Mert Kaya
C2100965

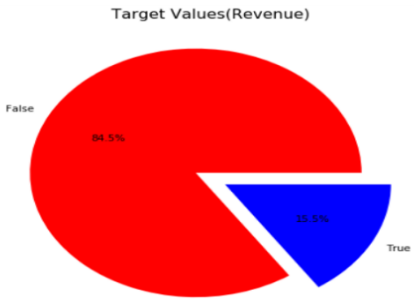
CMT – 307 Coursework 1

Introduction

First of all I want to summarize some key points of the project. Let's start with the dataset analysis. As we have seen from the data, the target variable is imbalanced. The imbalance ratio of the data is 0.1830 so in order to implement better classification models, we should fix the data otherwise we will have some trouble to predict the true labels correctly. In my project, to adjust the balance of the target variable, I used undersampling techniques, oversampling techniques and cost sensitive learning for the models of randomforest classifier, k-nearest neigbors classifier and logistic regression respectively. Then I compared the models via using roc-auc, precision-recall and probability calibration curves to decide which model have better performance. I looked at precision and recall values to compare the performance of the data and changed probability thresholds to find better results. Furthermore I also used some ensemble learning techniques to evaluate the performance as well. Since there is a word limit for the report, I wrote further explanations in the coding section.

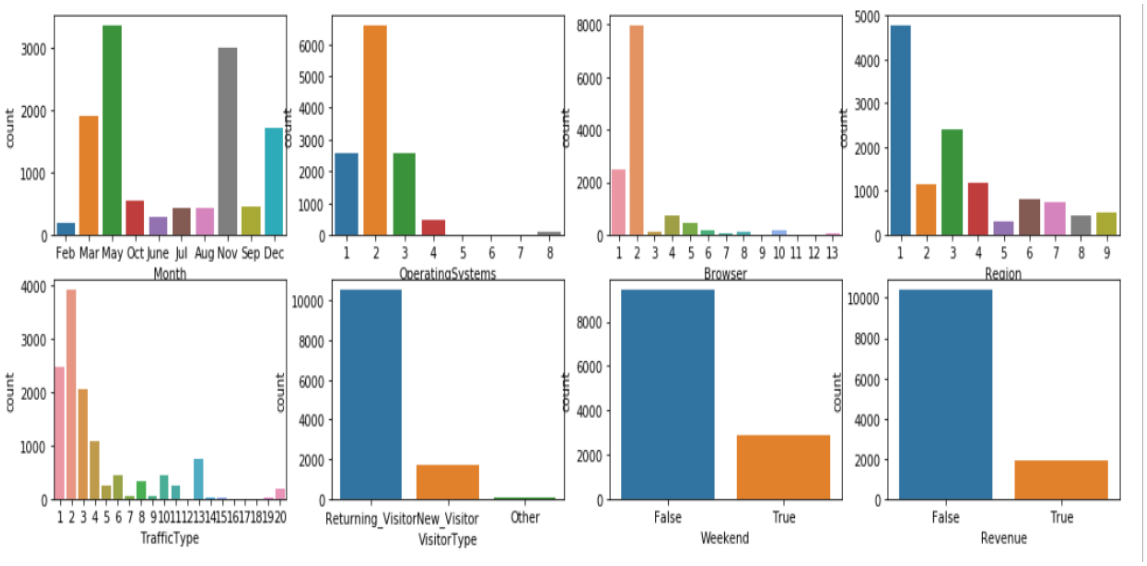
Data Exploration

I took advantage of pie charts, histograms, heat-map and bar charts to examine the characteristics of target variable, numerical variables and categorical variables. I calculated the data imbalance as 0.1830 in the target variable. The pie chart is effective to realize the imbalance in the target label to emphasize the imbalance on the target labels.

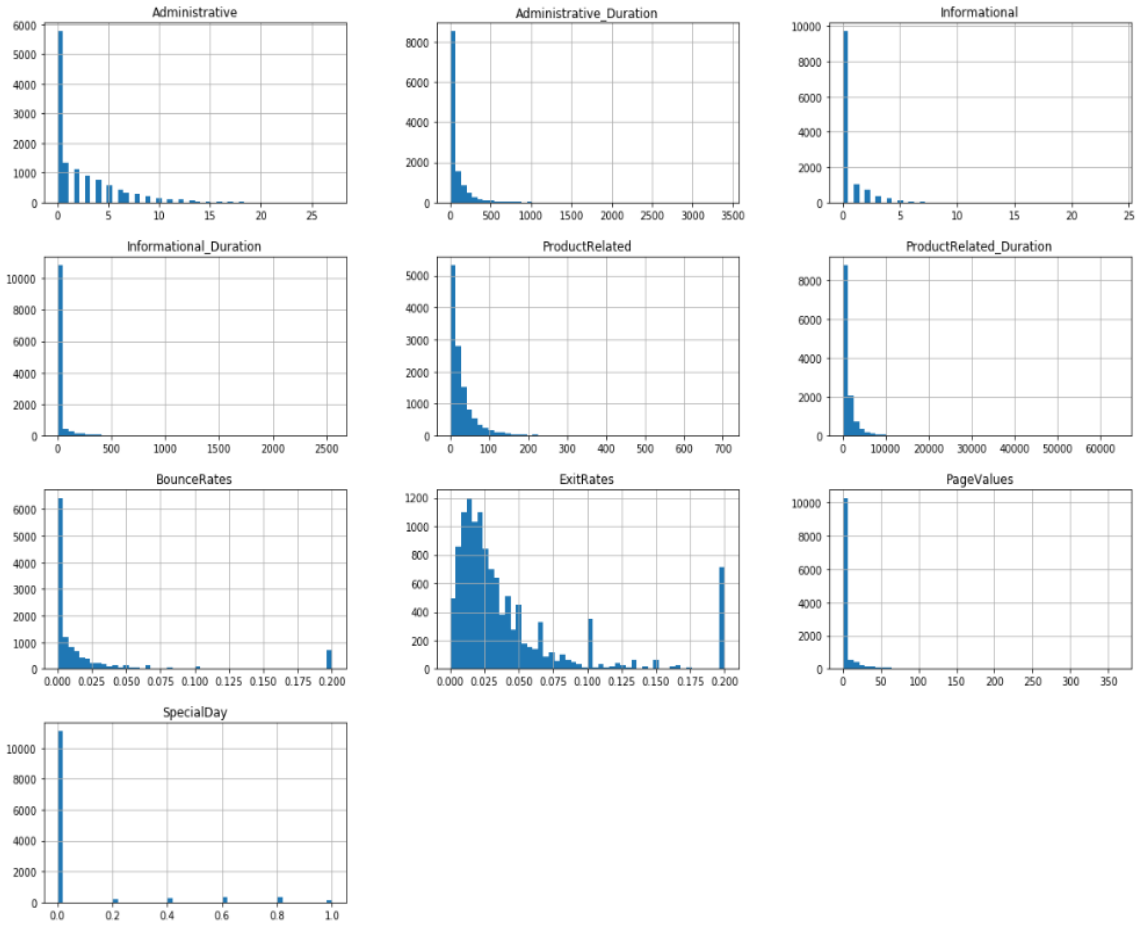


Pie Plot for Target Label

When it comes to analyze the numerical and categorical variables, observing the distribution of the numerical and categorical variables is very crucial to improve the distribution of the variables in the feature engineering part for acquiring better performance from the model.



Count Plot for Categorical Features



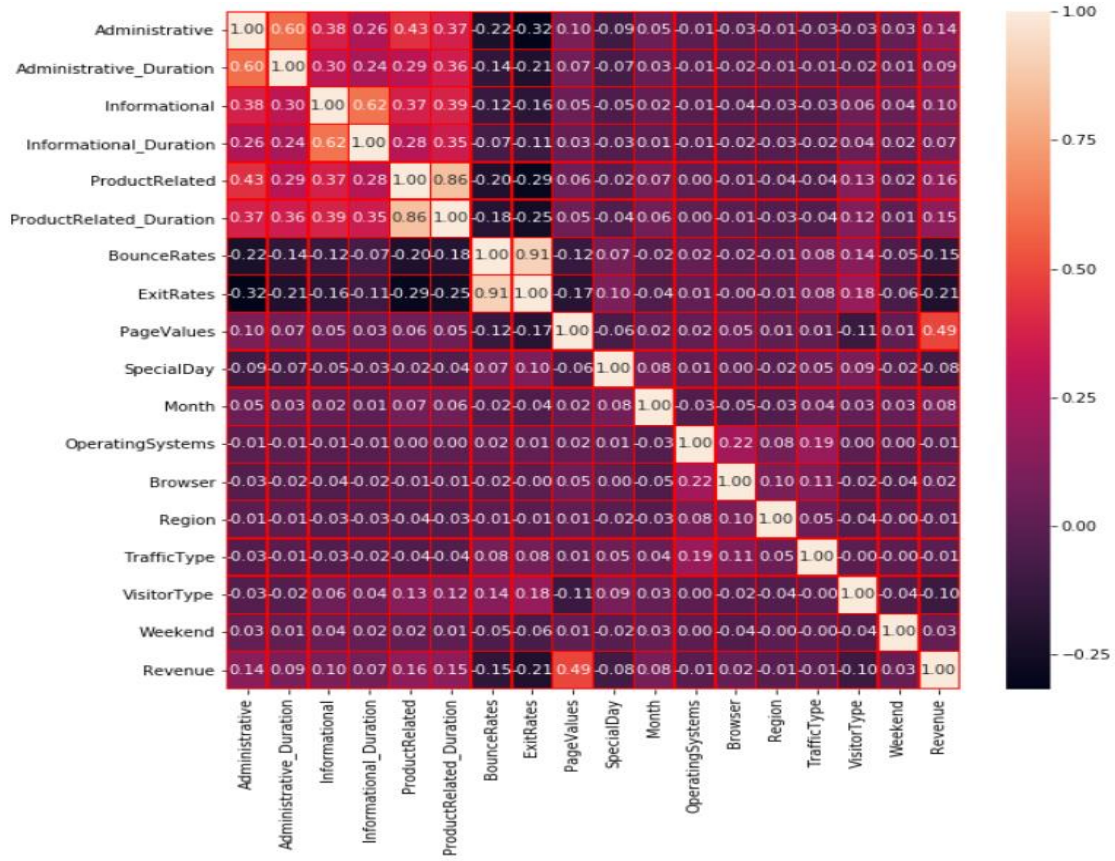
Histogram to Investigate the Distribution of Numerical Features

When it comes to check if there is missing data, I simply used the code below and investigated there are no missing data in the dataset.

```
In [169]: dataset.isna().sum()          # To observe whether there is missing data or not

Out[169]: Administrative             0
          Administrative_Duration    0
          Informational              0
          Informational_Duration     0
          ProductRelated             0
          ProductRelated_Duration    0
          BounceRates                0
          ExitRates                  0
          PageValues                 0
          SpecialDay                 0
          Month                      0
          OperatingSystems           0
          Browser                    0
          Region                     0
          TrafficType                0
          VisitorType                0
          Weekend                    0
          Revenue                    0
          dtype: int64
```

Missing Data Investigation



Correlation Heat-map

Then I looked at the heat-map to see the correlation relationship of the variables and the target variable. Some key points in the correlation matrix are:

- PageValues has the highest positive correlation of 0.49 with the target label
- ExitRates has the highest negative correlation of -0.21 with the target label
- ProductRelated is 0.86 positive correlated with ProductRelated_Duration
- BounceRates is -0.91 negatively correlated with ExitRates

Data Pre-Processing

I draw histogram of numerical variables and adjusted their distribution to obtain better performance by using standard scaler and yeo-Johnson transformation. So that the numerical features became more similar to normal distribution. Furthermore, I performed an outlier capping algorithm which capped the outliers that exceed minimum and maximum values of the determined capping points. There were no missing values in the data. I performed anova and chi tests for numerical and categorical variables respectively. I did not drop any features since I saw even the least useful feature had an impact (very little) and there are not much data. After applying feature selection, I decided to use one-hot encoder to fit my models since one-hot encoder gives better results for some models. After that, I split the model so that I can train the models via train data and test the performance of the models by using test data.

Mert Kaya
C2100965

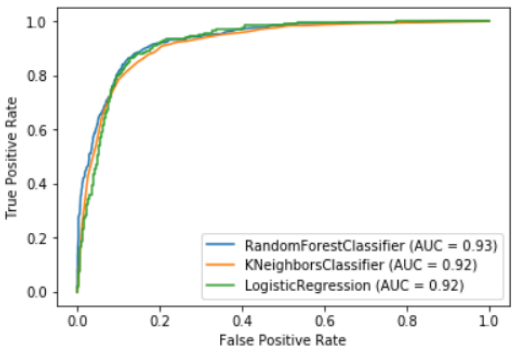
Model Implementation

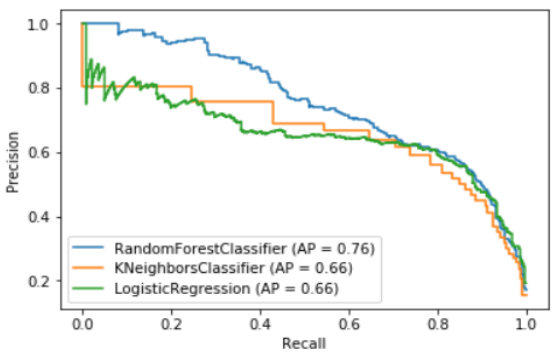
In this project, in order to perform undersampling and oversampling, I created dictionaries and put various techniques to perform at once, so that I would be able to compare their performance by looking at roc-auc curve and pick one of them. I used randomforestclassifier since it increases the randomness of the model, combines the prediction results of a lot of decision trees and gives high performance in overall. By comparing the performances of the undersampling method, I picked OneSidedSelection for undersampling. Then I performed hyper parameter tuning with grid search to find better performing parameters. I did same process for the k-nearest neighbors by creating a dictionary which includes various oversampling methods. I picked k-nearest neighbors because after I used box-cox transformation on the numerical data, I saw some of the data became similar to normal distribution and I knew that k-nearest neighbors algorithm gives nice results with data which is normally distributed since it uses distances in the model. When it comes to logistic regression, sometimes saying the probability of a customer's transaction label is more valuable than classifying it (and in many various businesses) and I knew that probability calibration of logistic regression is generally gives good results so I preferred to use logistic regression. I implemented cost sensitive learning, since most of the cases, predicting the true labels wrong costs much more than predicting the false case correctly. I used class weight balanced which rebalance the dataset utilizing balancing ratio. I also tested various cases too to evaluate the performances. I was also curious about the performances of all 3 combined (voting-classifier) and ada-boosting classifier so I compared both as well.

Performance Evaluation

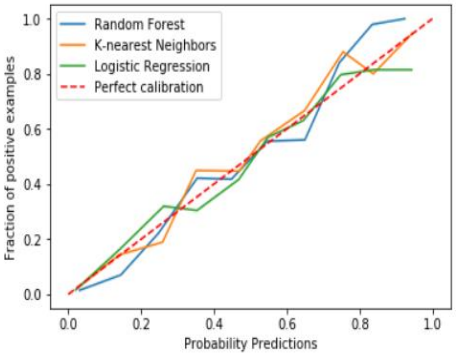
	Classifier_Type	f1_scores	Threshold Probability
0	RandomForest	0.700353	0.34
1	K_nearest_Neighbors	0.674473	0.80
2	Logistic_Regression	0.687075	0.66

I preferred roc-auc curves, precision-recall curves and maximum f1 scores to compare the models because roc-auc presents how well a classifier can separate positive and negative labels(after the imbalanced problem is solved). RF Classifier with undersampling gave the best result for it. When it comes to precision-recall curve to see the relationship of precision-recall for different thresholds I also compared them and the RF Classifier gave the best performance. When I compared the f-scores for all models with different threshold probabilities, again I saw the best model performance in RF Classifier as well with f-score = 0.70 approximately.

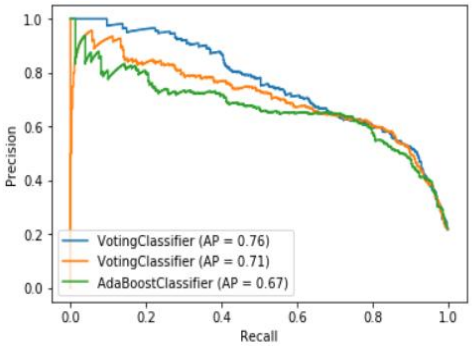


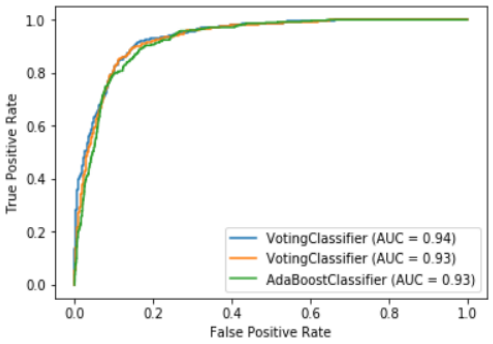


Furthermore, I compared the probability calibration of the models because sometimes saying the probability of an occasion is much better instead of providing 0 or 1. By looking at the probability of the classification, the resources of the business can be allocated more efficiently. In many areas such as diagnosing the cancer patients, saying the probability of patient having the cancer is 80% is more useful than just presenting 0 or 1 labels.



I was curious about the performance of ensemble learning methods so I combined all of 3 the models I used (RF, Nearest-Neighbors, Logistic Regression) under votingClassifier and I observed AUC = 0.94 and precision-recall ratio of 0.76. I compared it with Adaboost and oversampled version of votingClassifier.





Result Analysis and Discussion

In this section I am going to make some inferences to compare my models. First if I make the assumption the cost of False Positives are not that much since I am aiming to detect whether the e-commerce visitors have purchasing intentions or not. So I can focus on improving the recall value which will decrease precision because there is a trade-off between them. So I tested various probability thresholds to adjust better recall. So, I can make cost evaluation by changing the probability thresholds and take the best model according to my cost calculations. Since randomforestclassifier has the best AUC, precision-recall value and highest f-score I will pick it and adjust the threshold value according to my cost calculation of the business. But when it comes to provide probability of detecting whether the customer has buying intentions, I can pick the logistic regression classifier since it has the best probability calibration curve compared the both of them. Normally logistic regression classifier gives very good performance for the probability calibration curve.

When I combined all 3 models with voting classifier, I observed the best performance for the AUC = 0.94 and precision recall area as 0.76. So when I compare the ensemble learning methods, voting classifier with undersampled data gave the best results and again by adjusting the probability threshold to obtain higher recall, I can take the best metric according to my cost. Most of the cases predicting the true label is more important so I can optimize the recall by using voting classifier with undersampled data.

General Work-Flow Diagram