion model. Gradient boosting is used for model selection for the ensemble models as is seen in Figure 6.11. The final champion is the phase 3 voting ensemble according to the back-test performances.
- iv. R-square, MAE and MAPE performance results are shown in tables from 6.14 up to 6.19.

Figure 6.11 Models used in Voting Ensemble Regressor in Modeling Stage Phase 3

```
    regressor_fsel  = GradientBoostingRegressor()

    estimators = [
     ('linear'     , LinearRegression()          )
    ,('lasso'      , l_e['lasso'][i]              )
    ,('elasticnet', ElasticNet(alpha=0.1)        )
    ,('et'         , l_e['et'    ][i]             )
    ,('gb'         , GradientBoostingRegressor() )
    ]
    regressor_model = VotingRegressor(
         estimators      = estimators
        )
```

Table 6.14 Modeling phase 3 – a1. BACK-TEST Data Performance - R-square

	linear	lasso	elastic	et	gb	catb	naive	stack	voting
nl_f_01	0.578	0.582	0.594	0.55	0.569	0.506	0.53	0.602	0.612
nl_f_02	0.425	0.461	0.457	0.414	0.429	0.285	0.318	0.464	0.473
nl_f_03	0.309	0.344	0.348	0.311	0.292	0.169	0.124	0.348	0.365
nl_f_04	0.231	0.257	0.271	0.24	0.254	0.066	-0.04	0.287	0.297
nl_f_05	0.175	0.206	0.221	0.193	0.177	-0.09	-0.18	0.192	0.243
nl_f_06	0.144	0.175	0.189	0.16	0.123	-0.09	-0.3	0.161	0.203
nl_f_07	0.109	0.145	0.169	0.151	0.067	-0.08	-0.38	0.121	0.182
nl_f_08	0.078	0.131	0.153	0.142	0.024	-0.11	-0.44	0.099	0.171
nl_f_09	0.068	0.107	0.139	0.129	0.032	-0.15	-0.49	0.086	0.154
nl_f_10	0.037	0.098	0.128	0.123	0.057	-0.22	-0.54	0.075	0.146
nl_f_11	0.029	0.088	0.122	0.125	0.05	-0.19	-0.58	0.073	0.135
nl_f_12	0.015	0.092	0.12	0.117	0.059	-0.17	-0.61	0.075	0.141
nl_f_13	0.049	0.079	0.119	0.114	0.037	-0.22	-0.61	0.081	0.131
nl_f_14	0.027	0.102	0.119	0.114	0.079	-0.14	-0.63	0.095	0.143
nl_f_15	0.017	0.075	0.114	0.11	0.037	-0.18	-0.65	0.057	0.13
nl_f_16	0.029	0.071	0.112	0.106	0.02	-0.1	-0.64	0.033	0.131
nl_f_17	0.03	0.107	0.121	0.103	0.029	-0.16	-0.62	0.049	0.141
nl_f_18	0.017	0.064	0.119	0.103	0.035	-0.17	-0.58	0.069	0.134
nl_f_19	0.018	0.064	0.122	0.105	0.045	-0.21	-0.53	0.056	0.139
nl_f_20	0.018	0.101	0.123	0.108	0.024	-0.22	-0.48	0.03	0.13
nl_f_21	0.012	0.106	0.125	0.094	-0.02	-0.2	-0.45	0.011	0.131
nl_f_22	0.012	0.105	0.123	0.086	-0.03	-0.25	-0.43	0.005	0.128
nl_f_23	0.007	0.107	0.124	0.087	-0.03	-0.17	-0.41	-0	0.134
nl_f_24	-0.02	0.072	0.105	0.071	0	-0.35	-0.51	-0.01	0.128
nl_f_25	-0.03	0.049	0.09	0.078	0.04	-0.25	-0.58	0.036	0.119
nl_f_26	-0.05	0.042	0.084	0.081	0.003	-0.35	-0.65	0.007	0.105
nl_f_27	-0.05	0.037	0.08	0.076	0.029	-0.17	-0.72	0.005	0.105
nl_f_28	-0.04	-0	0.078	0.079	0.025	-0.19	-0.77	0.028	0.109
nl_f_29	-0.04	0.027	0.075	0.086	-0.07	-0.19	-0.83	0.046	0.119
nl_f_30	-0.04	0.025	0.073	0.096	0.052	-0.3	-0.88	0.071	0.121
nl_f_31	-0.05	0.025	0.073	0.092	0.036	-0.14	-0.9	0.06	0.11
nl_f_32	-0.05	0.026	0.074	0.097	0.021	-0.13	-0.92	0.034	0.109

Table 6.15 Modeling phase 3 – a2. BACK-TEST Data Performance - MAE

	linear	lasso	elastic	et	gb	catb	naive	stack	voting
nl_f_01	0.492	0.493	0.482	0.504	0.498	0.534	0.509	0.477	0.47
nl_f_02	0.579	0.556	0.559	0.579	0.581	0.664	0.62	0.56	0.55
nl_f_03	0.635	0.616	0.613	0.628	0.646	0.717	0.715	0.616	0.605
nl_f_04	0.676	0.662	0.65	0.66	0.666	0.764	0.784	0.648	0.64
nl_f_05	0.699	0.683	0.672	0.681	0.694	0.834	0.837	0.684	0.663
nl_f_06	0.715	0.697	0.684	0.695	0.715	0.818	0.878	0.698	0.679
nl_f_07	0.733	0.711	0.693	0.699	0.75	0.824	0.908	0.723	0.691
nl_f_08	0.748	0.716	0.699	0.704	0.775	0.836	0.928	0.741	0.698
nl_f_09	0.751	0.724	0.704	0.71	0.764	0.857	0.943	0.739	0.703
nl_f_10	0.759	0.726	0.707	0.713	0.757	0.879	0.955	0.747	0.705
nl_f_11	0.764	0.73	0.709	0.712	0.758	0.863	0.971	0.746	0.707
nl_f_12	0.769	0.727	0.71	0.715	0.759	0.864	0.983	0.75	0.706
nl_f_13	0.752	0.734	0.71	0.716	0.767	0.877	0.987	0.749	0.708
nl_f_14	0.763	0.722	0.711	0.717	0.753	0.847	0.994	0.743	0.705
nl_f_15	0.768	0.736	0.714	0.721	0.761	0.86	1.003	0.752	0.71
nl_f_16	0.764	0.743	0.715	0.723	0.771	0.831	1.002	0.765	0.709
nl_f_17	0.757	0.72	0.709	0.725	0.763	0.846	0.994	0.752	0.704
nl_f_18	0.767	0.74	0.71	0.725	0.766	0.854	0.976	0.753	0.707
nl_f_19	0.764	0.738	0.709	0.725	0.762	0.864	0.955	0.757	0.705
nl_f_20	0.764	0.722	0.708	0.726	0.766	0.872	0.935	0.763	0.708
nl_f_21	0.765	0.72	0.707	0.729	0.792	0.863	0.921	0.775	0.708
nl_f_22	0.764	0.721	0.708	0.734	0.792	0.882	0.913	0.777	0.709
nl_f_23	0.764	0.72	0.707	0.733	0.796	0.863	0.906	0.782	0.707
nl_f_24	0.779	0.74	0.717	0.739	0.79	0.93	0.951	0.791	0.712
nl_f_25	0.787	0.749	0.724	0.735	0.773	0.896	0.984	0.773	0.714
nl_f_26	0.792	0.753	0.727	0.735	0.781	0.939	1.01	0.779	0.722
nl_f_27	0.792	0.755	0.728	0.734	0.78	0.872	1.036	0.784	0.723
nl_f_28	0.792	0.774	0.729	0.736	0.782	0.876	1.054	0.779	0.721
nl_f_29	0.793	0.758	0.73	0.734	0.809	0.866	1.069	0.775	0.719
nl_f_30	0.794	0.759	0.73	0.73	0.779	0.913	1.08	0.767	0.719
nl_f_31	0.799	0.759	0.73	0.732	0.777	0.855	1.079	0.767	0.722
nl_f_32	0.795	0.759	0.73	0.731	0.782	0.845	1.081	0.776	0.722

Table 6.16 Modeling phase 3 – a3. BACK-TEST Data Performance - MAPE

	linear	lasso	elastic	et	gb	catb	naive	stack	voting
nl_f_01	305	270	249	248	295	338	368	260	257
nl_f_02	349	304	279	277	344	452	454	308	293
nl_f_03	376	323	288	270	347	494	523	316	304
nl_f_04	409	385	322	282	396	536	674	355	339
nl_f_05	402	372	310	273	379	551	722	346	321
nl_f_06	386	346	279	266	385	523	728	343	303
nl_f_07	385	326	249	245	403	661	706	356	285
nl_f_08	395	316	244	229	415	623	717	359	284
nl_f_09	387	328	239	220	373	565	711	334	276
nl_f_10	374	305	253	221	380	650	686	373	283
nl_f_11	389	309	258	216	372	606	678	350	280
nl_f_12	393	315	263	227	481	697	617	447	301
nl_f_13	397	362	274	235	494	672	628	418	312
nl_f_14	416	329	280	238	553	601	579	501	335
nl_f_15	409	337	277	240	401	617	594	378	303
nl_f_16	404	344	278	232	395	483	604	377	295
nl_f_17	379	314	268	240	445	676	622	382	306
nl_f_18	390	351	270	246	475	728	562	431	300
nl_f_19	401	355	272	247	501	705	564	469	310
nl_f_20	398	313	266	252	446	685	572	446	302
nl_f_21	422	323	274	254	517	673	561	473	319
nl_f_22	417	327	276	266	532	783	576	487	322
nl_f_23	407	313	264	257	575	688	609	490	322
nl_f_24	417	339	274	254	536	811	613	504	316
nl_f_25	417	341	275	256	469	682	630	453	305
nl_f_26	421	344	273	247	476	795	614	455	304
nl_f_27	413	344	269	236	464	672	720	436	304
nl_f_28	406	378	271	249	500	702	797	476	309
nl_f_29	402	353	268	247	544	612	822	466	308
nl_f_30	407	345	269	248	513	711	857	452	306
nl_f_31	406	346	270	248	481	552	840	452	306
nl_f_32	406	347	271	250	448	555	830	430	297

Table 6.17 Modeling phase 3 – b1. TEST Data Performance - R-square

	linear	lasso	elastic	et	gb	catb	naive	stack	voting
nl_f_01	0.618	0.579	0.587	0.553	0.631	0.679	0.496	0.624	0.617
nl_f_02	0.499	0.468	0.473	0.438	0.53	0.658	0.3	0.518	0.505
nl_f_03	0.422	0.391	0.394	0.366	0.467	0.654	0.159	0.453	0.432
nl_f_04	0.368	0.341	0.34	0.318	0.434	0.645	0.021	0.414	0.384
nl_f_05	0.33	0.309	0.304	0.287	0.404	0.662	-0.08	0.382	0.349
nl_f_06	0.308	0.284	0.276	0.261	0.388	0.654	-0.14	0.368	0.326
nl_f_07	0.277	0.264	0.251	0.247	0.376	0.666	-0.24	0.354	0.306
nl_f_08	0.261	0.245	0.238	0.244	0.377	0.667	-0.3	0.356	0.3
nl_f_09	0.255	0.236	0.225	0.233	0.366	0.673	-0.36	0.341	0.285
nl_f_10	0.246	0.225	0.217	0.229	0.364	0.67	-0.37	0.343	0.279
nl_f_11	0.242	0.219	0.211	0.22	0.345	0.666	-0.44	0.326	0.266
nl_f_12	0.235	0.217	0.211	0.217	0.349	0.668	-0.44	0.33	0.268
nl_f_13	0.236	0.235	0.211	0.222	0.342	0.679	-0.46	0.323	0.264
nl_f_14	0.238	0.219	0.211	0.219	0.338	0.671	-0.45	0.318	0.265
nl_f_15	0.239	0.218	0.207	0.223	0.337	0.682	-0.47	0.318	0.264
nl_f_16	0.245	0.22	0.206	0.22	0.354	0.676	-0.47	0.335	0.27
nl_f_17	0.249	0.222	0.214	0.217	0.332	0.667	-0.44	0.325	0.268
nl_f_18	0.246	0.244	0.21	0.22	0.348	0.669	-0.46	0.327	0.266
nl_f_19	0.253	0.246	0.213	0.219	0.353	0.681	-0.49	0.334	0.274
nl_f_20	0.253	0.222	0.215	0.217	0.34	0.667	-0.46	0.32	0.269
nl_f_21	0.25	0.218	0.213	0.217	0.351	0.67	-0.44	0.328	0.272
nl_f_22	0.248	0.219	0.213	0.216	0.348	0.665	-0.4	0.327	0.27
nl_f_23	0.248	0.216	0.209	0.216	0.347	0.671	-0.36	0.32	0.269
nl_f_24	0.232	0.201	0.193	0.204	0.354	0.662	-0.43	0.331	0.258
nl_f_25	0.225	0.195	0.183	0.204	0.337	0.675	-0.49	0.319	0.252
nl_f_26	0.214	0.191	0.176	0.204	0.346	0.649	-0.55	0.326	0.245
nl_f_27	0.202	0.184	0.168	0.202	0.347	0.679	-0.66	0.321	0.239
nl_f_28	0.195	0.192	0.166	0.203	0.336	0.662	-0.71	0.316	0.235
nl_f_29	0.191	0.188	0.163	0.207	0.34	0.687	-0.7	0.322	0.236
nl_f_30	0.191	0.178	0.163	0.208	0.342	0.695	-0.74	0.333	0.236
nl_f_31	0.196	0.179	0.165	0.211	0.332	0.692	-0.74	0.318	0.239
nl_f_32	0.198	0.18	0.166	0.215	0.341	0.689	-0.77	0.326	0.239

Table 6.18 Modeling phase 3 – b2. TEST Data Performance - MAE

	linear	lasso	elastic	et	gb	catb	naive	stack	voting
nl_f_01	0.534	0.554	0.548	0.559	0.518	0.49	0.583	0.527	0.527
nl_f_02	0.615	0.628	0.626	0.634	0.593	0.502	0.699	0.6	0.603
nl_f_03	0.668	0.68	0.68	0.682	0.638	0.505	0.772	0.647	0.655
nl_f_04	0.707	0.714	0.715	0.713	0.665	0.512	0.849	0.677	0.69
nl_f_05	0.732	0.736	0.738	0.734	0.688	0.498	0.902	0.701	0.713
nl_f_06	0.747	0.752	0.757	0.751	0.7	0.505	0.934	0.711	0.73
nl_f_07	0.764	0.764	0.77	0.761	0.704	0.494	0.978	0.718	0.742
nl_f_08	0.772	0.772	0.777	0.763	0.706	0.49	0.998	0.719	0.747
nl_f_09	0.774	0.779	0.785	0.771	0.713	0.488	1.027	0.727	0.755
nl_f_10	0.778	0.786	0.791	0.775	0.718	0.494	1.028	0.728	0.76
nl_f_11	0.783	0.79	0.795	0.781	0.727	0.496	1.06	0.738	0.768
nl_f_12	0.787	0.793	0.796	0.784	0.726	0.491	1.062	0.736	0.768
nl_f_13	0.792	0.79	0.797	0.781	0.727	0.485	1.07	0.738	0.771
nl_f_14	0.788	0.792	0.797	0.782	0.729	0.486	1.071	0.74	0.77
nl_f_15	0.789	0.793	0.799	0.781	0.734	0.478	1.079	0.744	0.771
nl_f_16	0.786	0.792	0.8	0.783	0.722	0.486	1.083	0.733	0.768
nl_f_17	0.785	0.792	0.797	0.783	0.735	0.493	1.07	0.741	0.77
nl_f_18	0.787	0.786	0.8	0.783	0.724	0.485	1.081	0.736	0.77
nl_f_19	0.782	0.784	0.798	0.783	0.722	0.477	1.086	0.732	0.767
nl_f_20	0.781	0.792	0.797	0.784	0.729	0.485	1.065	0.738	0.768
nl_f_21	0.785	0.795	0.799	0.783	0.72	0.483	1.066	0.734	0.768
nl_f_22	0.786	0.795	0.799	0.782	0.724	0.481	1.036	0.736	0.769
nl_f_23	0.786	0.797	0.801	0.782	0.723	0.481	1.015	0.74	0.769
nl_f_24	0.794	0.803	0.808	0.789	0.72	0.493	1.047	0.731	0.774
nl_f_25	0.796	0.805	0.812	0.79	0.728	0.488	1.07	0.737	0.777
nl_f_26	0.8	0.807	0.815	0.791	0.726	0.507	1.1	0.737	0.78
nl_f_27	0.806	0.811	0.82	0.792	0.727	0.477	1.139	0.741	0.783
nl_f_28	0.81	0.81	0.822	0.793	0.733	0.497	1.161	0.743	0.786
nl_f_29	0.811	0.812	0.823	0.792	0.734	0.481	1.176	0.742	0.787
nl_f_30	0.812	0.817	0.823	0.792	0.729	0.479	1.189	0.737	0.786
nl_f_31	0.811	0.816	0.823	0.791	0.735	0.473	1.189	0.744	0.785
nl_f_32	0.811	0.816	0.822	0.79	0.726	0.479	1.189	0.736	0.785

Table 6.19 Modeling phase 3 – b3. TEST Data Performance - MAPE

	linear	lasso	elastic	et	gb	catb	naive	stack	voting
nl_f_01	162	156	146	127	154	146	189	151	146
nl_f_02	180	180	169	140	170	159	227	163	161
nl_f_03	193	189	180	154	194	167	243	182	175
nl_f_04	217	214	201	165	199	156	289	196	194
nl_f_05	231	227	213	168	220	171	348	212	207
nl_f_06	236	227	213	165	213	168	343	210	208
nl_f_07	242	229	215	165	207	165	367	206	208
nl_f_08	234	223	211	163	211	153	357	208	205
nl_f_09	232	225	214	165	209	160	347	205	206
nl_f_10	230	226	215	169	214	166	363	212	208
nl_f_11	226	223	215	169	218	151	336	211	206
nl_f_12	227	222	213	168	209	153	335	205	204
nl_f_13	232	223	212	166	212	165	367	203	205
nl_f_14	229	215	211	164	211	155	352	207	204
nl_f_15	225	212	208	162	215	169	324	210	202
nl_f_16	220	218	207	161	202	160	319	198	197
nl_f_17	223	213	208	163	207	165	312	203	197
nl_f_18	226	217	209	161	204	147	342	196	197
nl_f_19	223	214	207	160	202	156	340	197	197
nl_f_20	219	212	206	161	200	156	332	195	195
nl_f_21	220	211	205	157	197	148	373	193	193
nl_f_22	221	212	204	159	198	149	360	195	195
nl_f_23	223	214	207	160	200	152	359	199	195
nl_f_24	229	219	210	160	198	157	378	196	200
nl_f_25	219	218	207	157	194	142	376	190	195
nl_f_26	222	218	208	155	199	165	362	194	197
nl_f_27	223	219	210	156	202	146	365	194	197
nl_f_28	222	219	212	157	201	161	339	197	199
nl_f_29	222	218	211	156	206	152	363	194	197
nl_f_30	220	219	211	156	194	158	358	190	196
nl_f_31	222	220	212	156	203	157	371	199	198
nl_f_32	220	220	212	157	200	156	381	197	198

6.6 Validation Stage

The validation and visualization stages are the last stages before the deployment of an industrial project, so the final controls are done at these stages. The validation stage is generally performed after finishing the work and scoring incoming new data as time passes. Data scientists and business owners examine the newly scored data to check if the results make sense in terms of business. It is the real indicator of the success of the model; however, we do not have new data consistently we acquire. That is why this stage is simulated by the using year 2020 data for test purposes (back-test) as it is time independent of the modeling data. We accept this back-test data as the base for the success of the machine learning pipeline we created. The validation step in the CRISP-DM framework is measuring the significance of the resultant machine learning framework with back-test data. This has already been performed during the modeling stage. The results of the modeling stage have shown that all the models provide a level of explanation compared to the random guess and the good ones beat the naive model. That is why currently the study can be accepted as.

6.7 Visualization Stage

Figure 6.12 shows the (T+1) up to (T+3) R-square results of bad performing models vs. naïve model and the champion, voting ensemble. Figure 6.13 shows the improvement in back-test R-square results of the champion model through the modeling phases. It is seen that the best results are obtained in modeling phase 3. Figure 6.14 shows the comparison of back-test and test results of the voting ensemble. It shows the importance of using timewise separated back-test data since randomly separated regular test data results seem a lot different. We can say regular test data causes overfit since the test data is from the same pool with modeling data in terms of time. Figure 6.15 shows the comparison of the overall champion model and naïve model in modeling phase 3 by R-square values. Naïve model cannot give any insight after 3 hours and voting ensemble model provide significant improvement for (T+1) up to (T+32).

Figure 6.12 Modeling Phase-1 Back-test Bad Performing Models vs. Naive Model Voting Ensemble Results

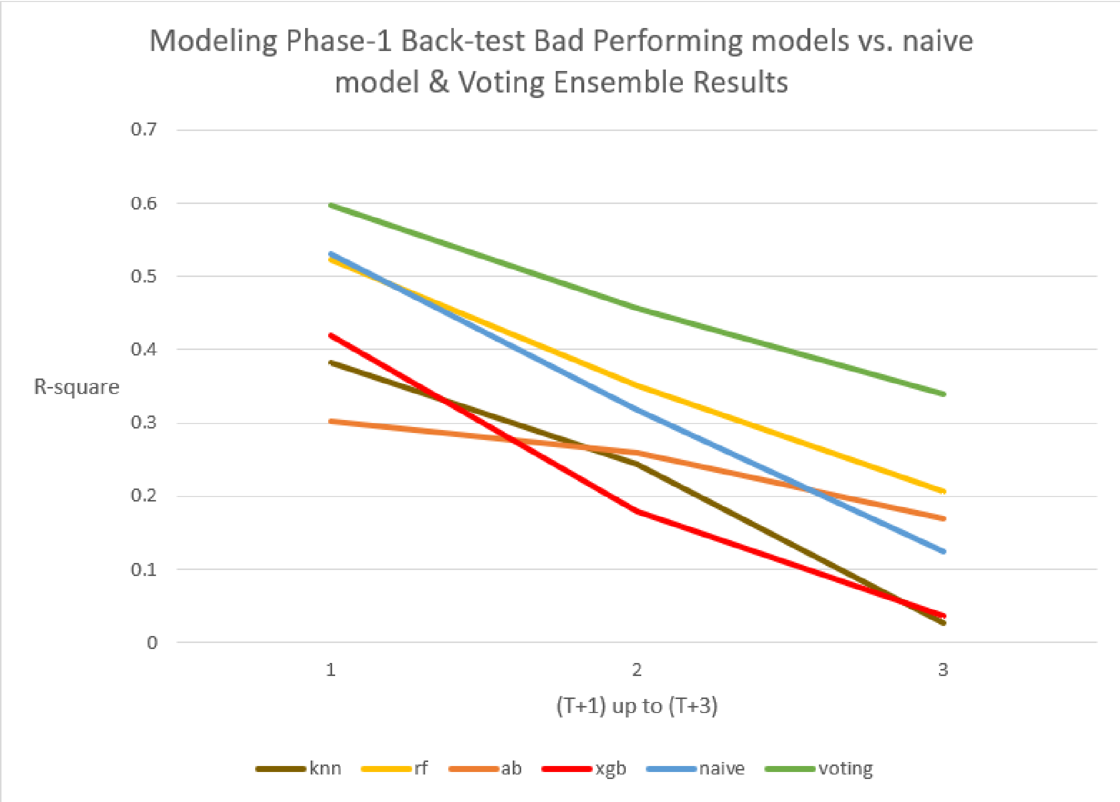


Figure 6.13 Improvements in Voting Ensembles' Back-test R-Square Results

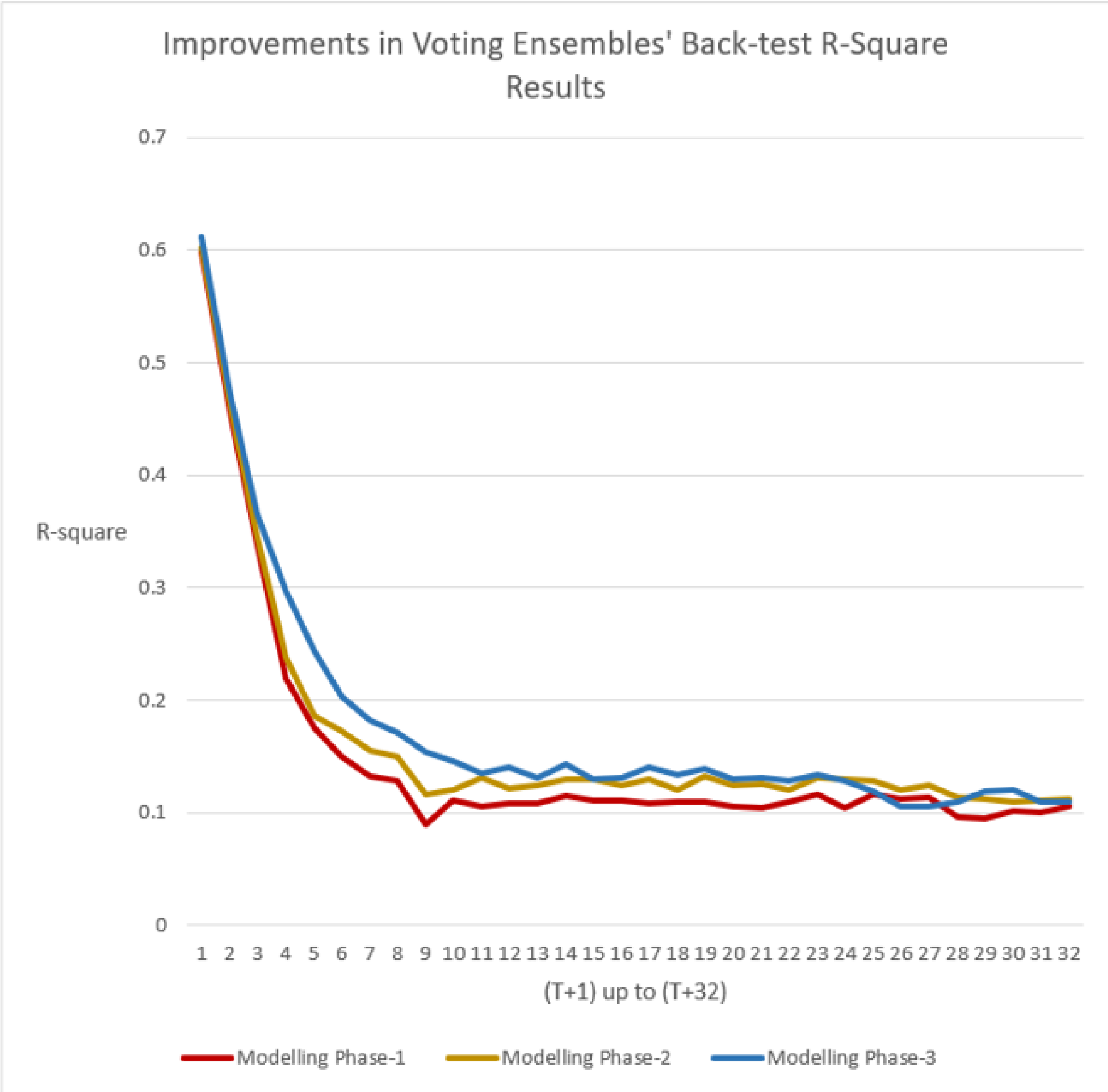


Figure 6.14 Modeling Phase-3 Voting Ensemble Back-test vs. Test Results

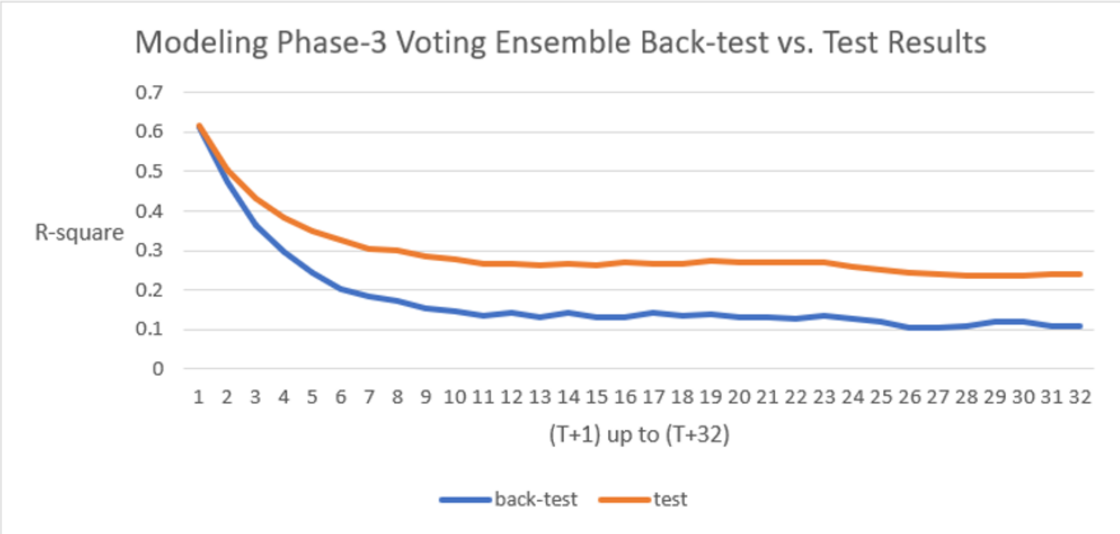
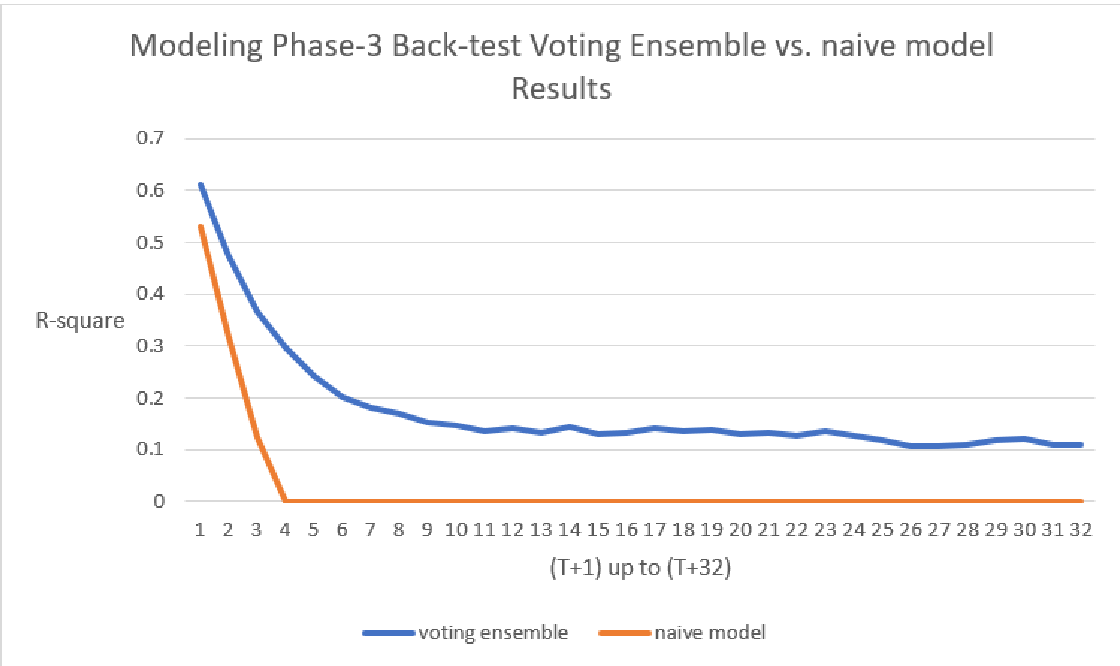


Figure 6.15 Modeling Phase-3 Back-test Voting Ensemble vs. Naive Model Results



7. CONCLUSION

In this study, energy imbalances at the delivery time in the Turkish Energy Market for the future 32 hours are tried to be predicted for the intraday market and balancing power market. The nature of the IM requires net imbalance predictions for (T+1) up to (T+24). Regarding the ability to forecast the energy imbalance which we call net imbalance the TradeCo that is partnered with can get commercial advantages as net imbalance is effective on the IM prices. Since the future hours that will be predicted are not far and the study is made with a TradeCo, obtaining successful results for the close hours like (T+1) up to (T+6) for IM has been our primary objective. The results are promising for this scope since the net imbalance predicted beats the naive model which is accepting the closest available net imbalance as the prediction with a single variable for all the future 32 hours. Calling a single variable model as a model may seem like a bold approach however it creates a good baseline for comparing other models as that single variable is by far the most explanatory variable for (T+1) up to (T+3). Besides, it makes sense in terms of business as currently the traders check the closest available past net imbalance values. The champion is an ensemble model voting regressor which consists of linear regression, lasso, elastic net, extra trees, gradient boosting models. Another ensemble model stacking and elastic net are the rivals for the champion model. The closest hour which is the next hour can be predicted with a 0.612 R-square performance. Even the 6-hours future net imbalance can be predicted with 0.203 R-square success. Beyond (T+6), R-square of predicting the future net imbalances goes below 0.2 value. From 6 hours to 32 hours, success gradually decreases to 0.1. The nature of the BPM requires net imbalance predictions from the 9-hour future to the 32-hour future. The 9-hour gap between the target hour and the hour that the prediction is performed is big, so predicting net imbalance values for these hours is harder. For (T+9) net imbalance prediction R-square found by the champion model voting ensemble with optimized parameters is 0.154 which is not as strong as the first 6 hours' predictions. However, without this study, even this level of understanding was not available, so this can be considered a success. These results are all taken by a timely separated test data obtained by keeping all the available the year 2020 data for testing purposes. The next step for this study is to predict the IM price using the results of this model.

BIBLIOGRAPHY

AlFuhaid, A., El-Sayed, M., & Mahmoud, M. (1997). Cascaded artificial neural networks for short-term load forecasting. *IEEE Transactions on Power Systems*, *12*(4), 1524–1529.

AYDEM (2020). Privatization process in electricity market in turkey.

Becirovic, E. & Cosovic, M. (2016). Machine learning techniques for short-term load forecasting. In *2016 4th International Symposium on Environmental Friendly Energies and Applications (EFEA)*. IEEE.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L. (2001). *Machine Learning*, *45*(1), 5–32.

Chen, T. & Guestrin, C. (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Chow, T. & Leung, C. (1996). Neural network based short-term load forecasting using weather compensation. *IEEE Transactions on Power Systems*, *11*(4), 1736–1742.

Dundar, S. & Utas, A. (2020). Energy sector unbundling in turkey.

ELDER (2020). Electricity distribution system operators of turkey.

EMO (2012). Electricity sector privatizations report.

EMRA (2012). Karar no: 4019 dağitim ve perakende satiş faaliyetlerinin hukuki ayriştirilmasina İlişkin usul ve esaslar.

EMRA (2019). Electricity market development repo.

EMRA (2020a). Becoming a free consumer as a residential subscriber.

EMRA (2020b). Electricity market development report.

Essl, A., Ortner, A., Haas, R., & Hettegger, P. (2017). Machine learning analysis for a flexibility energy approach towards renewable energy integration with dynamic forecasting of electricity balancing power. In *2017 14th International Conference on the European Energy Market (EEM)*. IEEE.

EXIST (2020). Energy exchange istanbul company.

Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Friedman, J. H. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*, 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42.

Gordon, A. D., Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. *Biometrics*, *40*(3), 874.

Gunduz, S., Ugurlu, U., & Öksüz, I. (2020). Transfer learning for electricity price forecasting. *ArXiv*, *abs/2007.03762*.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning*. Springer New York.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.

(2017). Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*.

Kearns, M. (1988). Thoughts on hypothesis boosting.

Kiartzis, S., Zoumas, C., Theocharis, J., Bakirtzis, A., & Petridis, V. (1997). Short-term load forecasting in an autonomous power system using artificial neural networks. *IEEE Transactions on Power Systems*, *12*(4), 1591–1596.

Klæboe, G., Eriksrud, A. L., & Fleten, S.-E. (2013). Benchmarking time series based forecasting models for electricity balancing market prices. *Energy Systems*, *6*(1), 43–61.

LegalGazette (2013). Electricity market law.

LegalGazette (2018). Khk-703.

Lindsay, R. & Stigler, G. J. (1969). The organization of industry. *The Journal of Finance*, *24*(5), 1013.

Madnani, N. (2020). Factoranalyzer.

Medium.com (2020). Using crisp-dm to predict car prices.

Möller, C., Rachev, S. T., & Fabozzi, F. J. (2011). Balancing energy strategies in electricity portfolio management. *Energy Economics*, *33*(1), 2–11.

Nickolas, S. (2019). How do economies of scope and economies of scale differ?

Oksuz, I. & Ugurlu, U. (2019). Neural network based model comparison for intraday electricity price forecasting. *Energies*, *12*(23), 4557.

Ostroumova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: unbiased boosting with categorical features. In *NeurIPS*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pollitt, M. G., Davies$*, S., Price*, C. W., Haucap, J., Mulder**, M., Shestalova, V., \& Zwart, G. (20$
$-310.$

Popławski, T., Dudek, G., & Łyp, J. (2015). Forecasting methods for balancing energy market in poland. *International Journal of Electrical Power & Energy Systems*, *65*, 94–101.

Poudineh, R. (2019). Liberalized retail electricity markets. Technical report.

Schapire, R. E. (1990). *Machine Learning*, *5*(2), 197–227.

TEIAS (2019). Energy production and generation reports in 2019.

TEIAS (2020a). Demand forecast report for 10 years.

TEIAS (2020b). Establishment and history of teias.

Uluatam, E. (2011). Privatization process in electricity market in turkey.

Uniejewski, B., Marcjasz, G., & Weron, R. (2019). Understanding intraday electricity markets: Variable selection and very short-term price forecasting using LASSO. *International Journal of Forecasting*, *35*(4), 1533–1547.

WorldBank (2020a). Databank: Turkey.

WorldBank (2020b). Databank: World development indicators; indicator name energy use (kg of oil equivalent per capita).

WorldBank (2020c). Databank: World development indicators; inflation, consumer prices (annual %) - united states, turkey.

WorldBank (2020d). Energy use (kg of oil equivalent per capita).

WorldBank (2020e). Renewable energy consumption (% of total final energy consumption).

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(2), 301–320.