07/01/2022

Time Series & Logistic Regression Project

Statistics for Data Analytics

Maria Migrova
STUDENT ID: 21146021

Contents

INTRODUCTION

TIME SERIES

A time series is a set of data gathered over time by observing a response variable (y). Time series analysis is a method of analyzing a collection of data points over a period of time. Instead of capturing data points infrequently or arbitrarily, time series analyzers capture them at regular intervals over a predetermined period of time. [1]

Components of a Time Series

We can see some trends in time series plots. These patterns are made up of one or more components that are combined to produce time series data. We can examine each component separately and then aggregate the predictions of each component to generate forecasts. Trend-seasonal analysis is a method for calculating four fundamental components of a time series. The definitions of four components are as follows:

- SECULAR TREND (T) - A long-term growth or reduction in the data. The trend pattern might be linear or non-linear in nature.
- CYCLICAL EFFECT (C) - The progressive up-and-down fluctuation (also known as business cycles) in opposition to the secular trend, in which the variations do not occur at a fixed frequency. The majority of business and commercial data contains a cyclic component that lasts more than a year.
- SEASONAL VARIATION (S) - The cyclical changes that occur at different times of the year (e.g., monthly, or quarterly). We commonly measure seasonal component in quarters in the commercial and finance sphere.
- RESIDUAL EFFECT (R) - Remaining variations remain after the secular, cyclical, and seasonal components have been removed. It's impossible to tell how long the effect will last. [2]

Time series data can be also used for forecasting - predicting future values based on historical values.
There are lots of forecasting techniques for predicting future values of a time series. In this project we will use four different forecasting methods:

- The Mean Forecast Method
- The Naive Method
- The Seasonal Naive Method
- ARIMA Method - The Auto-Regression Integrated Moving Average Method

THE MEAN FORECAST METHOD

For this method, we use the average of the time series to express the forecasts. That's because we assume that every piece of data in a time series is an equally useful predictor of all future values. When a time series does not contain substantial T (secular trend) and S (seasonal variation) components, this method works best. This approach produces stable forecasts, but it may not recognise all patterns.
The mean approach gives us a single number to work with (point forecast). If a period occurs several times over, a point forecast, usually the man of the probability distribution, offers an estimate of the average number of contributions. A prediction interval is a time interval used to forecast the outcome of a single observation. If a future event repeats, a 95 percent prediction interval means that the possibility of a given attribute measured from the event being within the prediction interval is 95 percent.

$$SMA = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

Figure 1 SMA

THE NAIVE METHOD

One of the most basic forecasting techniques is the Naive method. This method ignores the effect of T and S components. The forecast for a certain time period is the value of the previous period when employing this method. The naive method appears to be more effective when data is reported on a daily or weekly basis, or when there are no T or S components.

$$Y_t = Y_{t-1}$$

Figure 2 Naive Method

Here, Y_t is forecast value at time t and Y_{t-1} is value yesterday.

THE SEASONAL NAIVE METHOD

The Naive Method can be modified in a number of ways. In the case of highly seasonal data, the current season's estimate is dependent on the previous season's actual value. The seasonal Naive method is an adaptation of the Naive method. Complex methods, on the other hand, do not ensure correctness.

Seasonal Naïve Model[2] (Padhan, 2012): $\hat{Y}_{t+1} = Y_{t-k}$ Where k is seasonal lag.

Figure 3 Seasonal Naive Method

ARIMA METHOD

ARIMA model is an auto-regressive integrated moving average model. It is a broad category for time series models (models by Box Jenkins). It's a forecasting method in which anticipated values are a linear function of recent actual values and recent prediction mistakes (residuals). The AR(1) coefficient defines how quickly the series tends to revert to its mean in the AR(1) model. The series recovers to its mean fast if the coefficient is near zero, and slowly if the coefficient is near one.

The sum of the AR coefficients affects the pace of mean reversion in a model with two or more AR coefficients, and the series may also show an oscillatory pattern.

$$\text{ARIMA}\ \underbrace{(p,d,q)}_{\substack{\uparrow \\ \text{Non-seasonal part} \\ \text{of the model}}}\ \underbrace{(P,D,Q)_m}_{\substack{\uparrow \\ \text{Seasonal part} \\ \text{of the model}}}$$

Figure 4 ARIMA

Model Evaluation Methods

In this project forecast accuracies are used to determine which method produced a solid forecast. Forecast accuracy can be calculated by looking at forecast errors, which are the discrepancies between actual future values and expected values. Mean absolute percentage error (MAPE), and root mean squared error are the two widely accepted measurements (RMSE). The values of these measures may usually be calculated automatically by most software systems.

- The Mean Absolute Percent Error (MAPE) - For comparing the MAPE, we must first determine the absolute error at each time point, which is then divided by the observed value at that moment. The average of these equations is then written as a percentage.
- The Root Mean Squared Error (RMSE) - This is one of the most useful measurements. We don't want the forecast to have huge inaccuracies, hence we want the measure to be sensitive to them. When we square the error, we get a greater number for a large error and a lower value for a small error.

The RMSE estimates the standard deviation of the prediction error and helps users to determine the expected value, while the MAPE is a metric that compares the degree of prediction error to the true value.

LOGISTIC REGRESSION

One of the most used algorithms in the Supervised Machine Learning technique is logistic regression. It's a method for predicting a categorical dependent variable based on a set of independent factors. The output of a categorical dependent variable is predicted using logistic regression. As a result, the output must be discrete or categorical in nature. It can be Yes or No, 0 or 1, true or false, and so on, but instead of exact numbers like 0 and 1, it delivers probabilistic values that are somewhere in the between. Logistic Regression is extremely similar to Linear Regression in terms of how it is used. Linear Regression is used to tackle regression problems, whereas Logistic

Regression is utilized to solve problems using Logistic Regression. The logistic regression's value must be between 0 and 1, and it cannot exceed this limit, resulting in a "S" curve. The Sigmoid function, often known as the logistic function, is the S-form curve. The concept of the threshold value is used in logistic regression to describe the probability of either 0 or 1. Values over the threshold value tend to be 1, while those below the threshold value tend to be 0. [3]

LOGISTIC REGRESSION ASSUMPTIONS

- The dependent variable needs to be categorical.
- There should be no multi-collinearity in the independent variable.
- Independence of errors
- linearity in the logit for continuous variables [4]

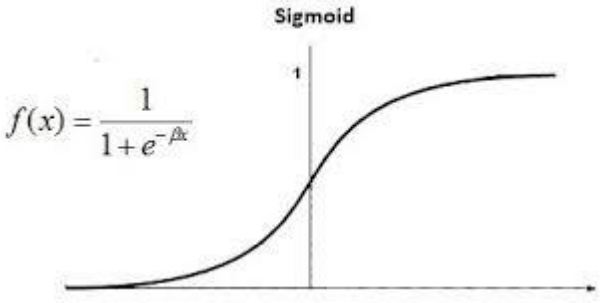


Figure 5 Logistic Regression

METHODOLOGY

DATA DESCRIPTION FOR TIME SERIES PART

We are using the R Studio software for this analysis.
For this project we are using a csv file called eComm_US.csv. This dataset consists of 2 variables of 87 observations. These variables are:

- DATE - Quarterly time series commencing Q4, 1999
- ECOMNSA - United States e-commerce retail sales (in $billions)

DATA PREPARATION

After summarising the data using summarise() function we found out that DATE is in the character format, we changed it into date format using as.Date() function.
ECOMNSA variable is in double format.

VISUALISING THE TIME SERIES DATA

Time Series Visualization Visualization aids in the discovery of a time series' underlying behavior as well as the detection of time series abnormalities. In a time series, we frequently witness multiple distinct patterns. We can then choose relevant analytic methods based on these patterns. Time series can be visualized using a variety of technologies, including R, Python, and Excel. A time series diagram compares each observation to the time when it

was measured, is a common way of presenting time series data. We can save a time series as a time series object in R while performing time series analysis (i.e., a ts object) using ts_ts() function.

MULTIPLICATIVE vs ADDITIVE DECOMPOSITION

Trend and seasonality interactions are usually classified as either additive or multiplicative. In this piece, we'll look at how we may classify a time series as either one or the other to make further processing easier.

Before we go any further, it's critical to understand the distinction between a multiplicative and an additive time series. A time series is made up of three parts as we mentioned earlier. Whether a time series is multiplicative, or additive is determined by the interaction of these three components.

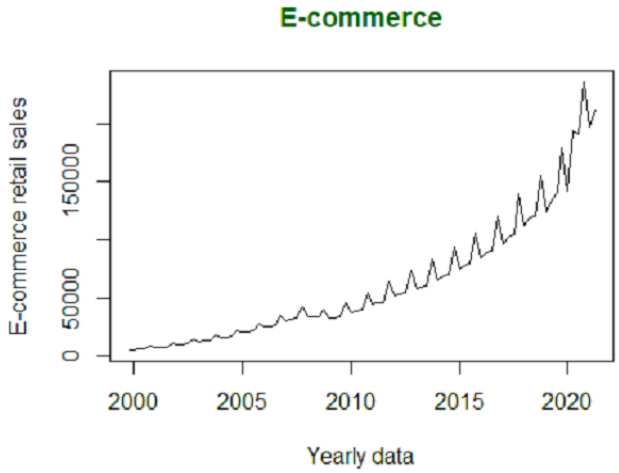


Figure 6 E-commerce sales

The time series is formed by multiplying the components of a multiplicative time series. If the trend is rising, the amplitude of seasonal activity rises. Everything becomes a lot more dramatic at this point. This is a common occurrence while looking at site traffic.

$$y_t = T_t \times S_t \times R_t$$

The time series is formed by adding the components of an additive time series together. If you have a rising trend, the peaks and troughs will be around the same magnitude throughout the time series. In indexed time series, this is typical because the absolute value grows while the changes stay relative.

$$y_t = T_t + S_t + R_t$$

How do I tell if a time series is additive or multiplicative? One method is to just plot the original time series data, which is known as a run-sequence graphic. We have an additive series if the seasonality and residual components are independent of the trend. We have a multiplicative series if the seasonality and residual components are dependent, meaning they fluctuate on trend. The time series must be broken into its components in order to determine if it is additive or multiplicative. [5]

In the above graph we can see multiplicative trend and seasonality, therefore we will use the multiplicative decomposition using decompose() function with type = 'multiplicative'.

In this graph we can see the 4 components. First is the observed, original graph. Second is the trend, then seasonal component and residual component. For this decomposition, we can clearly see seasonality variation, and the residual component is constant. Therefore, the time series is multiplicative.

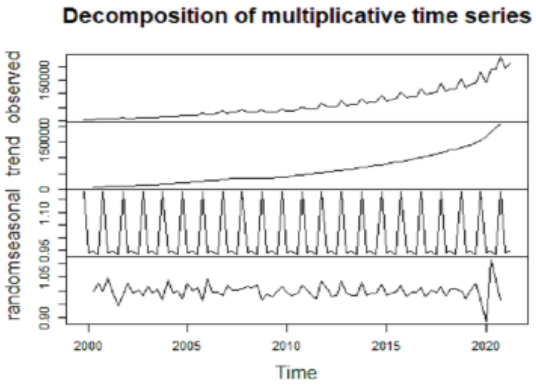


Figure 7 Decomposition

SMOOTHING THE DATA

We started our analysis by smoothing the data using Centered Moving Average Smoothing with order set to 3 using ma() function and then I save it to the actual dataset, which I we will use for the later analysis.

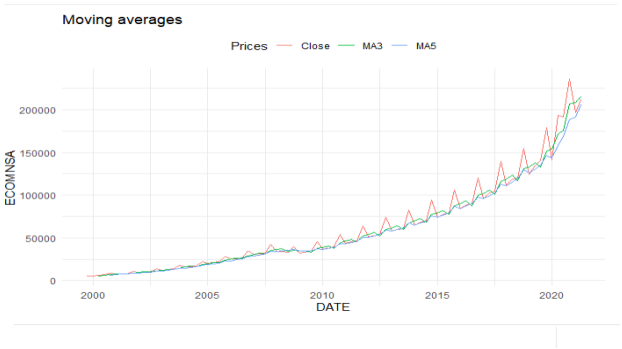


Figure 8 Moving average

Here we can see the smoothed data. The red line represents the actual data before the smoothing, the blue line represents smoothing with order set to 3 and the green line represents smoothing with order set to 5.

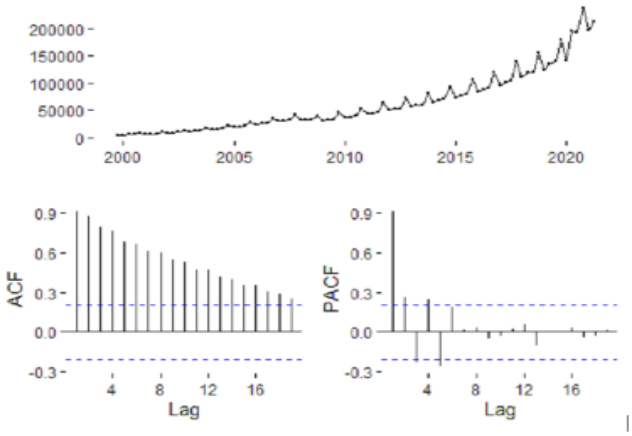


Figure 9 ACF, PACF

After performing a ggtsdisplay() function, we got this plot of the raw time series with Auto-Correlation Function Plot and Correlation Function Plot. We will use these later for the ARIMA model.

DATA DESCRIPTION FOR LOGISTIC REGRESSION PART

We are also using R studio for this analysis.
For this project we are using a sav file called House Categories.sav. This dataset consists of 13 variables of 1709 observations. These variables are:

- lotSize - lot size
- Age - age of the house in years
- landValue - land value
- livingArea - living area
- pctCollege - percentage of local residents with college education
- Bedrooms - number of bedrooms
- Fireplaces - number of fireplaces
- Bathrooms - number of bathrooms
- Rooms - number of rooms
- Fuel - fuel used for heating system
- Waterfront - waterfront property
- newConstruction - whether a new construction or not
- PriceCat - price category of the house (expensive / budget)

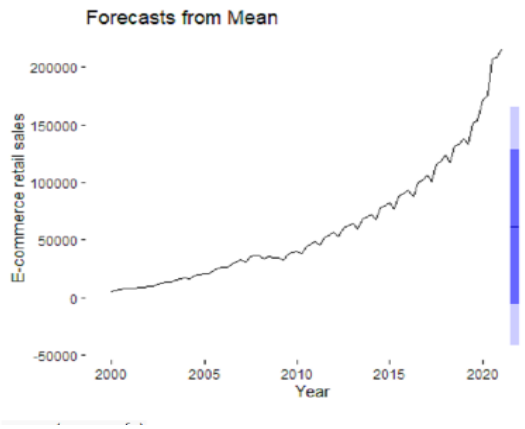
DATA PREPARATION

After summarising the data using summarise() function we found the data types. We decided to change some of the data using as.factor() function.
We also checked for NA and Inf values using is.na() and is.infinite() functions. We found any of these values in our dataset.
Then we continued by splitting the data into train and test set in 70:30 ratio. The train set contains 1181 values and the test set contains 528 values.

RESULTS AND EVALUATION

TIME SERIES DATA

THE MEAN FORECAST METHOD



Firstly, we started with the Mean Forecast method. We got our model by using the meanf() function and we set h to 3 so we got a prediction for 3 periods ahead. Using the autoplot() function we got this plot.

By using the summary() function we came to these model evaluation results.

RMSE	MAPE
51368.05	160.32

Figure 10 Mean Forecast Method

We can see that our RMSE value is equal to 51000, which means how concentrated the data is around the line of best fit. The smaller the value, the better. Then we have a MAPE value equal to 160. The MAPE measures this accuracy as a percentage, which means it's equal to 160%. MAPE >100% means that the errors are "much greater" then the actual values. Therefore, our model didn't perform very well.

THE NAIVE METHOD

The second method we used was the Naive method. We created the model by using the naive() function and we set the h to 3. Using the autoplot() function we got this final plot.

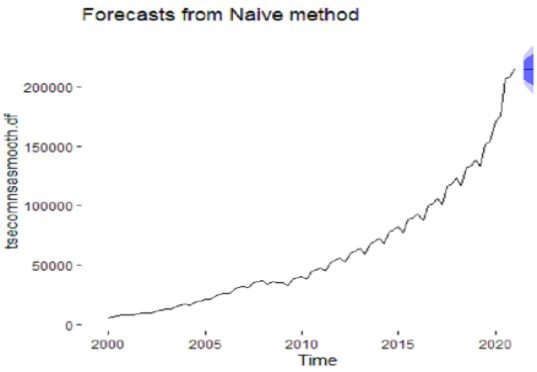


Figure 11Naive Method

By using the summary() function we obtained the following model evaluation results.

Here we can see that RMSE is equal to 6000, which is much smaller than

RMSE	MAPE
6097.34	6.11

in the previous model. MAPE is equal to only 6%, which is great. This model performed much better than the first one.

THE SEASONAL NAIVE METHOD

After that, we used the Seasonal Naive method. The model was created using the snaive() function with the h set to 3. We got the following graph using the autoplot() function.

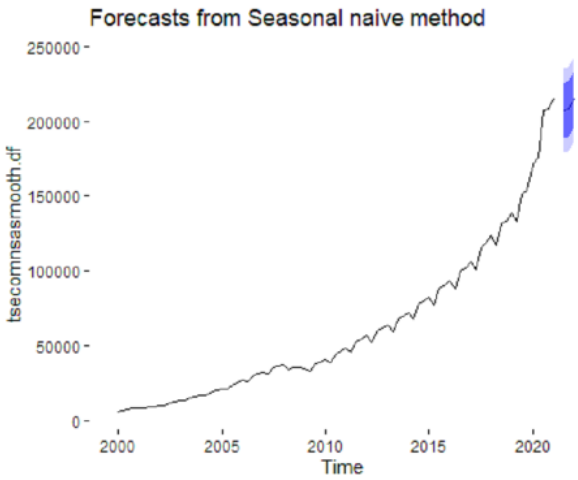


Figure 12 Seasonal Naive Method

By using the summary() function we obtained the following model evaluation results.

RMSE	MAPE
14343.18	15.22

The RMSE of this model is 14000, which is higher than the last model. And the MAPE value is equal to 15%, which is also higher than the last model. However, this model also performed much better than the first model.

ARIMA MODEL

After examining the ACF and PACF graph, we came to a conclusion to use ARIMA(1 1 0) (1 1 0) (by using the auto.arima() function, we came to the same conclusion). The model was created by using the function arima() and by using the function forecast() we observed this plot.

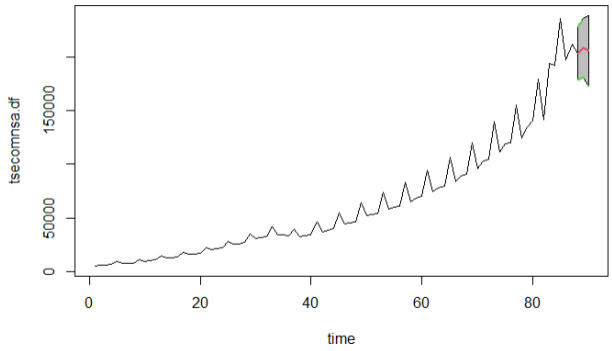


Figure 13 ARIMA

By using the summary() function we obtained the following model evaluation results.

RMSE	MAPE
12639.1	11.068

The RMSE of this model is 12600, which is smaller than the last 2 models, but still higher than the RMSE for the Naïve Method. The MAPE is equal to 11%, which is not bad value, but the Naïve model still performed better.
After performing a Box-Ljung test, we found a big p-value equal to 0.49, which indicates that the residuals are independent. We also performed a qqnorm() function to check a normality assumption, and everything seemed ok as the points follow a straight line with some outliers around.

LOGISTIC REGRESSION DATA

We started our logistic regression analysis by creating the first model which included all the variables in the dataset using the glm() function.
By using the summary() function we obtained the results of our first model.

```
Call:
glm(formula = PriceCat ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9764  -0.5647  -0.2153   0.5048   2.9600

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       -7.010e+00  6.579e-01 -10.654  < 2e-16 ***
lotSize            3.498e-01  1.711e-01   2.045   0.0409 *
age               -5.618e-03  4.247e-03  -1.323   0.1859
landValue          4.839e-05  5.131e-06   9.431  < 2e-16 ***
livingArea         2.401e-03  3.248e-04   7.392 1.44e-13 ***
pctCollege        -1.272e-02  9.802e-03  -1.298   0.1944
bedrooms          -8.833e-02  1.558e-01  -0.567   0.5707
fireplaces        -4.749e-02  1.846e-01  -0.257   0.7970
bathrooms          1.035e+00  1.984e-01   5.216 1.83e-07 ***
rooms              2.887e-02  5.868e-02   0.492   0.6227
fuelgas           -2.697e-02  2.448e-01  -0.110   0.9123
fueloil           -8.591e-02  3.594e-01  -0.239   0.8111
waterfrontYes      2.493e+00  1.074e+00   2.321   0.0203 *
newConstructionYes -4.455e-01  5.018e-01  -0.888   0.3747
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1627.14  on 1180  degrees of freedom
Residual deviance:  882.94  on 1167  degrees of freedom
AIC: 910.94

Number of Fisher Scoring iterations: 6
```

Now we can look at the fitting and figure out what the model is trying to tell us.
Firstly, we can see that bedrooms, fireplaces, rooms, fuelgas, fueloil are not statistically significant. As for the statistically significant variables, landValue and livingArea has the lowest p-value, which is suggesting a strong association of these variables with the probability of being classified as Expensive or Budget. The positive coefficient for these predictors is telling us that all the variables being equal, the higher the livingArea or landValue is more likely to be classified as Expensive.

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: PriceCat

Terms added sequentially (first to last)

                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                                1180    1627.14
lotSize            1    30.89      1179    1596.24 2.724e-08 ***
age                1    18.20      1178    1578.05 1.991e-05 ***
landvalue          1   325.60      1177    1252.45 < 2.2e-16 ***
livingArea         1   330.08      1176     922.36 < 2.2e-16 ***
pctCollege         1     2.49      1175     919.87  0.114499
bedrooms           1     0.06      1174     919.81  0.806680
fireplaces         1     0.15      1173     919.66  0.696179
bathrooms          1    29.03      1172     890.63 7.127e-08 ***
rooms              1     0.22      1171     890.41  0.639036
fuel               2     0.05      1169     890.36  0.973567
waterfront         1     6.66      1168     883.70  0.009887 **
newConstruction    1     0.77      1167     882.94  0.380979
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To assess the table of deviation, we can now use the anova() function on the model. Here we can see the difference between the null and residual deviances, which illustrates how well our model performs in comparison to the null model (the one which only has the intercept). This range should be as broad as possible. We can see the drop in the deviance by adding each of the variables. It's visible that adding landValue, livingArea, pctCollege, and bathrooms reduces residual deviation considerably. The remaining factors have a smaller impact on the model. A large p-value indicates that the model without the variable explains roughly the same amount of variation as the model with the variable. We want to see a big reduction in deviation. [6]
The next step in evaluation of the model is a creation of the confusion matrix using the ConfusionMatrix() function.

	0	1
0	259	70
1	28	171

Here we can see that our first model correctly predicted 171 PriceCat being classified as Expensive and 259 being classified as Budget. It wrongle predicted 70 being Expensive when they were actually Budget and 28 wrongle predicted as Budget when they were actually Expensive.

We will also calculate theTotal Misclassificatio Rate: The percentage of total incorrect classifications made by the model by using the misClassError() function. The result value is 17.8%, which means that 82.2% of data was classified correctly.

After that we decided to create a new model. In this model we used all the variables which had the smallest p-values. These variables are: landValue, age, lotSize and bathrooms.

```
Call:
glm(formula = PriceCat ~ landvalue + livingArea + age + lotSize +
    bathrooms, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.9077  -0.5650  -0.2173   0.5269   2.9641

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.613e+00  4.667e-01 -16.310  < 2e-16 ***
landvalue    4.578e-05  4.542e-06  10.081  < 2e-16 ***
livingArea   2.291e-03  2.414e-04   9.491  < 2e-16 ***
age         -4.088e-03  3.798e-03  -1.076   0.2817
lotSize      3.466e-01  1.602e-01   2.163   0.0305 *
bathrooms    1.022e+00  1.940e-01   5.271 1.36e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1627.1  on 1180  degrees of freedom
Residual deviance:  894.1  on 1175  degrees of freedom
AIC: 906.1

Number of Fisher Scoring iterations: 6
```

Here we can see the results from the summary() function and the anova() function.
From these results we can notice that the AIC value is a little smaller than in the first model.

```
Analysis of Deviance Table

Model: binomial, link: logit

Response: PriceCat

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                        1180    1627.14
landvalue   1   315.57       1179    1311.57 < 2.2e-16 ***
livingArea  1   372.50       1178     939.07 < 2.2e-16 ***
age         1    12.53       1177     926.55 0.0004014 ***
lotSize     1     4.18       1176     922.36 0.0408916 *
bathrooms   1    28.26       1175     894.10  1.06e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The conclusion matrix was created next.

	0	1
0	259	68
1	28	173

Here we can see that our second model correctly predicted 173 PriceCat being classified as Expensive and 259 being classified as Budget. It wrongle predicted 68 being Expensive when they were actually Budget and 28 wrongle predicted as Budget when they were actually Expensive.
The result of the total misclassification rate was 17.42%, which means that 82.58% of the data was classified correctly. Which is 0.38% better than the original model.

To finalise our logistic regression analysis, we also created a graph using the ggplot() function.
On this graph we can see that most of the houses with Expensive price category (those in green) are predicted to have a high probability of having Expensive price category. And most of the houses in Budget price category (those in orange) are predicted to have a low probability of having Expensive price category.
That means that our logistic regression done a pretty good job.

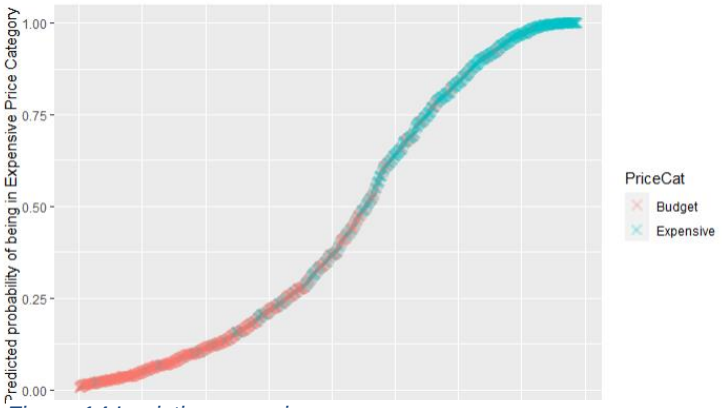
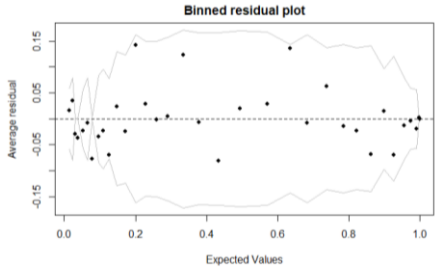


Figure 14 Logistic regression

Logistic Regression Assumptions

1. **Independence of residuals assumption**



We got this plot using the binnedplot() function.
The grey lines represent +- 2 SE bands, each of which should contain around 95% of the observations. The majority of the fitted values tend to fall inside the SE bands thus this model appears to be plausible. Therefore, the residual assumption is justified.

Figure 15 Independence of residuals

2. **Multicollinearity assumption**

```
landValue livingArea        age    lotSize  bathrooms
 1.152867   1.377419   1.356191   1.013701   1.660510
```

Here we checked for multicollinearity by using the VIF() function. A VIF score greater than 5 or 10 implies a significant amount of collinearity. There is no collinearity in our example: all variables have VIF values considerably below 5. Therefor the multicollinearity assumption is justified.

3. Influential values assumption

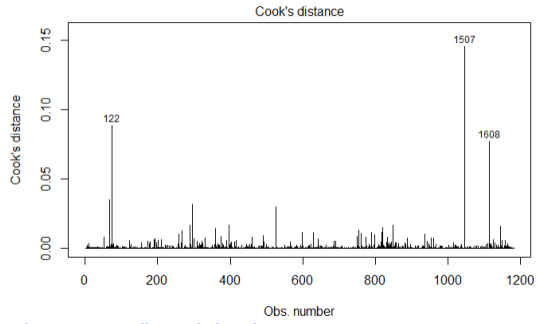


Figure 16 Influential values

Extreme individual data points are variables that can alter the quality of a logistic regression model.
To analyze the data's most extreme values, use Cook's distance values. This graph displays the top three largest values.
Not all outliers are noteworthy occurrences. To evaluate if the data contains any potentially influential observations, look at the standardized residual error.
Outliers are data points with absolute standardized residuals greater than 3 and should be looked into further.

After plotting the standardised residuals, we can see that there are no residuals greater than 3, therefor this assumption is justified.

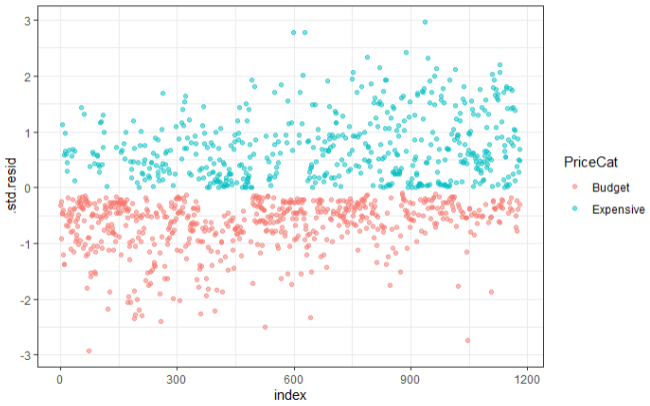


Figure 17 Standardised residuals

4. Linearity

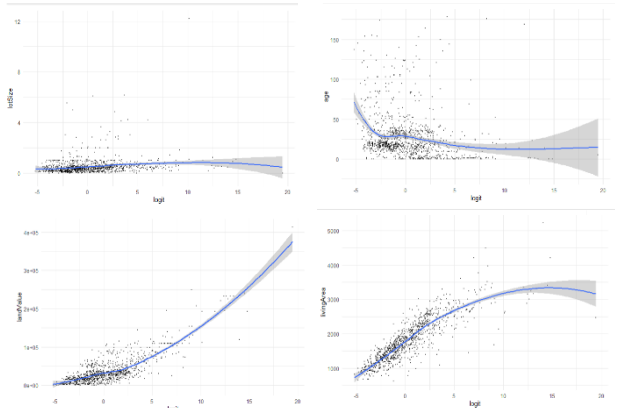


Figure 18 Linearity

The last assumption for the logistic regression is that the continuous variables have linearity against the log odds of the dependent variable. As we can see on the graph all of our continuous variables have a fairly linear relationship, therefor the linearity assumption is justified.

CONCLUSION AND FUTURE WORK

TIME SERIES DATA

After examining RMSE and MAPE values for all the models, we can come to the conclusion that the model with the best performance was the Naïve Method. How to improve our time series forecasting model? We could use some more complex time series forecasting models such as SARIMA, XGBoost or LSTM algorithms.

LOGISTIC REGRESSION DATA

After performing the Confusion Matrix and total missclasification functions on our model we found out that the second model performed a little bit better than the original one. But the difference is just very small, it is less than 1%. How could we improve the logistic regression model? We could do it by normalising, class imbalance improvement, hyperparameter tuning, exploring more classifiers or error analysis.

BIBLIOGRAPHY

[1] Tableau, Time Series Analysis: Definition, Types, Techniques, and When It's Used, available online: https://www.tableau.com/learn/articles/time-series-analysis
[2] Toppr, Components of Time Series, available online: https://www.toppr.com/guides/business-mathematics-and-statistics/time-series-analysis/components-of-time-series/
[3] Java T Point, Logistic Regression in Machine Learning, available online: https://www.javatpoint.com/logistic-regression-in-machine-learning
[4] National Library of Medicine, Logistic regression: a brief primer, available online: https://pubmed.ncbi.nlm.nih.gov/21996075/
[5] R-bloggers, Is my time series additive or multiplicative, available online: https://www.r-bloggers.com/2017/02/is-my-time-series-additive-or-multiplicative/
[6] R-bloggers, How to perform a logistic regression in R, available online: https://www.r-bloggers.com/2015/09/how-to-perform-a-logistic-regression-in-r/