MEF UNIVERSITY

TRACTOR SALES FORECAST USING MACHINE LEARNING

Capstone Project

Yiğitcan Tunay

İSTANBUL, 2018

MEF UNIVERSITY

TRACTOR SALES FORECAST USING MACHINE LEARNING

Capstone Project

Yiğitcan Tunay

Advisor: Prof. Dr. Özgür Özlük

İSTANBUL, 2018

MEF UNIVERSITY

Name of the project: Tractor Sales Forecast Using Machine Learning
Name/Last Name of the Student: Yiğitcan Tunay
Date of Thesis Defense: 07/09/2018

I hereby state that the graduation project prepared by Your Name (Title Format) has been completed under my supervision. I accept this work as a "Graduation Project".

07/09/2018
Prof. Dr. Özgür Özlük

I hereby state that I have examined this graduation project by Your Name (Title Format) which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

07/09/2018
Prof. Dr. Özgür Özlük

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of Yiğitcan Tunay and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Prof. Dr. Özgür Özlük	………………………..

Academic Honesty Pledge

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Yiğitcan Tunay 07.09.2018 Signature

EXECUTIVE SUMMARY

TRACTOR SALES FORECAST USING MACHINE LEARNING

Yiğitcan Tunay

Advisor: Prof. Dr. Özgür Özlük

SEPTEMBER, 2018, 35 pages

This study presents a machine learning model to forecast tractor sales using four years of number of tractor sales based on year, month, city, town, brand and model provided by Turkey Statistical Institute. Tractor sales can vary depending on many different factors. Therefore, it is a challenging task for any company to estimate number of tractor sales that will be sold next year. Having the ability to predict that accurately will contribute companies in many distinct ways. Foreseeing market trends, keeping pace with the competition, delivering the right product to the right customer at the right time, reducing inventory costs, better production planning and cash flow management are major advantages of accurate forecasting. Within the scope of this study, models were developed to predict tractor sales using different statistical and machine learning methods. In further steps of the study, meaningful variables can be added to the dataset in order to reach a better result. Also, market share can be estimated by using different simulation methods which take into consideration those variables.

Key Words: Tractor Sales, Forecasting, Machine Learning.

ÖZET

TRACTOR SALES FORECAST USING MACHINE LEARNING

Yiğitcan Tunay

Tez Danışmanı: Prof. Dr. Özgür Özlük

EYLÜL, 2018, 35 sayfa

Bu çalışma, Türkiye İstatistik Kurumu'ndan sağlanan yıl, ay, şehir, ilçe, marka ve model bazında dört yıllık traktör satışı verilerini kullanarak traktör satışlarını tahminleyebilecek bir makine öğrenmesi modeli geliştirmesini ele almaktadır. Traktör satışları bir çok etkene bağlı olarak değişkenlik gösterebilmektedir. Bu yüzden, her firmanın bir sonraki yıl satacakları traktör adedini tahmin edebilmeleri zorlu bir görev haline gelmektedir. Bir firmanın hangi model traktörden ne kadar satacağını doğru bir şekilde tahminleyebilmesi, firmaya bir çok farklı konuda katkı sağlayacaktır. Pazar trendlerini önceden öngörmek, rekabete ayak uydurmak, doğru zamanda doğru ürünü müşteriye ulaştırabilmek, stok maliyetlerinin azaltılması, daha doğru bir üretim planı ve nakit akışı yönetimi bu avantajların başlıcalarıdır. Çalışma kapsamında, traktör satışlarını farklı istatistiksel ve makine öğrenmesi metodları kullanılarak tahmin edecek modeler geliştirilmiştir. Çalışmanın daha ileri adımlarında, traktör satışlarını etkilemesi muhtemel farklı değişkenlerin de modele eklenmesi ve farklı simülasyon yöntemleri ile bu değişkenlerin alabileceği değerler göz önüne alınarak pazar payının tahminlenmesi sağlanabilir.

Anahtar Kelimeler: Traktör Satışları, Tahminleme, Makine Öğrenmesi.

TABLE OF CONTENTS

1. INTRODUCTION

The agriculture sector has been Turkey's largest employer and the main contributor to the country's GDP. Turkey has a robust agriculture and food industry that employs almost 20 percent of the country’s working population and accounts for 6.1 percent of the country’s GDP in 2016. The sector’s financial contribution to the overall GDP increased 40 percent from 2002 to 2016, reaching USD 52.3 billion in 2016. [1] The strengths of the industry include the size of the market in relation to the country’s young population, a dynamic private sector economy, substantial tourism income and a favorable climate. Turkey is one of the few countries in the world that is self-sufficient in terms of food. The country's fertile soil, access to sufficient water, a suitable climate, and hard-working farmers, all make for a successful agricultural sector. In addition, a broad range of crops can be raised because of the variety of different climates throughout the land. This has allowed Turkey to become the largest producer and exporter of agricultural products in the Near East and North African regions. [2] As a result of this, agricultural mechanization is getting more and more important in terms of technological progress. Tractor sector plays a significant role in this issue. The technological improvement in tools and equipment have increased the quantity and quality of the product. Moreover, it provided less production time.

The competition in the tractor market has been growing day by day. Forecasting sales have been extremely important for companies. Any forecast can be termed as an indicator of what is likely to happen in a specified future time frame in a particular field. Therefore, the sales forecast indicates as to how much of a particular product is likely to be sold in a specified future period in a specified market at a specified price. Accurate sales forecasting is essential for any business to produce the right product, required quantity at the right time. Moreover, it makes the arrangement in advance for raw materials, equipment, and labor etc. Some firms manufacture based on order, but in general, firms produce or order their material in advance to meet the future demand. [4]

There are so many advantages for companies to forecast the future correctly. Your company can manage its inventory, avoiding both overstock and stock-out situations. A stable inventory also means better production plans. Correspondingly, supply chain can be managed more efficiently. It helps you to balance resources and order on time. They all assist in sales planning, demand planning, and financial planning while affecting marketing as well. Companies can prepare some promotions or discount at the right time. [5] Forecasting is not one man’s job. It needs proper co-ordination of all departmental heads in a company.

Thus, by bringing participation of all concerned in the process of forecasting, team spirit and co-ordination is automatically encouraged. Forecasting provides the information which helps the achievement of effective control. The managers become aware of their weaknesses during forecasting and by implementing better effective control they can overcome these weaknesses. [3]

In this paper, it is studied creating a machine learning model for tractor sales forecasting. Tractor sales data in Turkey is used which has been provided by Turkey Statistical Institute thanks to Basak Tractor Company.

1.1. Literature Review

Forecasting any product sales accurately is getting harder due to rapidly changing demands. Each firm tries many different methods to do that. Some of them are traditional methods which face worksheets in excel and some of them are supported by statistics. Machine learning algorithms have been using many different areas in recent years and one of them is sales forecasting.

One of the most commonly used forecasting methods is ARIMA which stands for Autoregressive Integrated Moving Average. ARIMA is a forecasting technique that projects the future values of a series based entirely on its own inertia. Its main application is in the area of short-term forecasting requiring at least 40 historical data points. It works best when your data exhibits a stable or consistent pattern over time with a minimum amount of outliers. ARIMA is usually superior to exponential smoothing techniques when the data is reasonably long and the correlation between past observations is stable. [8] It is specified by three order parameters: *(p,d,q)* An autoregressive component refers to using past values for the series *Y* in the regression equation. Parameter *p* stands for the number of lags used in the model. For instance, ARIMA is showed as $Y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t$ where φ1 and φ2 are parameters for the model. The *d* is represented as the degree of differencing in the integrated component. Differencing a series involves simply subtracting its current and previous values *d* times. A moving average component represents the error of the model as a combination of previous error terms. The order *q* determines the number of terms includes in the model.

$$Y_t = c + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q} + e_t$$

Differencing, autoregressive, and moving average components make up a non-seasonal ARIMA model which can be written as a linear equation:

$$Y_t = c + \phi_1 y_{d\quad t-1} + \phi_p y_{d\quad t-p} + \ldots + \theta_1 e_{t-1} + \theta_q e_{t-q} + e_t$$

where *yd* is *Y* differenced *d* times and *c* is a constant.

ARIMA models can be set up using R which can be found inside of the forecast package. The forecast package allows the user to explicitly specify the order of the model using the arima() function, or automatically generate a set of optimal *(p, d, q)* using auto.arima(). This function searches through combinations of order parameters and picks the set that optimizes model fit criteria. [7]

According to a study in Thailand, they developed hybrid models for forecasting in agricultural production planning. The data was including Thailand's orchid export and Thailand's pork product. Support Vector Machine (SVM) and ARIMA can be represented by both linear and nonlinear values. In this study, many different experiments were performed on the combination of SVM and ARIMA. In the hybrid model, two models were made up of the linear model and nonlinear functions to forecast linear and nonlinear demands. ARIMA was run to predict the linear values of future value. After that, the residuals that were obtained from the ARIMA were entered SVM as the dataset. They trained and predicted the data using that dataset. They used mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) as criteria for the experiment in cross-validation check process. As a result, the most precision model was found as SVM and ARIMA hybrid model by using statistical criteria when you compared to single forecasting models. [9]

In a different study in South Korea, they developed an ARIMA model for the demand forecast for tractor, riding type transplanter and combine harvester. They predicted their demands for three types of machines from 2012 to 2021 in South Korea. They created 6 final models, supply-based ones and stock-based ones for each machine which were generated from 32 tentative models. The stationarities of the series were examined by using ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots. They decided which parameters were suitable for their models observing these plots. They decreased the number of models to 14 after checking their convergence and significance. They determined the final 6 models by comparing AIC (Akaike Information Criterion) and SBC (Schwarz Bayesian Criterion) and the significance of parameter estimation. AIC and SBC are model selection criteria based on residuals. Smaller AIC and SBC statistics generally indicate the better fitting model. As a result, the demand forecast results showed fluctuations with a two-year period or large variation. They explained the reason for that with the policy change of agricultural machinery supply, the presence of outlier, and insufficiency of data. [10]

1.2. About Basak Tractor

Basak Tractor was founded in 1914. In 1962, they started to import tractors from Ford Company. In 1968, they were in cooperation with Ford Company to produce 65% of the tractor parts as local production. In 1976, the company made an agreement with STEYR Company to produce tractors under the name of STEYR brand. They have been producing tractors under the name of Basak since 1996. The company was incorporated by Sanko Group in 2012. They keep producing their tractors 40.000 m² closed area on 275.000 m² area in Sakarya. Their production capacity is 10.000 tractors per year. They have 20 different models which require low maintenance and have low fuel consumption for different segments.

1.3. About the Dataset

The dataset which was provided from Turkey Statistical Institute was shared by Basak Tractor Company. The dataset consists 227122 rows for four years between 2014 and 2017 with the columns of year, month, city code, city, town, brand, model, detailed model, gear type, cabin type, horsepower of the tractor, horsepower segment, tractor segment, model year, origin, region, and count. The explanations can be seen as below.

Year: It indicates in which year a tractor was sold.
Month: It indicates in which month a tractor was sold.
City Code: It indicates in which city a tractor was sold according to Turkish plate code.
City: It indicates in which city a tractor was sold. It includes all the city names in Turkey.
Town: It indicates in which town a tractor was sold. There are 598 distinct town names.
Brand: It indicates the brand of a tractor. There are 27 distinct brands. Brands are shown in Appendix B.
Model: It indicates the model of a tractor. There are 677 distinct models.
Detailed Model: It indicates the model with some details like how many gears that a tractor has.
Gear Type: It indicates the gear type of a tractor. There are two different gear types which are two or four gears.
Cabin Type: It indicates what kind of cabin types a tractor has. There are two different cabin types.
HP (Horsepower) of Tractor: It indicates the horsepower of a tractor. There are 173 distinct horsepower values.

HP Segment: It indicates the horsepower segment of a tractor. There are 12 distinct segments which are grouped by horsepower.
Tractor Segment: It indicates the segment of a tractor. There are three different segments which are the farm, garden, and common service.
Model Year: It indicates the model year of a tractor. There are 6 different model years between 2012 and 2017.
Origin: It indicates the origin of a tractor. It is separated as domestic production and foreign production.
Region: It indicates in which region a tractor was sold. The cities have been divided into regions by Basak Tractor based on their branches.
Count: It indicates the number of tractors sold based on other variables. This variable will be our label while we are creating a model.

The data had more than one value based on year, month, brand, model, city and town. It was obliged to fix that before creating any model so the data was reduced to unique based on year, month, brand, model, city and town by aggregating them.

2. PROJECT DEFINITON

In this section, the objective and scope of the project will be discussed by highlighting the business priorities.

2.1. Project Objectives

Basak Tractor was having difficulties in forecasting their tractor sales. As a result of interviews with the company officials, it has been decided to predict tractor sales. The main objective of the project is to develop a statistical model using machine learning algorithms instead of using traditional methods. After a model is created, the company can use the results for their benefits.

2.2. Project Scope

Analyzing and understanding the given dataset was the first part of the project. Discovering each column is one of the most important parts in machine learning before starting to create any model. For that reason, all columns were examined from a point of business view and technically. Normalizing the data and missing value handling were managed in the exploratory data analysis part. New features were created based on the existing dataset by using principal component analysis. The first step was to try a common forecasting method which was ARIMA. Secondly, we created a basic multiple linear regression model to find out how the dataset changes based on year-month and brand. Finally, three different regression models were created by using Azure. Creating a machine learning model and developing it were challenges during the project.

The economic variables are unstable in Turkey. Especially, exchange rates are highly volatile and inflation has been in double digits since February 2017. Also, disinflation is projected to be slow. The market has been struggling to trust Central Bank of Turkey recently. [11,12] Therefore, while next year's sales were being predicted, it would be a meaningful outlook to create a simulation model by considering economic conditions. But it will not be in this project's content. It can be attempted in the further steps.

3.EXPLORATORY DATA ANALYSIS

Exploratory data analysis is probably the most important part of the project. Understanding and cleaning the data is one of the longest processes in this kind of project. In this part, each column was examined one by one. All the anomalies and wrong values were fixed.

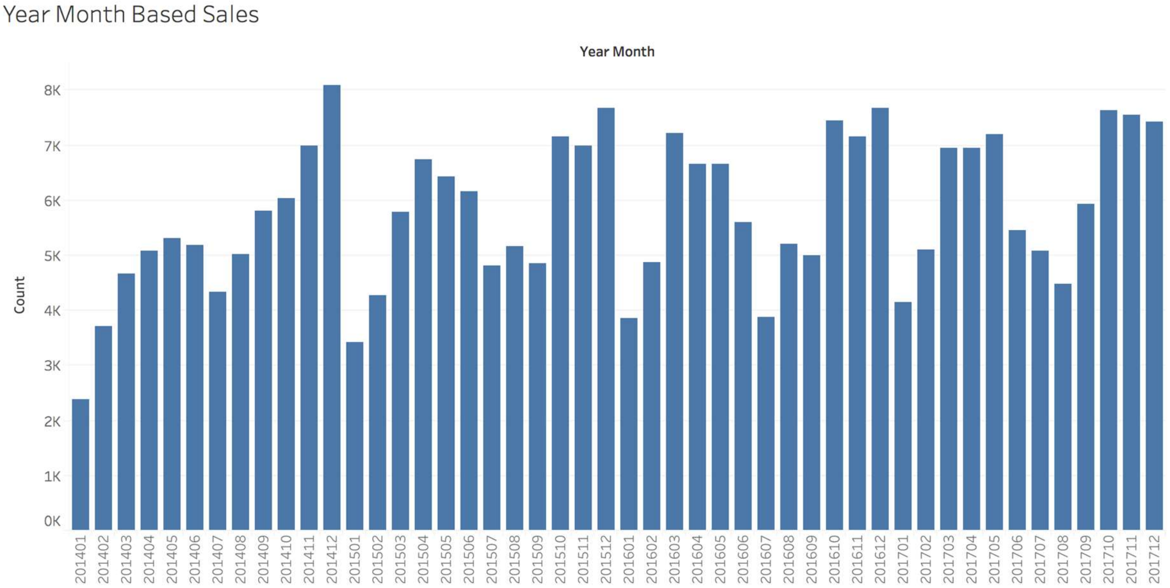
3.1. Summary Statistics

In this section, I examined all the columns from different perspectives. Especially, I analyzed how the number of tractors sold changes based on other variables. You can see the summary statistics of the count column as below. The most remarkable value was the maximum number of count column which was 115 because when we compared to the mean, median, and standard deviation, it was pretty high.

Table 1. Summary Statistics for COUNT column.

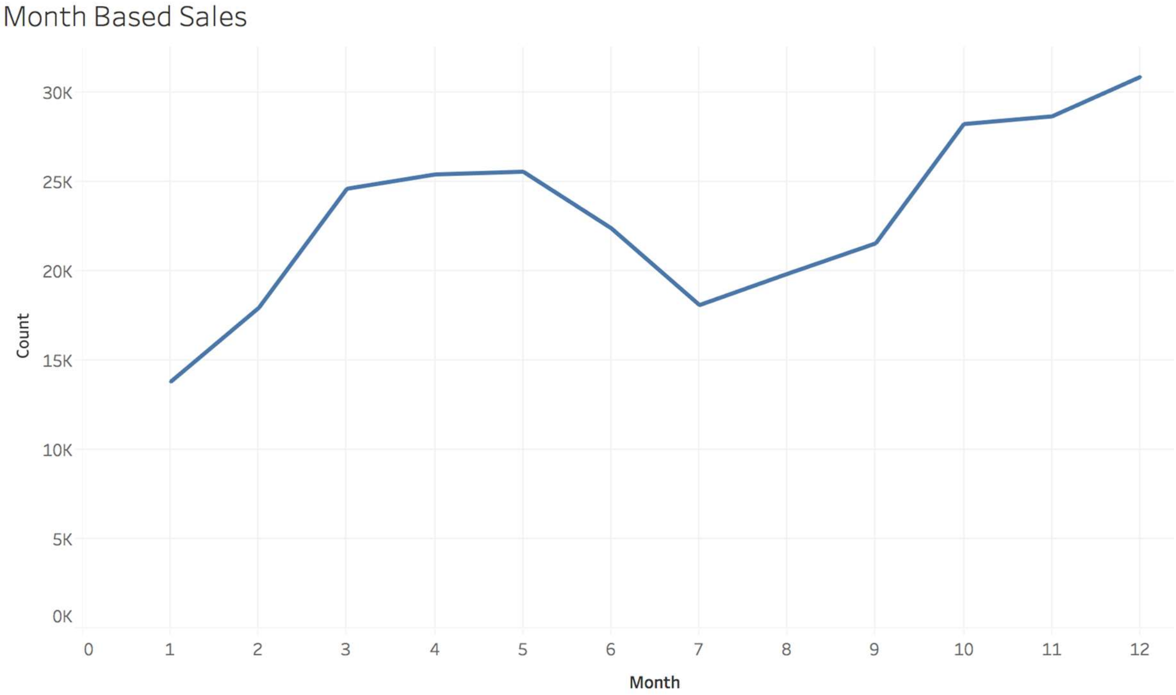
COUNT	
Mean	1.2229
Median	1
Min	1
Max	115
Standard Deviation	0.8731
Unique Values	37
Missing Values	0

Figure 1. Year Month Based Tractor Sales



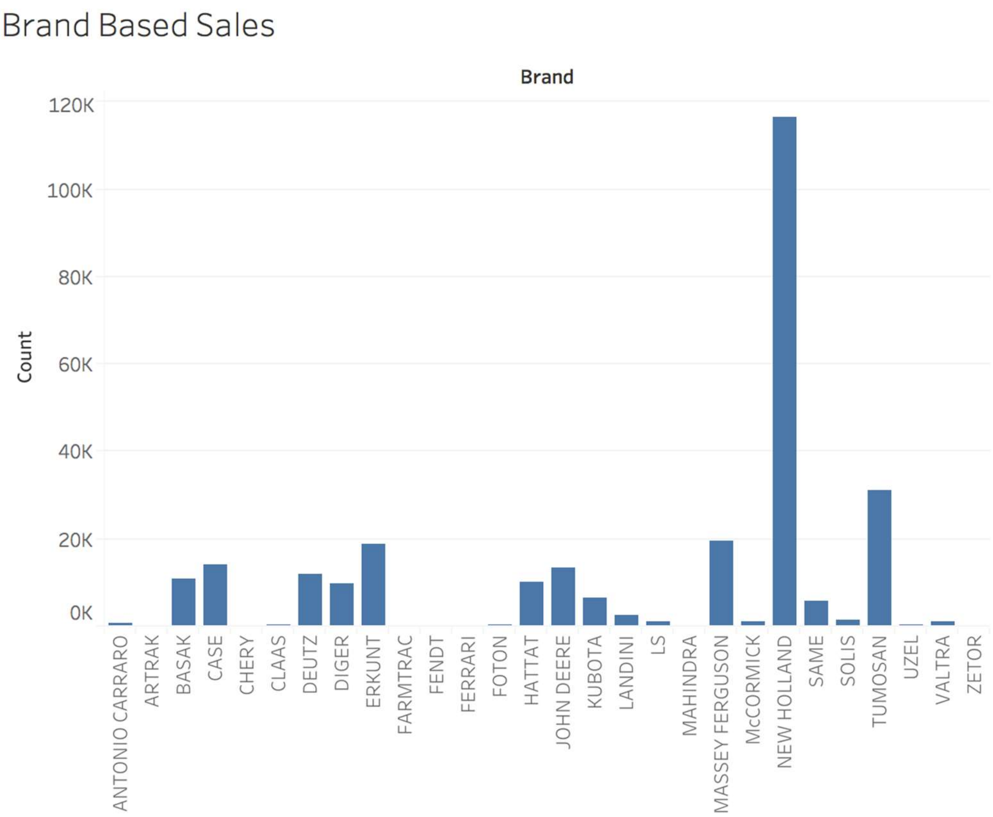
When the year month based sales were analyzed as above, the number of sales is low at the beginning of the year. Conversely, the number of sales slightly goes up at the end of the year, especially in the last month.

Figure 2. Month Based Tractor Sales



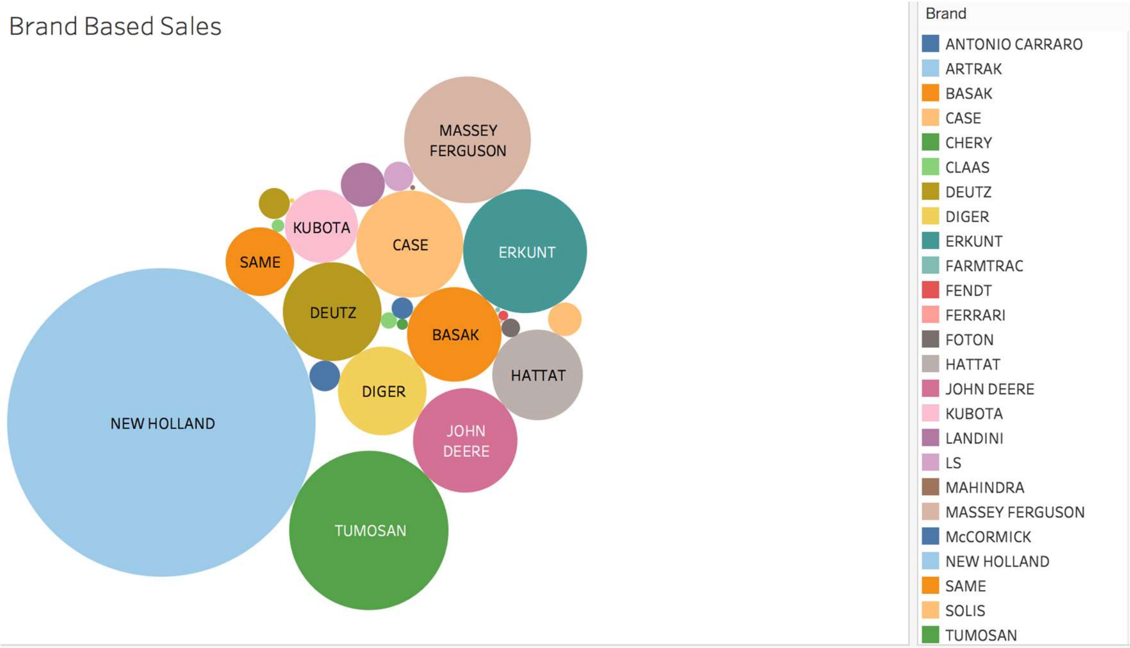
When the sales based on month was analyzed, the ups and downs are shown up more clearly. Companies have some though sales targets every year so the hike in last months can be explained with that.

Figure 3. Brand Based Tractor Sales



As it can be seen, New Holland is the leader of the sector by far and away. Besides, some brands have too few sales and one of the brands was unknown which was called "DIGER" so it can be excluded before creating a model related to Brand.

Figure 4. Brand Based Sales 2



From a different point of view, the magnitudes of sales based on brand can be seen.

All the categorical variables were examined one by one to figure out data anomalies. After the exploration of the dataset, it was seen that the maximum value which was found in summary statistics was abnormal because of missing brands. It was mentioned how anomalies and missing values were handled in the data cleaning part.

3.2. Data Cleaning

City names were consisting of duplicate values because of wrong typing or space in value so they were fixed by the right ones. City code and city names were not matching so they were updated based on Turkish plate codes. The model year of tractors was consisting of some abnormal values like “2013-“ instead of 2013. They were updated with the proper values. Gear column had one value which was including space. It was updated with the ordinary value. Some of the columns had some null values. The model would work based on brand and model of the tractor and that columns cannot include null values so they were removed. There were some unknown brand names called “DIGER” in the dataset. It would be pointless to add those rows to the model which was included the brand feature so they were subtracted from the dataset. After removing these records, the remaining number of rows was 217845. Basak tractor had a region information in order to track their branches. That information added to the dataset. Origin column had some wrong matches with the brand names so they were updated according to their real origins.

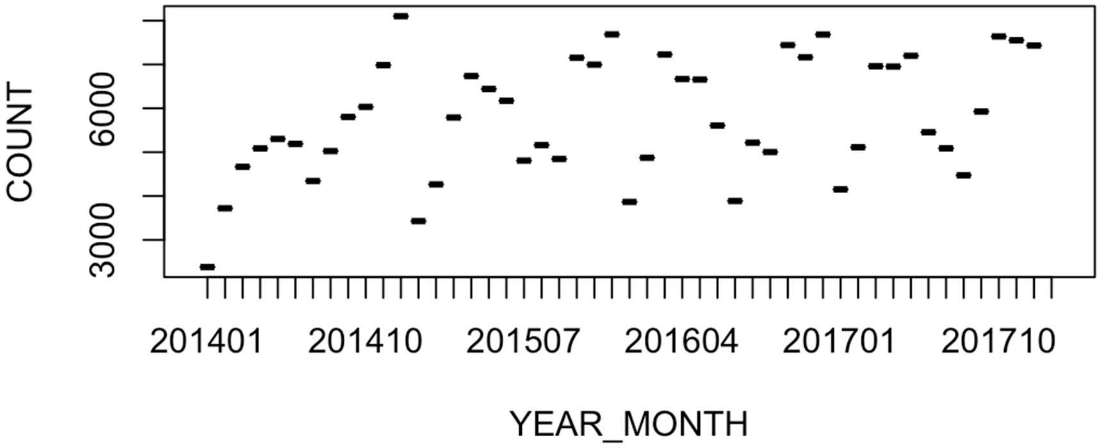
4. METHODOLOGY AND RESULTS

4.1. Simple ARIMA Model

ARIMA (Autoregressive Integrated Moving Average) is one of the commonly used technique to fit time series data and forecasting. ARIMA was the first forecasting method in this project. The number of tractors sold by grouping year and month before creating an ARIMA model were summed up.

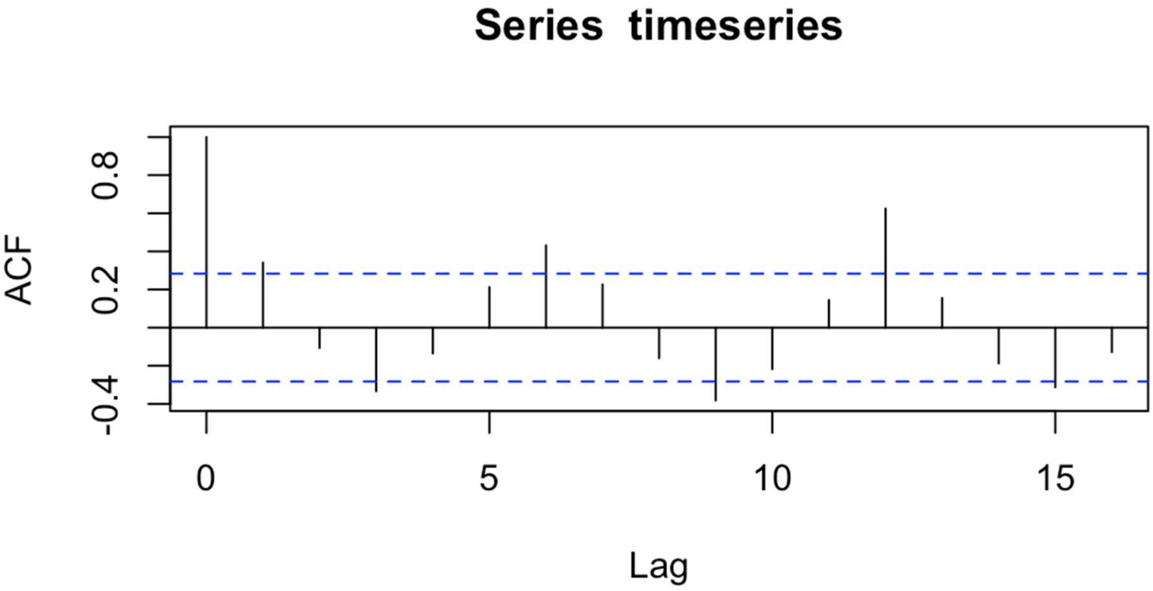
As it was mentioned in the literature review part, ARIMA involves defining three variables *p, d, q* which states the ARIMA model. First of all, the stationarity of the data was checked to determine *d* value. As it was plotted year and month based sales in Figure 5, it can be seen that the data was stationary. That means the *d* value must be zero.

Figure 5. Stationarity



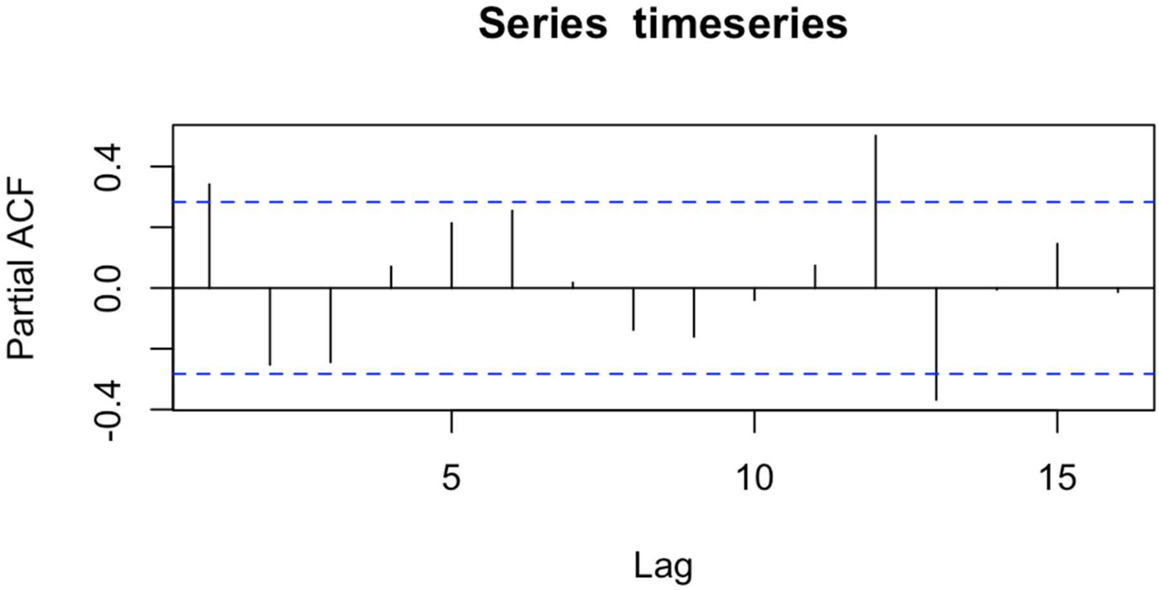
After defining the *d* value, *p* and *q* values should be stated. To do this, it was plotted the ACF (Autocorrelation Function) in Figure 6 and PACF (Partial Autocorrelation Function) in Figure 7. ACF and PACF are measures of contiguity between current and past series values. Also, they indicate which past series values are most beneficial in predicting future values. The order of processes in an ARIMA model with that knowledge can be decided. ACF is the correlation between series values that are *k* intervals apart at lag *k.* PACF is the correlation between series values that are *k* intervals apart, accounting for the values of the intervals between at lag *k.*

Figure 6. Autocorrelation Function



Parameter *p* is the order of AR (Auto Regression). AR is a class of linear model where the variable of interest retreated on its own lagged values. As it is plotted the ACF graph above, it can be seen that the ACF cuts off after lag 2. That means the *p* parameter might be 2.

Figure 7. Partial Autocorrelation Function



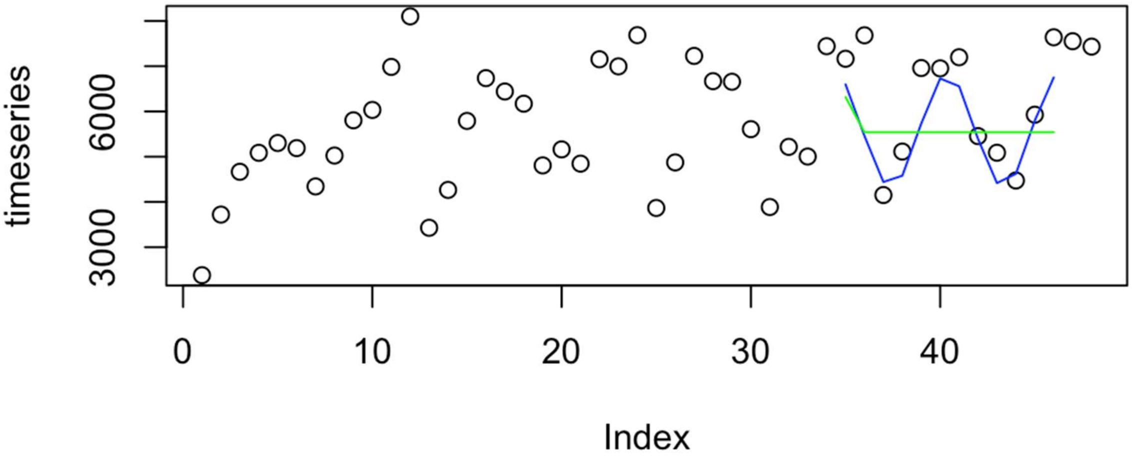
Parameter *q* is the order of MA (Moving Average). MA has a similar form to the classic linear regression of another class. The output or the variable of interest is modeled via its own wrongfully predicted values of current and past times. As the PACF is plotted in

Figure 7, it can be seen that the PACF cuts off after lag 2. That means the *q* parameter might be 2 as well.

An ARIMA model was created using the year and month based sales data. Firstly, the first 34 data points are split as a train set which is almost 70% of the data. That corresponds to the first 34 months of the data between 2014 and 2016. The 14 data points are left for the test set which corresponds to 14 months. The model for ARIMA(2,0,2) was run. Also, a model was created by using auto.arima function which tries to find the best parameters. It initially searches for a range *p* and *q* values, after restoring *d* parameter by Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. It decides the parameters which have the lowest AIC (Akaike Information Criterion). AIC is a commonly used measure of a statistical model. It is an estimator of the relative quality of statistical models for a given dataset. That score does not tell whether the model is good or not. It is for comparing two models and the one with lower AIC score means usually better.

The auto ARIMA model was presented ARIMA(0,0,1). When the AIC scores were compared, the first one was 581.15 and the second one was 592.08. It is seemed that the model had a lower AIC score but when the predictions were visualized as below, the green line indicates the auto ARIMA and the blue line indicates the model which we created.

Figure 8. ARIMA Model Results



4.2. Multiple Linear Regression Model

As a beginning of using regression algorithms, it is tried to select a basic model which is multiple linear regression to predict tractor sales based on year-month and brand features.

Linear regression aims to find the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an independent variable and the other one is considered to be a dependent variable. Before fitting a linear model, the variables must be examined if there is any relationship between them. A scatterplot can be helpful to determine the strength of the relationship between two variables. Similarly, multi linear regression is explained the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical. Multiple linear regression fits a line through a multi-dimensional space of data points, unlike linear regression. The model for multiple linear regression can be defined as *yi = B0 + B1xi1 + B2xi2 + ... + Bpxip + E. yi* stands for dependent variable. *B0* stands for *y*-intercept at time zero. *xi1* stands for independent variable. *B1* stands for a regression coefficient that measures a unit change in the dependent variable when *xi1* changes. *E* stands for random error in prediction, that is a variance that cannot be accurately predicted by the model. Also known as residuals. When it is used multilinear regression, it should be known that the model is based on some assumptions as below.

- There is a linear relationship between the dependent variables and the independent variables.
- The independent variables are not too highly correlated with each other.
- *yi* observations are selected independently and randomly from the population.
- Residuals should be normally distributed.

There are different areas to use multiple linear regression. One of them is to predict trends and future values like in our case.

It is created the model in R. As it was mentioned above, year - month and brand features were used to predict the number of tractors sold. Some of the brands were excluded which had too few data points. We had 528 data points after normalizing the dataset. 70% of the data was split as the training set. After the model was run, the results for the training set were shown as below.

Residual standard error: 157.3 on 311 degrees of freedom.

Multiple R squared: 0.9543

Adjusted R squared: 0.9459

F-statistic: 114 on 57 and 311 DF, p-value: < 2.2e-16.

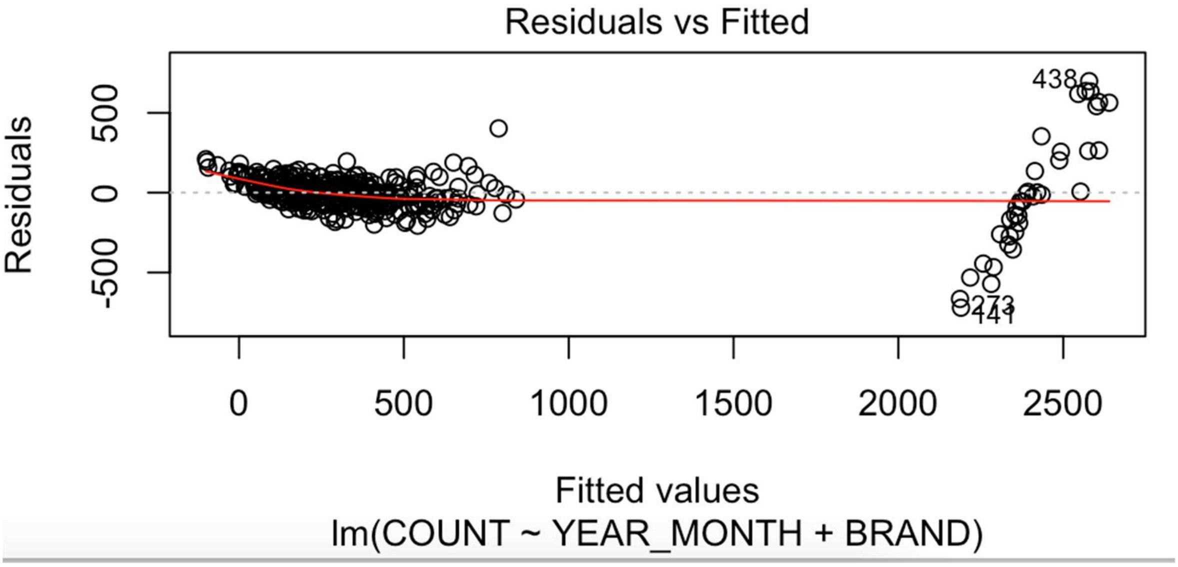
The summary results were fairly well when the obtained numbers were observed. The R squared presents that how well the model is fitting the actual data. The measure explains the observed variance in the response variable if it is close to 1. In the example, the response variable can be explained by the predictor variables, and the p-value is too small. In general, the null hypothesis were rejected when the p-value is less than the level of significance of the test. Also ANOVA test was used to find critical value. Critical value is the point on the scale of the test statistic beyond which the null hypothesis is rejected for the level of significance *a* of the test which was used 0.05. Our test results can be seen in Table 2. Our critical value is much smaller than our F statistic value. On the other hand, our residual standard error is quite high to predict right values. There is a relationship between the independent variables and the dependent variable but the predicted values were not close to the actuals.

Table 2. ANOVA Test Results

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	Critical Value
YEAR_MONTH	47	9640612	205119.41	8.294441	5.505302e-33	2.643221
BRAND	10	151031150	15103115.03	610.726661	7.932963e-198	2.643221
Residuals	311	7690951	24729.75	NA	NA	2.643221

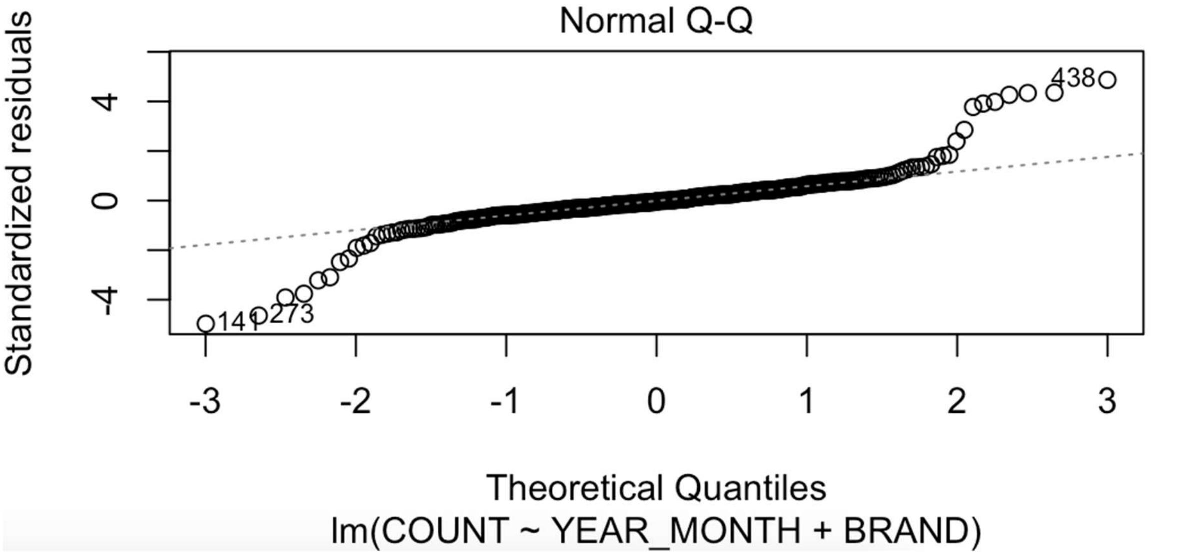
The model assumptions have been checked by examining plots of the residuals or errors. To do so plot function is used in R. As you residuals vs fitted graph can be seen in Figure 9, the relationship between the number of tractors sold which is “COUNT” column and year-month and brand is approximately linear but there are some outliers.

Figure 9. Residuals vs Fitted Plot



The second graph that we plotted is called Q-Q or quantile-quantile plot in Figure 10. The Y-axis is the ordered, observed, standardized residuals. On the X-axis is the ordered theoretical residuals. This is what would be expected the residuals to be if the errors or the residuals are truly normally distributed. If the Y values or errors or residual terms are normally distributed, these points should fall roughly on a diagonal line. So it is seen that the data is close to normal distribution but there are some far data points from the line.

Figure 10. The Quantile-Quantile Plot



4.3. Boosted Decision Tree Regression

All the categorical variables were examined in the data exploration part and it is seen that there are too much distinct values for “Town” and “Model” variable so it would be inappropriate to use them in a model. The content was so detailed and there were not many sales in some towns or models. It was decided to ignore them before creating a model. Besides, some basic machine learning models were created to see the effect of variables which was related to the model features like gear, cabin type, and model year. It was seen that the variables did not differentiate in a reasonable way and it did not affect the model results in a good way. After all, it was decided to ignore them as well.

Although some of the categorical variables were excluded, there were still categorical features like brand, city, HP segment and region. After those columns were examined, it was decided to use decision tree algorithms because there were some clear data points which could be separated easily. For instance, there were some cities which were distinctive when the sales were considered. It was decided to design our models in the Azure platform and boosted decision tree algorithm was picked at first.

The boosted decision tree algorithm creates an ensemble of regression trees using boosting. An ensemble is just a collection of predictors which come together to give a final prediction. It helps us to reduce variance and bias. The term of boosting refers to a family of algorithms which transforms weak learner to strong learners. In Azure Machine Learning Studio, the MART gradient boosting algorithm was implemented. This system trains many models sequentially. Each model gradually minimizes the loss function of the whole system using that method. It finally presents a new fitted model which provides the more accurate estimate of the response variable.

It was decided to exclude some of the brands which were too few data points because it was going to cause noise. Also, some of the models were discarded which had missing values. Related features which were the year, month, brand, city, region and HP segment were selected for the model. Then, dummy features were created by using the brand, city, month, region and HP segment variables because they usually have a higher efficiency when variables have a high dependency on the class label. Regressions cannot naturally deal with qualitative data. It is useful because they enable us to use a single regression equation to represent multiple groups. That usually gives us better results and higher performances in our models. After new features were created by using “Convert to Indicator Values” function in Azure, 121 new features were remained. After that, PCA (Principal Component Analysis) was used to decrease the number of features. PCA is a technique that is used for identification of a smaller number of uncorrelated variables known as principal components from a larger set of data. The purpose is to reduce features dimensionality while only losing a small amount of information. 10 new features were created by using the default parameters in Azure. After the PCA step, 70% of the data were split as the training set. The default parameters were used in the model creation part as well.

As it can be seen in Table 3, the first model results are fairly well according to numbers. Mean absolute error and root mean squared error are quite low. And the R-squared which is also known as the coefficient of determination is fairly good that is 0.60. When the model predictions were examined, it saw seen that the model underestimated some of the large numbers in the dataset.

Table 3. Model 1 Results

Mean Absolute Error	1.608786
Root Mean Squared Error	2.884781
Relative Absolute Error	0.64678
Relative Squared Error	0.393824
Coefficient of Determination	0.606176

Secondly, the same algorithm were created by changing the model parameters. Tuning for model parameters, Azure has a perfect function which is "Tune Model Hyperparameters". The module builds and tests various models by using a different combination of settings that you define, and compares to results which are generated from those combinations. It learns an optimal set of hyper parameters which might be different for each decision tree or dataset. There are three types of parameter sweeping mode. The first one is the entire grid which loops over a grid predefined by the system to try different combinations and select the best option. It is beneficial for cases where you do not know the best parameter settings and you want to try all the possible combination of values. The second one is the random grid which you can reduce the size of the grid. The third one is the random sweep which will randomly pick parameter values over a system-defined range. You decide the maximum number of runs that the module executes. This one is useful when you both want performance and parameter tuning. The third option was selected due to performance issues. The maximum number of runs on the random sweep was selected as 20. It can be seen the second model results in Table 4. Better results were obtained when it was compared to the first model. R-squared has risen 6% and the error terms dropped relatively. Moreover, the predicted values were much better than the first model.

Table 4. Model 2 Results

Mean Absolute Error	1.51596
Root Mean Squared Error	2.673564
Relative Absolute Error	0.609461
Relative Squared Error	0.338266
Coefficient of Determination	0.661734

4.4. Decision Forest Regression

Decision forest is a regression model based on an ensemble of decision trees. Each tree in a regression forest outputs a Gaussian distribution as a prediction. An aggregation is performed over the ensemble of trees to find a Gaussian distribution closest to the combined distribution for all trees in the model. [18] It was thought that it would be a good idea to create a different ensemble model with different types of parameters in Azure. The model has these advantages like efficiency in computation and memory usage during training and prediction, representing non-linear decision boundaries, performing integrated feature selection and classification. The bagging method was selected for resampling. Bagging is a general procedure that can be used to decrease the variance for those algorithms that have high variance. In Azure, each tree in a regression decision forest outputs a Gaussian distribution via prediction. The collection is to find a Gaussian whose first two moments match the moments of the blending of Gaussians given by combining all Gaussians returned by individual trees. Unlike the boosted decision tree algorithm, we picked a different training data set and parameter tuning method. The data was split based on the year which the first three years were selected for training set and the last year was selected for the test data. The reason was to observe how good the next year's sales can be estimated with the data that we already had. For the parameter sweeping method, the entire grid was selected to lift the accuracy by maximizing the coefficient of determination. The results were not as good as we expected. When the predicted results were examined, it was seen that the model underestimated a huge number of tractor sales.

Table 5. Model 3 Results

Mean Absolute Error	1.60393
Root Mean Squared Error	3.271168
Relative Absolute Error	0.717366
Relative Squared Error	0.690692
Coefficient of Determination	0.309308

5. CONCLUSION

In this study, two different ARIMA models and multiple linear regression, decision boosted tree regression and decision forest regression algorithms were applied to predict tractor sales. The models were created from the general to the specific. The ARIMA models were attempted to forecast year-month based sales in order to have a general overview. In the ARIMA models, it was inconvenient to use all the variables that we had so we tried to create a basic model using the year-month variable. When it was compared to auto.arima and the ARIMA model which the parameters were defined by us, our model results were shown better than the auto.arima. However, it seemed us the model was overfitting. On the other hand, it was hard to say that auto.arima results can be used to predict next year's tractor sales. In the multiple linear regression model, it was really though to use categorical variables to get a reasonable result so it was used only year-month and brand variables to predict tractor sales. We obtained the multiple R squared as 0.9543 and the adjusted R squared as 0.9459 in our model statistics. Even though the statistical results seemed well, the predicted values were underwhelming.

Different types of decision tree algorithms were experienced in Azure. The best results were obtained in boosted decision tree algorithm by tuning the parameters. We concluded that the boosting method was more efficient in our case than the bagging method in parameter tuning. It may be because the bagging generates the learners independently, but the boosting tries to add new features that do well where previous models fail. We acquired quite good results in some brands which did not have abnormal sales in some cities or in some months. That kind of sales caused underestimation in our model. Overall predicted sales were almost the same as our actual sales. The model results comparisons can be seen as below.

Table 6. Azure Model Results Comparisons

	Boosted Decision Tree Regression	**Boosted Decision Tree Regression With Tuning Parameters**	**Decision Forest Regression**
Mean Absolute Error	1.608786	1.51596	1.60393
Root Mean Squared Error	2.884781	2.673564	3.271168
Relative Absolute Error	0.64678	0.609461	0.717366
Relative Squared Error	0.393824	0.338266	0.690692
Coefficient of Determination	0.606176	0.661734	0.309308

As a result, it can be said that if the company wants to estimate market sales, the statistical models will assist them. However, if the company wants to predict sales more detailed like brand or city based, the machine learning models underestimate the number of sales so it will not be a proper choice to use them. In the further steps, it can be searched why some of the brands were sold much more in some cities or in some months. If that reason can be explained, they can add some sensible variables to model in order to get better results.

6. REFERENCES

[1] http://www.invest.gov.tr/en-US/sectors/Pages/Agriculture.aspx

[2] http://www.nationsencyclopedia.com/economies/Asia-and-the-Pacific/Turkey-AGRICULTURE.html#ixzz5Jzfir4k6

[3] Chand, S. Business Forecasting and its Importance to Business. http://www.yourarticlelibrary.com/business/business-forecasting-and-its-importance-to-business/25634

[4] Nikhila, C. http://www.businessmanagementideas.com/sales/forecasting-sales/sales-forecasting-meaning-importance-and-methods/7122

[5] Metcalf,T. (2018, April 5) Top 10 Reasons Why Sales Forecasting Is Important. https://yourbusiness.azcentral.com/top-10-reasons-sales-forecasting-important-24818.html

[6] Scott,S.L (2017, July 11) Fitting Bayesian structural time series with the bsts R package. http://www.unofficialgoogledatascience.com/2017/07/fitting-bayesian-structural-time-series.html

[7] Dalinina,R. (2017, October 1) Introduction to Forecasting with ARIMA in R. https://yourbusiness.azcentral.com/top-10-reasons-sales-forecasting-important-24818.html

[8] Thirunav,A. (2016, November 26) What is ARIMA?. https://www.quora.com/What-is-ARIMA

[9] Sujjaviriyasup, T., Pitiruek.K. (2013). Hybrid ARIMA-Support Vector Machine Model for Agricultural Production Planning. Applied Mathematical Sciences, Vol. 7, 2013, no. 57, 2833 – 2840

[10] Kim, B., Shin,S., Kim,T., Yum,S., Kim, J. (2013). Forecasting Demand of Agricultural Tractor, Riding Type Rice Transplanter and Combine Harvester by using an ARIMA Model

[11] Strohecker,K. (2018, March 23) Graphic: Turkey's economic troubles in five charts. https://www.reuters.com/article/us-turkey-markets-currency/graphic-turkeys-economic-troubles-in-five-charts-idUSKBN1GZ2MK

[12] OECD ECONOMIC OUTLOOK, VOLUME 2018 http://www.oecd.org/eco/outlook/economic-forecast-summary-turkey-oecd-economic-outlook.pdf

[13] Khan. R. (2017, December 31) ARIMA model for forecasting– Example in R https://rpubs.com/riazakhan94/arima_with_example

[14] Keshvani, Abbas. (2013, August 14) USING AIC TO TEST ARIMA MODELS https://coolstatsblog.com/2013/08/14/using-aic-to-test-arima-models-2/

[15] Weisberg, S. (2005). Applied Linear Regression. New Jersey: John Wiley & Sons, Inc..

[16] Rego, F. (2015, October 23) QUICK GUIDE: INTERPRETING SIMPLE LINEAR MODEL OUTPUT IN R. https://feliperego.github.io/blog/2015/10/23/Interpreting-Model-Output-In-R

[17] (2018, January 11) Boosted Decision Tree Regression. https:// docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/boosted-decision-tree-regression

[18] (2018, January 16) Decision Forest Regression. https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/decision-forest-regression

APPENDIX A

```
# Reading the data
tractor <- read.csv("/Users/Tunay/Desktop/TRAKTÖR/Tractor.csv")
region_lkp <- read.csv("/Users/Tunay/Desktop/TRAKTÖR/region_lkp.csv")

# Libraries
library(sqldf)
library(ggplot2)
library(dplyr)
library(varhandle)
library(mlr)
library(forecast)

# Summary of the data
summary(tractor)
#Getting the variables' names and data types
str(tractor)
# Data Cleaning
# Updating MONTH column
tractor$MONTH[tractor$MONTH=="1"] <- "01"
tractor$MONTH[tractor$MONTH=="2"] <- "02"
tractor$MONTH[tractor$MONTH=="3"] <- "03"
tractor$MONTH[tractor$MONTH=="4"] <- "04"
tractor$MONTH[tractor$MONTH=="5"] <- "05"
tractor$MONTH[tractor$MONTH=="6"] <- "06"
```

```
tractor$MONTH[tractor$MONTH=="7"] <- "07"
tractor$MONTH[tractor$MONTH=="8"] <- "08"
tractor$MONTH[tractor$MONTH=="9"] <- "09"
tractor$YEAR_MONTH <- paste0(tractor$YEAR,tractor$MONTH)
tractor$YEAR_MONTH <- as.factor(tractor$YEAR_MONTH)
table(tractor$YEAR_MONTH)
sqldf("select YEAR_MONTH,COUNT(*) from tractor GROUP BY YEAR_MONTH ")
tractor <- subset (tractor, !YEAR_MONTH=="NANA")
sqldf("select distinct [CITY.CODE], CITY from tractor ")
sqldf("select distinct CITY from tractor ")
# There are some missing and wrong data in CITY.CODE. Let's update them.
tractor$CITY.CODE[tractor$CITY=="ADANA"] <- "1"
tractor$CITY.CODE[tractor$CITY=="ADIYAMAN"] <- "2"
tractor$CITY.CODE[tractor$CITY=="AFYON"] <- "3"
tractor$CITY.CODE[tractor$CITY=="AĞRI"] <- "4"
tractor$CITY.CODE[tractor$CITY=="AMASYA"] <- "5"
tractor$CITY.CODE[tractor$CITY=="ANKARA"] <- "6"
tractor$CITY.CODE[tractor$CITY=="ANTALYA"] <- "7"
tractor$CITY.CODE[tractor$CITY=="ARTVİN"] <- "8"
tractor$CITY.CODE[tractor$CITY=="AYDIN"] <- "9"
tractor$CITY.CODE[tractor$CITY=="BALIKESİR"] <- "10"
tractor$CITY.CODE[tractor$CITY=="BİLECİK"] <- "11"
tractor$CITY.CODE[tractor$CITY=="BİNGÖL"] <- "12"
tractor$CITY.CODE[tractor$CITY=="BİTLİS"] <- "13"
tractor$CITY.CODE[tractor$CITY=="BOLU"] <- "14"
tractor$CITY.CODE[tractor$CITY=="BURDUR"] <- "15"
```

```
tractor$CITY.CODE[tractor$CITY=="BURSA"] <- "16"

tractor$CITY.CODE[tractor$CITY=="ÇANAKKALE"] <- "17"

tractor$CITY.CODE[tractor$CITY=="ÇANKIRI"] <- "18"

tractor$CITY.CODE[tractor$CITY=="ÇORUM"] <- "19"

tractor$CITY.CODE[tractor$CITY=="DENİZLİ"] <- "20"

tractor$CITY.CODE[tractor$CITY=="DİYARBAKIR"] <- "21"

tractor$CITY.CODE[tractor$CITY=="EDİRNE"] <- "22"

tractor$CITY.CODE[tractor$CITY=="ELAZIĞ"] <- "23"

tractor$CITY.CODE[tractor$CITY=="ERZİNCAN"] <- "24"

tractor$CITY.CODE[tractor$CITY=="ERZURUM"] <- "25"

tractor$CITY.CODE[tractor$CITY=="ESKİŞEHİR"] <- "26"

tractor$CITY.CODE[tractor$CITY=="GAZİANTEP"] <- "27"

tractor$CITY.CODE[tractor$CITY=="GİRESUN"] <- "28"

tractor$CITY.CODE[tractor$CITY=="GÜMÜŞHANE"] <- "29"

tractor$CITY.CODE[tractor$CITY=="HAKKARİ"] <- "30"

tractor$CITY.CODE[tractor$CITY=="HATAY"] <- "31"

tractor$CITY.CODE[tractor$CITY=="ISPARTA"] <- "32"

tractor$CITY.CODE[tractor$CITY=="İÇEL"] <- "33"

tractor$CITY.CODE[tractor$CITY=="İSTANBUL"] <- "34"

tractor$CITY.CODE[tractor$CITY=="İZMİR"] <- "35"

tractor$CITY.CODE[tractor$CITY=="KARS"] <- "36"

tractor$CITY.CODE[tractor$CITY=="KASTAMONU"] <- "37"

tractor$CITY.CODE[tractor$CITY=="KAYSERİ"] <- "38"

tractor$CITY.CODE[tractor$CITY=="KIRKLARELİ"] <- "39"

tractor$CITY.CODE[tractor$CITY=="KIRŞEHİR"] <- "40"

tractor$CITY.CODE[tractor$CITY=="KOCAELİ"] <- "41"
```

```
tractor$CITY.CODE[tractor$CITY=="KONYA"] <- "42"

tractor$CITY.CODE[tractor$CITY=="KÜTAHYA"] <- "43"

tractor$CITY.CODE[tractor$CITY=="MALATYA"] <- "44"

tractor$CITY.CODE[tractor$CITY=="MANİSA"] <- "45"

tractor$CITY.CODE[tractor$CITY=="K.MARAŞ"] <- "46"

tractor$CITY.CODE[tractor$CITY=="MARDİN"] <- "47"

tractor$CITY.CODE[tractor$CITY=="MUĞLA"] <- "48"

tractor$CITY.CODE[tractor$CITY=="MUŞ"] <- "49"

tractor$CITY.CODE[tractor$CITY=="NEVŞEHİR"] <- "50"

tractor$CITY.CODE[tractor$CITY=="NİĞDE"] <- "51"

tractor$CITY.CODE[tractor$CITY=="ORDU"] <- "52"

tractor$CITY.CODE[tractor$CITY=="RİZE"] <- "53"

tractor$CITY.CODE[tractor$CITY=="SAKARYA"] <- "54"

tractor$CITY.CODE[tractor$CITY=="SAMSUN"] <- "55"

tractor$CITY.CODE[tractor$CITY=="SİİRT"] <- "56"

tractor$CITY.CODE[tractor$CITY=="SİNOP"] <- "57"

tractor$CITY.CODE[tractor$CITY=="ŞIRNAK"] <- "58"

tractor$CITY.CODE[tractor$CITY=="TEKİRDAĞ"] <- "59"

tractor$CITY.CODE[tractor$CITY=="TOKAT"] <- "60"

tractor$CITY.CODE[tractor$CITY=="TRABZON"] <- "61"

tractor$CITY.CODE[tractor$CITY=="TUNCELİ"] <- "62"

tractor$CITY.CODE[tractor$CITY=="ŞANLIURFA"] <- "63"

tractor$CITY.CODE[tractor$CITY=="UŞAK"] <- "64"

tractor$CITY.CODE[tractor$CITY=="VAN"] <- "65"

tractor$CITY.CODE[tractor$CITY=="YOZGAT"] <- "66"

tractor$CITY.CODE[tractor$CITY=="ZONGULDAK"] <- "67"
```

```
tractor$CITY.CODE[tractor$CITY=="AKSARAY"] <- "68"
tractor$CITY.CODE[tractor$CITY=="BAYBURT"] <- "69"
tractor$CITY.CODE[tractor$CITY=="KARAMAN"] <- "70"
tractor$CITY.CODE[tractor$CITY=="KIRIKKALE"] <- "71"
tractor$CITY.CODE[tractor$CITY=="BATMAN"] <- "72"
tractor$CITY.CODE[tractor$CITY=="SİVAS"] <- "73"
tractor$CITY.CODE[tractor$CITY=="BARTIN"] <- "74"
tractor$CITY.CODE[tractor$CITY=="ARDAHAN"] <- "75"
tractor$CITY.CODE[tractor$CITY=="IĞDIR"] <- "76"
tractor$CITY.CODE[tractor$CITY=="YALOVA"] <- "77"
tractor$CITY.CODE[tractor$CITY=="KARABÜK"] <- "78"
tractor$CITY.CODE[tractor$CITY=="KİLİS"] <- "79"
tractor$CITY.CODE[tractor$CITY=="OSMANİYE"] <- "80"
tractor$CITY.CODE[tractor$CITY=="DÜZCE"] <- "81"
# Updating some of the unstructured model years.
sqldf("select distinct [MODEL.YEAR] from tractor ")
tractor$MODEL.YEAR[tractor$MODEL.YEAR=="2013-"] <- "2013"
tractor$MODEL.YEAR[tractor$MODEL.YEAR=="2014-"] <- "2014"
# Checking the updates
sqldf("select distinct [CITY.CODE], CITY from tractor ")
sqldf("select distinct [MODEL.YEAR] from tractor ")
# Reformating features


tractor$F_CITY_CODE <- as.factor(tractor$CITY.CODE)
# Checking the updates
sqldf("select count(*), [MODEL.YEAR] from tractor group by [MODEL.YEAR] ")
sqldf("select count(*), CITY from tractor group by CITY ")

# Examining some of the columns

summary(tractor$GEAR)

# Updating GEAR column

tractor$GEAR[tractor$GEAR=="4WD "] <- "4WD"

summary(tractor$CABIN.TYPE)

summary(tractor$HP)

summary(tractor$HP.SEGMENT)

summary(tractor$SEGMENT.DETAIL)

summary(tractor$SEGMENT)

summary(tractor$ORIGIN)

summary(tractor$REGION)

# Updating REGION column

tractor$REGION<-region_lkp[match(tractor$CITY.CODE, region_lkp$CITY.CODE),3]

# ORIGIN column update

sqldf("select count(*), BRAND, ORIGIN from tractor group by BRAND, ORIGIN order
by BRAND asc")

tractor$ORIGIN[is.na(tractor$ORIGIN) & tractor$BRAND =="BASAK"] <- "YERLI"

tractor$ORIGIN[tractor$BRAND =="CASE"] <- "YERLI"

tractor$ORIGIN[tractor$BRAND =="CLAAS"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="DEUTZ"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="FERRARI"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="FOTON"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="HATTAT"] <- "YERLI"

tractor$ORIGIN[tractor$BRAND =="JOHN DEERE"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="KUBOTA"] <- "ITHAL"
```

```
tractor$ORIGIN[tractor$BRAND =="LS"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="MASSEY FERGUSON"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="McCORMICK"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="NEW HOLLAND"] <- "YERLI"

tractor$ORIGIN[tractor$BRAND =="SAME"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="TUMOSAN"] <- "ITHAL"

tractor$ORIGIN[tractor$BRAND =="VALTRA"] <- "ITHAL"

write.csv(tractor, file = "tractor_v2.csv")

# BRAND

summary(tractor$BRAND)

sqldf("select sum(COUNT) AS CNT, BRAND from tractor group by BRAND order by CNT asc")

summary (subset(tractor, BRAND=="DIGER" ))

sqldf("select count(distinct model) AS CNT, BRAND from tractor group by BRAND order by CNT asc")

# We don't have so much information about DIGER brand so

# we can ignore them before we create a model.

tractor_v2 <- subset (tractor, !BRAND=="DIGER")

summary(tractor_v2$CABIN.TYPE)

summary(tractor_v2$HP)

summary(tractor_v2$HP.SEGMENT)

summary(tractor_v2$SEGMENT.DETAIL)

summary(tractor_v2$SEGMENT)



# I am planning to create a model based on "Brand" and "Model", so I need to discard NA rows which "Model" column have.

tractor_v3 <- subset (tractor_v2, !MODEL=="NA")
```

```
subset (tractor_v3, MODEL=="NA")

subset (tractor_v3, CABIN.TYPE=="NA")

subset (tractor_v3, HP=="NA")

subset (tractor_v3, SEGMENT.DETAIL=="NA")

subset (tractor_v3, MODEL.YEAR=="NA")

# Now we don't have any NA columns based on brand and model.

# Creating a basic ARIMA model

tractor_year_month <- sqldf("select YEAR_MONTH, sum(COUNT) AS COUNT from tractor group by YEAR_MONTH")

plot(tractor_year_month)

timeseries=tractor_year_month$COUNT

train_series=timeseries[1:34]

test_series=timeseries[35:48]

arimaModel_1=arima(train_series, order=c(2,0,2))

print(arimaModel_1)

forecast1=predict(arimaModel_1, 14)

AutoArimaModel=auto.arima(train_series )

AutoArimaModel

forecast4= predict(AutoArimaModel, 14)

plot(timeseries)

lines(forecast1$pred,col="blue")

#lines(forecast2$pred,col="red")

#lines(forecast3$pred,col="yellow")

lines(forecast4$pred,col="green")

acf(timeseries)

pacf(timeseries)
```

```
# Creating a Basic Linear Regression Model Based on Year_month and Brand

# Let's check the number of tractor sales based on brand

sqldf("select sum(COUNT) AS COUNT, BRAND

    from tractor

    group by BRAND ORDER BY COUNT desc")

# Some brands have too few data points so they were discarded.

tractor_brand_based <- sqldf("select sum(COUNT) AS COUNT,

                   YEAR_MONTH,BRAND from tractor

where brand NOT IN (
'DIGER','ARTRAK','FERRARI','ZETOR','FARMTRAC','MAHINDRA','FENDT','CHER
Y','UZEL','CLAAS','FOTON','ANTONIO CARRARO',

              'LS','McCORMICK','VALTRA','SOLIS','LANDINI')

                   group by YEAR_MONTH,BRAND")



sqldf("select * from tractor_brand_based order by BRAND asc, YEAR_MONTH asc")

set.seed(100)  # setting seed to reproduce results of random sampling

trainingRowIndex <- sample(1:nrow(tractor_brand_based),
0.7*nrow(tractor_brand_based))  # row indices for training data

trainingData <- tractor_brand_based[trainingRowIndex, ]  # model training data

testData  <- tractor_brand_based[-trainingRowIndex, ]

lmMod <- lm(COUNT ~ YEAR_MONTH+BRAND, data=trainingData)  # build the
model

distPred <- predict(lmMod, testData)  # predict distance

summary (lmMod)

actuals_preds <- data.frame(cbind(actuals=testData$COUNT, predicteds=distPred))  #
make actuals_predicteds dataframe.

actuals_preds

plot(lmMod)
```

```
myanova<- anova(lmMod)

cbind(myanova, 'CriticalValue' = qf(1-.05, myanova[1,1], myanova[2,1]))
```

APPENDIX B

BASAK

CASE

DEUTZ

ERKUNT

HATTAT

JOHN DEERE

KUBOTA

LANDINI

MASSEY

FERGUSON

McCORMICK

NEW HOLLAND

SAME

TUMOSAN

VALTRA

SOLIS

LS

CLAAS

UZEL

ANTONIO CARRARO

FARMTRAC

FOTON

FENDT

FERRARI

ZETOR

MAHINDRA

CHERY

ARTRAK