Comparison of performance of different Machine Learning Models in different contexts

Samya Bose
MSc in Data Analytics (School of
Computing)
National College of Ireland
Dublin, Ireland
x18180523@student.ncirl.ie

Abstract—**This project report contains the study and evaluation of different Machine Learning methods when applied on different datasets. For the purpose of variation, the datasets selected have completely different structure, which includes categorical, numerical, and time stamped data. This report also talks about the pros and cons of each ML method when applied to a specific data scenario. The datasets contain car crash data, Euro survey data and Border Crossing data. ML methods suitable for the datasets have been applied. Some methods have not acted sonicely, and some have given good results.**

Keywords—K-Mode, Clustering, Naïve Bayes, Logistic Regression, ARIMA, Auto Correlation

I. Introduction

The purpose of the project is to critically evaluate different Machine Learning methods, and find out their performance and application limitations. Since all types of ML methods cannot be applied on all types of data, I have taken 3 datasets of completely different structures, which will be helpful in determining the objective. As per description, I have taken 5 ML methods, to be applied on the 3 datasets. These are: K-Modes Clustering, Decision Tree, Logistic Regression, ARIMA, and Naïve Bayes. The three datasets are: Car Crash Data, Border Crossing Data, Euro Survey Data. Car Crash Data has mostly categorical variables, Border Entry Data has some categorical, some continuous and date wise data, and Euro Survey Data has mostly continuous and numeric categorical data.

II. Related Work

A lot of studies have been done in the field of Machine Learning. Here, I have viewed some previous papers on the ML methods that I am going to use in my project. Since, I have taken my datasets from Kaggle and data.world, and hence there is a deficit of research papers on the data sets.

Starting with K-Means Clustering, [14] Oyelade, O. J., Oladipupo, O.O, Obagbuwa, I. C. in their paper, they have provided a simple and qualitative methodology to compare the predictive power of clustering algorithm and the Euclidean distance as a measure of similarity distance. They have improved on the limitations of the existing methods. The journal [15] by Chunhui Yuan and Haitao Yang gives some good explanations K-means Algorithm, and focuses on the evaluation methods that I have decided to use in my model.

Coming to the Decision Tree Classification, [16] Bhaskar N. Patel, Satish G. Prajapati and Dr. Kamaljit I. Lakhtaria, has given good insight on the Decision Tree model. They have also given some ways to speed up the process which will be helpful for me. They have explained *"The decision tree algorithm is a top-down induction algorithm. The aim of this algorithm is to build a tree that has leaves that are homogeneous as possible. The major step of this algorithm is to continue to divide leaves that are not homogeneous into leaves that are as homogeneous as possible."*

Coming to the third process, Naïve Bayes, [17] I.Rishs' paper on empirical study of the Naïve Bayes classifier, provides a good detail on the whole Naïve Bayes procedure. It explains all the concepts and theorems. This paper will help me understand the concepts fully and apply it on the dataset more easily.

I had initially started with K-Means clustering for the Car Crash dataset, but since there are a lot of categorical variables, I have chosen K-Mode clustering, instead of K-Means. This method of clustering is a modification of K-Means, as it uses mostly the frequency of occurrence of categories instead of the Euclidian distance between the cluster mid points to create the clusters.

There has been some EDA done on the Border Crossing Dataset in Kaggle, however, there is no ML method that has been applied in the Kaggle Kernels. So, I have taken this dataset to perform ARIMA, as it has a seasonality part to it. Same with the Euro Survey Dataset, there has been o ML methods that has been applied in Kaggle, so, I have used Naïve Bayes on this dataset.

III. Methodology

A. Approach

I have used KDD[18] approach to in this project. KDD has seven main steps. These are:

i. Data Cleaning
ii. Data Integration
iii. Data Selection
iv. Data Transformation
v. Data Mining
vi. Pattern Evaluation
vii. Knowledge Representation

The data cleaning part has been done by doing EDA of missing values, and then either imputing them with the randomised most occurring values in case of categorical columns, and imputing by mean for the continuous columns. However, the columns, which had almost 70% NA or missing values have been omitted altogether, as it did not contribute to any analysis. I have done sampling of the datasets, as each dataset had row counts of more than 100000. This sampling has been done after shuffling the dataset, so that there is no bias in the analysis.

Since I have taken the datasets from a single source, there was no need for the data integration part.

All the datasets had a large number of columns, especially Euro Survey dataset, which had more than 300 columns. So, to satisfy the Data Selection part of KDD, I have taken the columns which were of interest for my intended analysis.

The raw data that I had taken from Kaggle and Data.World, was not suitable for using in analysis. There was a lot of transformation, that had to be done. This included binning of categorical and continuous variables, changing the date stamps to proper date format, one hot encoding for categorical variables, which created dummy variable for each category under a column.

Five Data Mining and Machine Learning methods have been applied on the three datasets. The results, patterns and classifications have been explained in the later stages.

B. *Dataset Information and Preparation*

1. **Car Crash Data:** [1] This dataset comprises of data mostly from Montgomery County Police. The data set has been taken from data.world. This dataset primarily concentrates on data regarding Car Crash information with detailed data of Route Type, surface conditions, car impact locations, date and time of crash, etc. The dataset initially had 42 columns and more than 100000. However, during the initial cleaning of the data, the row count has decreased to 35000.This is due to the presence of huge amount of NA values in most of the columns. I have also reduced the column count to 13 for the ease of data cleaning and also because rest of the columns will not be playing any role in the using of the ML model.

   The final dataset is actually a merge of three datasets that were connected by Report Number and Local Case Number. The merging of the dataset created duplicates of each of the entries, which was not desirable, hence, I had to remove all the duplicate values. In this dataset, the NA values have been imputed with the top most occurring values. There were a lot of wrong year entries as well, which needed human intervention. So I have taken the wrong year values, and have tried to correct it to a meaningful value.

2. **Border Crossing Data:** [2] This dataset consists of data of vehicles like buses, trailer trucks, trains entering the US - Canada and US – Mexico land border ports. The data has been recorder since 1996 till 2019, so, it gives a historical approach to the analysis. The dataset can be used to predict the type and number of vehicles which may cross the border in near future. Also, it can be used to predict the ports through which the vehicles may cross through. This information might be useful for the police as they can get an insight on the vehicle movement pattern.

   The main issue with this dataset was to convert the date and time stamp to a proper date format, and get some initial idea on the seasonality of the data. For this, I have referred to the EDA done by other people in Kaggle for this dataset. This gave an idea on which columns to be used for applying the ML method. I have kept almost all the columns of the original dataset. I have removed some of the columns having unnecessary data like location based on coordinates, since I will not be plotting a map for clustering.

3. **Euro Survey Data:** [3] This is one of the largest dataset, with variables well over 300 and more than 150000 records. This dataset consists of data gained over multiple surveys in European countries. Each survey has multiple questions which has been made into its own column. There are few administrative columns of the dataset which has dates, ID No, country data, etc. Out of all these columns, I have taken 7 columns which tend to give the output to my applied ML method. For this dataset, I had to do binning for categorical variable so as to fit it as the outcome variable. Also, there were quite some NA and missing values, which have been imputed with mean or most occurring values depending on continuous and categorical variable.

C. *Machine Learning Methods*

1. **K-Mode Clustering:** [4] "The *k*-modes algorithm an extension of the k-means algorithm. The data given by data is clustered by the *k*-modes method (Huang, 1997) which aims to partition the objects into *k* groups such that the distance from objects to the assigned cluster modes is minimized.

By default, simple-matching distance is used to determine the dissimilarity of two objects. It is computed by counting the number of mismatches in all variables. Alternative this distance is weighted by the frequencies of the categories in data (see Huang, 1997, for details).

If an initial matrix of modes is supplied, it is possible that no object will be closest to one or more modes. In this case less cluster than supplied modes will be returned and a warning is given."

2. **Decision Tree:** According to Rajesh S. Brid [5], *"A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements."*

Prashant Gupta [6] gives a simple explanation of the Decision tree and compares it as an upside down tree, where the root node is at the top, and then the tree branches out with each node being a decision question having a dichotomous output. It might be possible that both the output leads to more branching, or one of the output leads to a terminal node. This ML method can be used on the Car Crash dataset to get an idea on which type of vehicles are more prone to crashes and which conditions can lead to more crashes.

3. **Logistic Regression:** Logistic Regression is a type of supervised Machine Learning model, which can give only dichotomous output. It is almost like linear regression but it uses sigmoid function to bring in the predicted value between 0 and 1. This is because the predicted value of the logistic regression can vary anywhere between positive infinity to negative infinity. We can apply this method in Car Crash Data to predict if a specific type of car will meet with an accident or not.

4. **ARIMA:** ARIMA stands for Auto Regressive Integrated Moving Average. Going by the explanation in Machine

Identify applicable funding agency here. If none, delete this text box.

Learning Plus [7], *"ARIMA is actually a class of models that 'explains' a given time series based on its own past values, that is, its own lags and the lagged forecast errors, so that equation can be used to forecast future values"*. One good thing about ARIMA is that it can be applied to any non-seasonal time series data that exhibits pattern and not random noise. If seasonality is added to the equation, then it becomes Seasonal ARIMA or SARIMA in short. Since ARIMA is a linear regression model, we cannot have correlations in predictors, and the time series should be stationary. So, to get rid of this correlations, differencing might be required. This method is suitable for the Border Crossing Entry Data set because it is time series dataset, and we can forecast the border crossing characteristics as well. If we look closely, we also might find a seasonality factor in the data, which can be used for SARIMA.

5. **Naive Bayes:** Machine Learning Plus [8] says that *"Naive Bayes is a probabilistic machine learning algorithm that can be used in a wide variety of classification tasks. Typical applications include filtering spam, classifying documents, sentiment prediction etc."*. It is a set of supervised learning methods which is based on Baye's theorem. It is called naïve because of the naïve assumption of conditional independence of the input features that are being used in the model. This model is good for the Euro Survey Dataset, as it contains a lot of ranked data based on surveys. We can do some predictive analysis on the response of people to the next surveys.

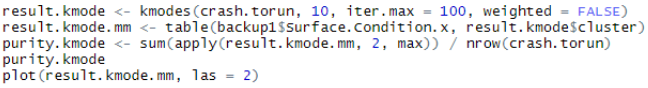
6. **Random Forest:** A random forest is an ML method, that is based on Decision Tree. Where a decision tree is a single tree that gives binary output, a random forest creates a number of decision trees. This is mostly useful for variables that have a large number of categories, and a number of variables to consider. One good aspect of Random Forest is that it can give estimates of the importance of variables

IV. Evaluation

Following are the evaluations for all the methods that have been applied.

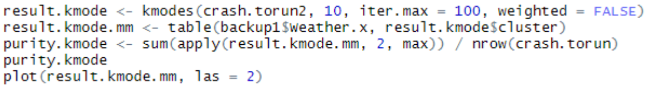
A. K-Modes Classification

I have performed the K-Modes classification for 2 variables. One is Surface Condition of the road, and the second is for the Weather Conditions. Following is code snippet for the Surface Condition clustering:



```
result.kmode <- kmodes(crash.torun, 10, iter.max = 100, weighted = FALSE)
result.kmode.mm <- table(backup1$Surface.Condition.x, result.kmode$cluster)
purity.kmode <- sum(apply(result.kmode.mm, 2, max)) / nrow(crash.torun)
purity.kmode
plot(result.kmode.mm, las = 2)
```

Fig 1: Clustering on Surface Conditions



```
result.kmode <- kmodes(crash.torun2, 10, iter.max = 100, weighted = FALSE)
result.kmode.mm <- table(backup1$Weather.x, result.kmode$cluster)
purity.kmode <- sum(apply(result.kmode.mm, 2, max)) / nrow(crash.torun)
purity.kmode
plot(result.kmode.mm, las = 2)
```

Fig 2: Clustering on Weather Conditions

From the above codes, we get the cluster plot which are as follows:

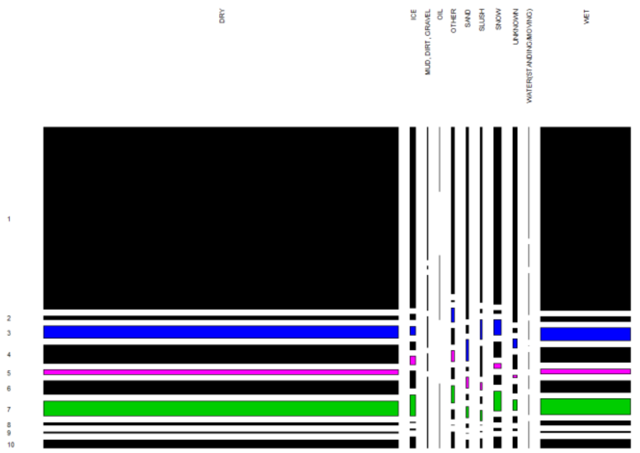


Fig 3: Clusters for the Surface Conditions

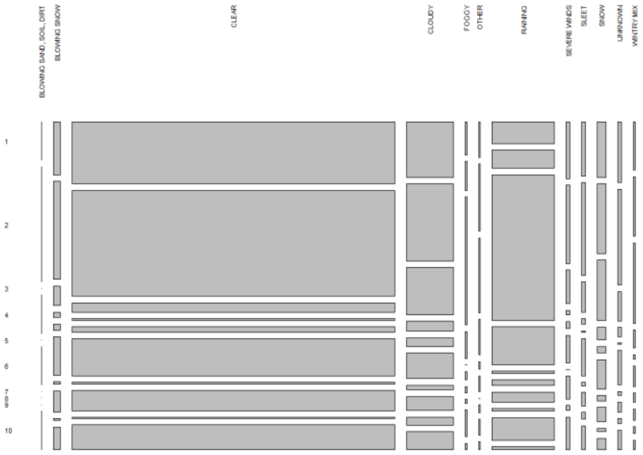


Fig 4: Clusters for the Weather Conditions

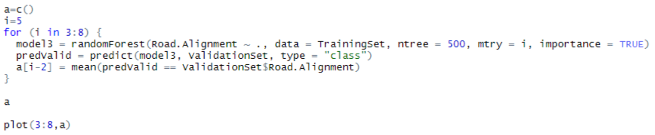
From the first cluster, we can see that the most number of crashes are taking place on Dry Road conditions followed by wet road conditions. The other categories are insignificant compared to these two categories. From the second cluster, we can see that the most number of crashes are happening in clear weather followed by rain and cloudy.

We can infer from the clusters that most of the crashes are happening in clear weather and dry surface condition, which can be accounted for the high traffic density in these conditions. The second biggest clusters are for the rainy and cloudy weather and wet road conditions, which is quite normal because of the low visibility and aquaplaning that occurs in these situations.

B. Random Forest

I have performed random forest algorithm on 7 columns namely: Route Type, Weather, Surface Condition, Traffic Control, Road Alignment, Accident Fatality, and Time of Day. I have Random forest instead of Decision Tree because of this number of columns and the number of categories in each of the columns. The outcome variable here is the Road Alignmrnt. I have tried to predict how much the Road Alignment affects the crashes because of the other factors.

First I have tried with the default number of tries for the random forest, and then I have checked the change in accuracy of the model by changing the number of tries. Also, I have checked the importance of the variables affecting the accuracy of the model. Following is the code snippet for the model:



```
a=c()
i=5
for (i in 3:8) {
  model3 = randomForest(Road.Alignment ~ ., data = TrainingSet, ntree = 500, mtry = i, importance = TRUE)
  predValid = predict(model3, ValidationSet, type = "class")
  a[i-2] = mean(predValid == ValidationSet$Road.Alignment)
}

a

plot(3:8,a)
```

Fig 5: Code for variable importance check

Following are the graphs generated for the Random Forest:

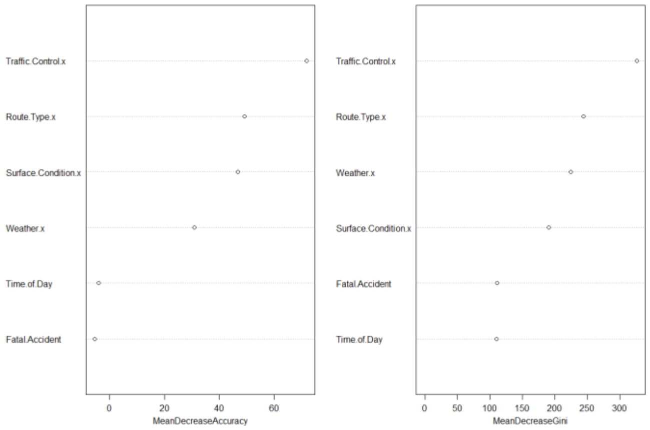


Fig 6: Checking importance of the variables in play

We can see from the above graph, that the importance of Traffic Control is the highest for the Accuracy of the model. Also, for the importance of Gini, we can see that Traffic Control is the most important variable, followed by Route Type, Weather and Surface Condition.

Following is the graph of accuracy for different number of tries:

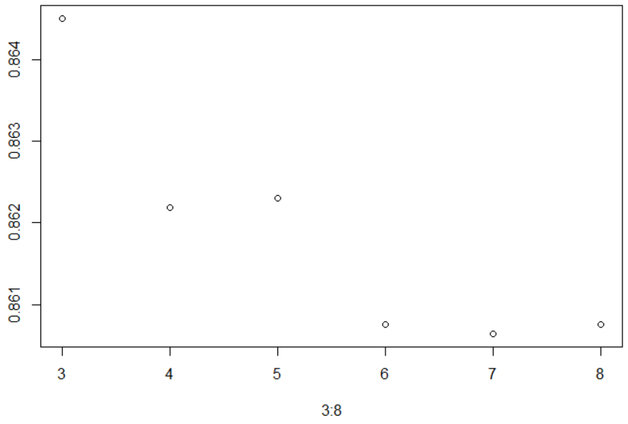


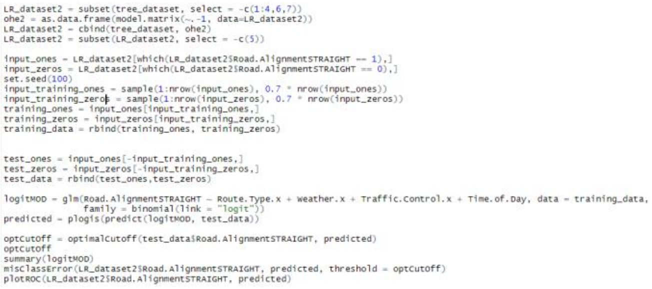
Fig 7: Accuracy change over number of mtry

From the above graph we can see that the accuracy of the model decreases with the number of tries. So, we take the default number of tries for the maximum accuracy of the model.

I have not included the random forest tree visualisation in this report as the structure is too large to be included. It is present in the code base.

C. Logistic Regression

I have applied Logistic regression on the Car Crash dataset as well. For this regression, I have taken Road Alignment as the outcome variable. Since, the outcome variable for Logistic Regression has to be in binary form, I have implemented One hot encoding for the variable. This essentially creates dummy variables for each of the category in the variable. I have taken the Straight raod alignment for the purpose of logistic regression, and I have kept all the other variablecategories intact. Following is the code for the logistic regression:



```
LR_dataset2 = subset(tree_dataset, select = -c(1:4,6,7))
ohe2 = as.data.frame(model.matrix(~.-1, data=LR_dataset2))
LR_dataset2 = cbind(tree_dataset, ohe2)
LR_dataset2 = subset(LR_dataset2, select = -c(5))

input_ones = LR_dataset2[which(LR_dataset2$Road.AlignmentSTRAIGHT == 1),]
input_zeros = LR_dataset2[which(LR_dataset2$Road.AlignmentSTRAIGHT == 0),]
set.seed(100)
input_training_ones = sample(1:nrow(input_ones), 0.7 * nrow(input_ones))
input_training_zeros = sample(1:nrow(input_zeros), 0.7 * nrow(input_zeros))
training_ones = input_ones[input_training_ones,]
training_zeros = input_zeros[input_training_zeros,]
training_data = rbind(training_ones, training_zeros)


test_ones = input_ones[-input_training_ones,]
test_zeros = input_zeros[-input_training_zeros,]
test_data = rbind(test_ones,test_zeros)

logitMOD = glm(Road.AlignmentSTRAIGHT ~ Route.Type.x + Weather.x + Traffic.Control.x + Time.of.Day, data = training_data,
               family = binomial(link = "logit"))
predicted = plogis(predict(logitMOD, test_data))

optCutOff = optimalCutoff(test_data$Road.AlignmentSTRAIGHT, predicted)
optCutOff
summary(logitMOD)
misClassError(LR_dataset2$Road.AlignmentSTRAIGHT, predicted, threshold = optCutOff)
plotROC(LR_dataset2$Road.AlignmentSTRAIGHT, predicted)
```

Fig 8: Code for Logistic Regrssion

The optimal cutoff received for the model based on the Straight Road Alignment of the test data and the predicted data is 0.6299 which is approximately 63%. The classification error received is 0.147. Following is the ROC curve for the Regression:

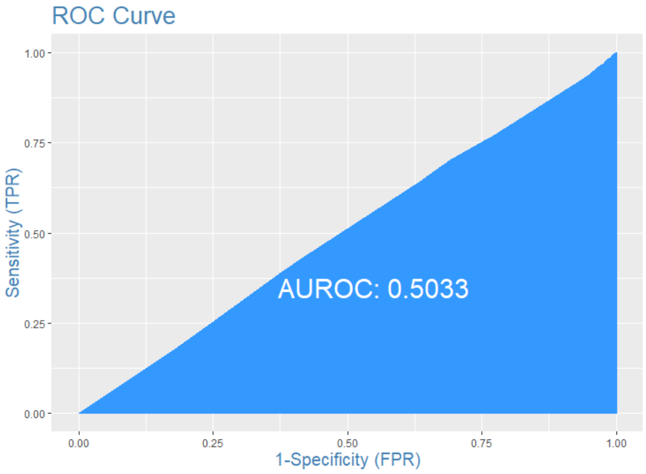


Fig 9: ROC Curve

We know that the prediction capacity of the model is better when the curve tends toward (1,0), and the intercept is the diagonal line with slope of 0.5. From this graph, we see that the model barely crosses the intercept line. We can say from this graph that the regression model has almost a 50-50 chance of predicting the True Positives and True Negatives. I have tried the same model with other variables as well and the result is almost the same.

D. ARIMA

I have applied ARIMA on the Border Crossing Dataset. The method mainly revolves around the number of vehicles crossing the border each year. There is data for 20 years, from 2000 to 2020. Following is the non-normalised graph for the number of vehicles crossing the border:

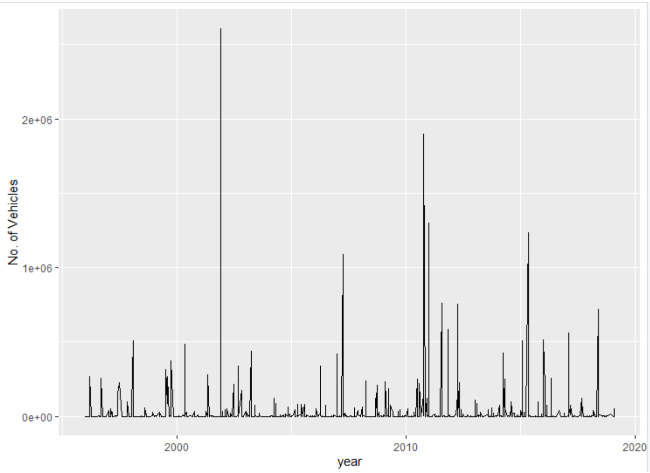


Fig 10: No. of vehicles crossing through the years

Next is the time series normalised graph for the same:

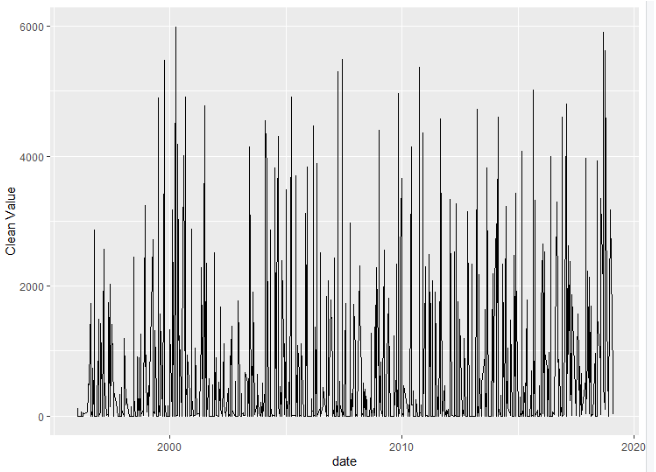


Fig 11: Time Series normalised graph

I have taken the daily and monthly trend for the passage of vehicles. Following is the trend and seasonality for the same.

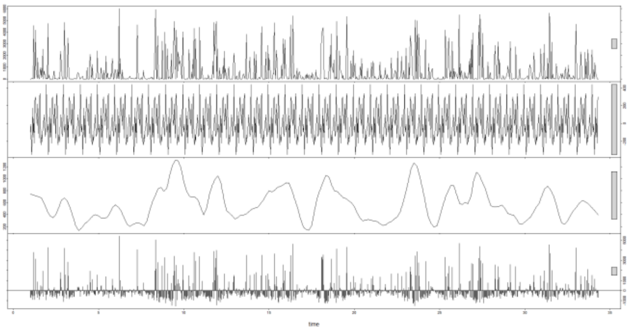


Fig 12: Showing the seasonality and trend of the vehicle passage

Due to the large variance in the data, one step of differencing was required to get the desired results from the ARIMA model. Following are the graphs of auto correlations and partial auto correlations:

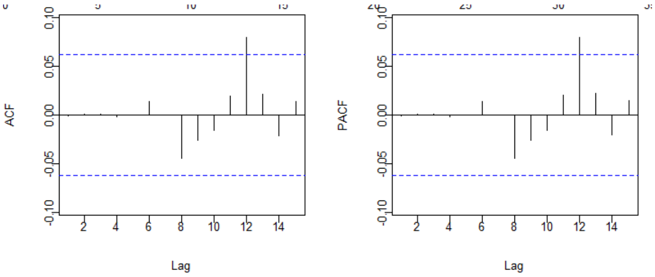


Fig 13: ACF and PACF graphs

The next graph shows the seasonal model residuals:

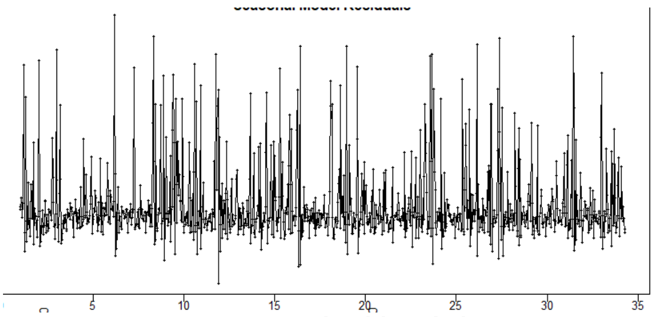


Fig 14: Seasonal model Residuals

Basically, the AR part of Arima means Auto Regression, which means that the current valus of the given time series can be obtained from its past values. The MA is Moving Average., which means that the current values of the given series is based on the past errors as its linear combination. From the ACF and PACF graphs, we can see the variations that are happening in the dataset for the AR model.

E. Naïve Bayes

I have used the Naïve Bayes ML method on the Euro Survey dataset. Due to the sheer large amount of variables present, I have taken only 8 variables which serve the purpose of the model. I have done binning on the Political Interest so that it has a binary outcome. I had to do some cleaning based on the missmap graph. Following is the graph before cleaning:

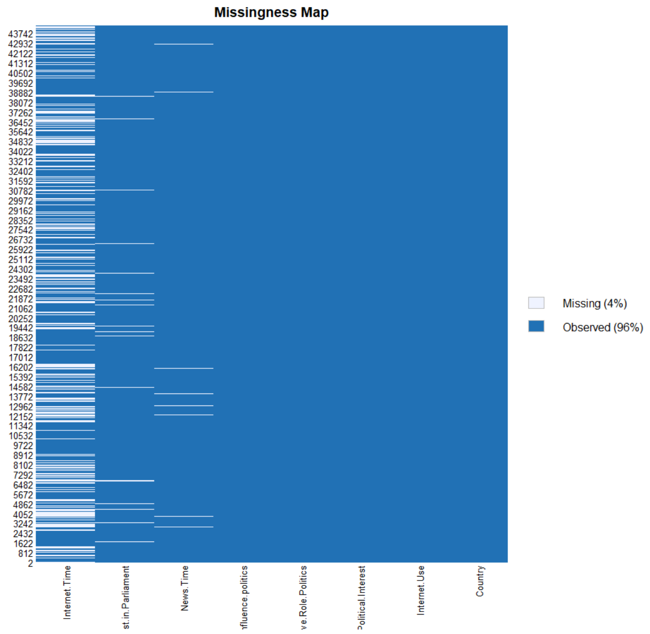


Fig 15: Missingness Map

Following is the graph after cleaning:

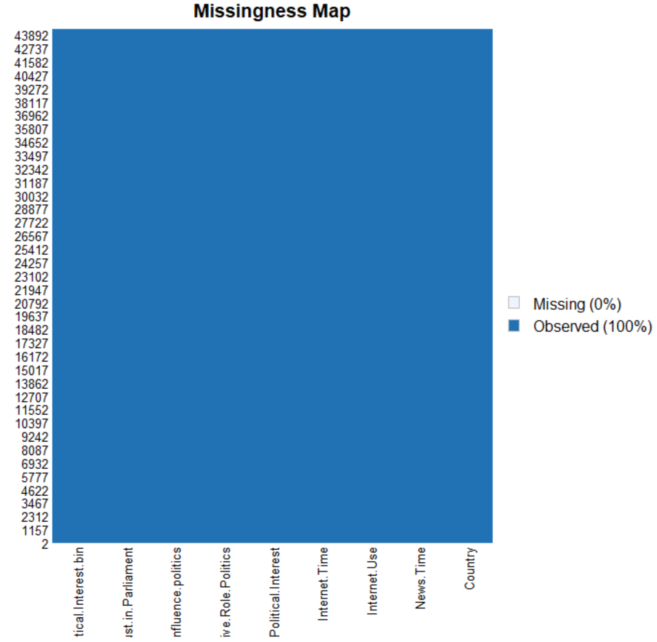


Fig 16: Missingness Map after cleaning

The next few graphs show the numbers of people spending different amounts of time on Internet and watching News, and their interest in politics.

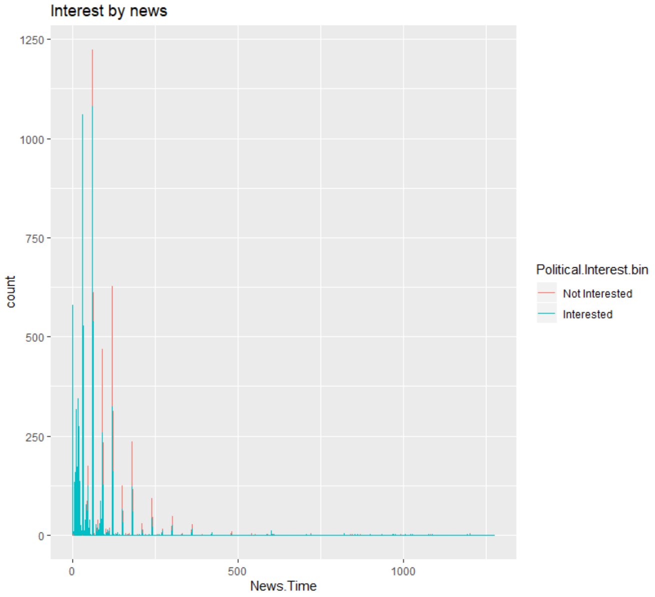


Fig 17: Interest in politics based on Time spent watching news

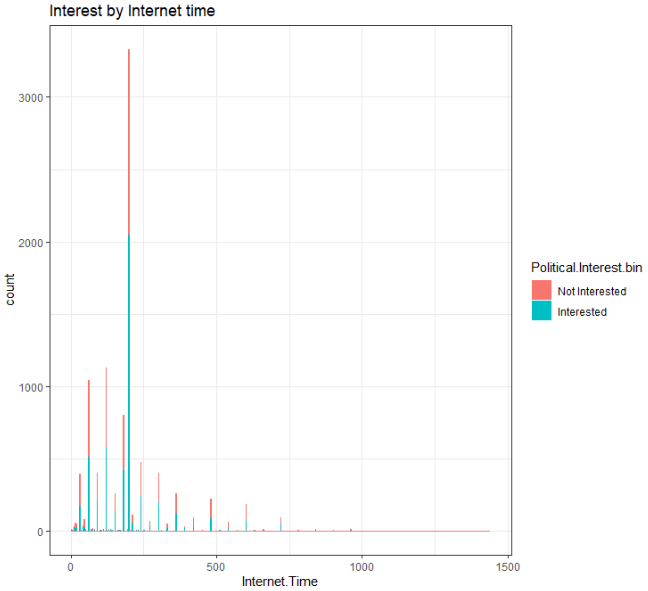
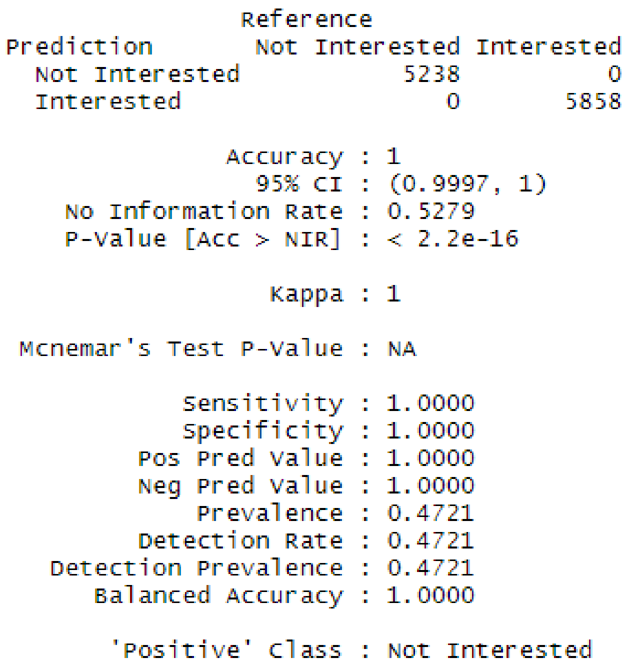


Fig 18: Political interest based on time spent on internet

In this process, I have taken Interest for Politics as a outcome variable and a 75:25 split for the training and the validation set.

The fitted model gives a kappa value of 0.99 which is same as the accuracy value. This means that the model is almost overfitting. But based on the distribution of values in the dataset, the positive and negative values are mostly 50-50 distributed through the dataset.

Following is the Confusion Matrix based on the prediction created by the applied model:



```
                Reference
Prediction       Not Interested Interested
  Not Interested           5238          0
  Interested                  0       5858

               Accuracy : 1
                 95% CI : (0.9997, 1)
    No Information Rate : 0.5279
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
             Prevalence : 0.4721
         Detection Rate : 0.4721
   Detection Prevalence : 0.4721
      Balanced Accuracy : 1.0000

       'Positive' Class : Not Interested
```

V. Conclusion and Future work

From the above application of ML methods and the evaluations, we can conclude that some machine learning methods are better than others when applied to specific types of data. For example, K-Means clustering serve the purpose of clustering very well when it comes to Categorical data. Decision Tree is not a good option for data, where the number of variables are large and where each variable is having quite a large number of categories. Also, I have observed that equal distribution of values in the outcome variable is not very good for the ML methods, as it might create an overfitting model based on the training data, and it might not work well on unseen data.

There is a lot of scope for future work on the datasets. Due to time constraint, only few columns could be used for the analysis. A lot more columns can be used for each of these datasets to get better results for all the models. For example, interest in politics can be also based on influence in parliament, trust in parliament, and Active Role in politics.

References

[1] US Open Data Portal, data.gov. [Online]. Available: https://data.world/montgomery-county-of-maryland/0ca5b758-c60a-40c7-bfb5-fda26ceee4c8 [Accessed on: Oct. 22, 2019]

[2] U.S. Government Works [Online]. Available: https://www.kaggle.com/akhilv11/border-crossing-entry-data [Accessed: Nov. 4, 2019]

[3] ESS Round 8: European Social Survey Round 8 Data (2016). Data file edition 2.1. NSD - Norwegian Centre for Research Data, Norway – Data Archive and distributor of ESS data for ESS ERIC. doi:10.21338/NSD-ESS8-2016. [Online]. Available: https://www.kaggle.com/pascalbliem/european-social-survey-ess-8-ed21-201617 [Accessed: Nov. 2,2019]

[4] Dr. Michael J. Garbade, "Understanding K-means Clustering in Machine Learning", Towards Data Science, Sep. 12, 2018. [Online]. Available: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1, [Accessed on: Sep. 11, 2019]

[5] Rajesh S. Brid, GREYATOM, Oct. 26, 2018. [Online]. Available: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb, [Accessed on: Nov. 5, 2019]

[6] Prashant Gupta, "Decision Trees in Machine Learning", Towards Data Science, May. 17, 2017. [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052, [Accessed on: Nov. 5, 2019]

[7] "ARIMA Model – Complete Guide to Time Series Forecasting in Python", [Online]. Available: https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/ , [Accessed on: Nov. 6, 2019]

[8] "How Naive Bayes Algorithm Works?", Machine Learning Plus, [Online]. Available: https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/ , [Accessed on: Nov. 6, 2019]

[9] Imad Dabbura, "K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks", Towards Data Science, [Online]. Available: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a , [Accessed on: Nov.7, 2019]

[10] "Selecting the number of clusters with silhouette analysis on KMeans clustering", scikit learn, [Online]. Available: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html, [Accessed on: Nov. 6, 2019]

[11] "Explanation of the Decision Tree Model", webfocusinfocenter, [Online]. Available: https://webfocusinfocenter.informationbuilders.com/wfappent/TLs/TL_rstat/source/DecisionTree47.htm, [Accessed on: Nov. 6, 2019]

[12] atmathew, "Evaluating Logistic Regression Models", R-Bloggers, Aug. 17, 2015. [Online]. Available: https://www.r-bloggers.com/evaluating-logistic-regression-models/ , [Accessed on: Nov. 7, 2019]

[13] "Arima - Evaluation of the model", STATISTICA Help, [Online]. Available: https://documentation.statsoft.com/STATISTICAHelp.aspx?path=TimeSeries/TimeSeries/Overview/Arima/ARIMAEvaluationoftheModel, [Accessed on: Nov. 7, 2019]

[14] Oyelade, O. J., Oladipupo, O.O & Obagbuwa, I. C, "Application of k-Means Clustering algorithm for prediction of Students' Academic Performance",Ph.D Thesis, Department of Computer Science, Covenant University, Ota, Nigeria, 2010. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1002/1002.2425.pdf [Accessed On: Nov. 6, 2019]

[15] Chunhui Yuan and Haitao Yang, "Research on K-Value Selection Method of K-Means Clustering Algorithm", Multidisciplinary Scientific Journal(*MDPI*), pp. 227-230. June, 2019, Available: https://arxiv.org/pdf/1002.2425 [Accessed On: Nov. 10, 2019]

[16] Bhaskar N. Patel, Satish G. Prajapati and Dr. Kamaljit I. Lakhtaria, "Efficient classification of Data Using Decision Tree", 2012 [Online]. Available: https://pdfs.semanticscholar.org/1fe4/7722d5e65829c7e04b19648f39b22384d28c.pdf [Accessed on. Nov. 7, 2019]

[17] I.Rish, "An empirical study of the naive Bayes classifier", T.J. Watson Research Cente, [Online]. Available: https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf , [Accessed on: Nov. 7, 2019]

[18] "KDD Process in Data Mining", GeeksforGeeks, [Online]. Available: https://www.geeksforgeeks.org/kdd-process-in-data-mining/