CIND 119: Introduction to Big Data Analytics
Final Project

Bank Marketing Dataset

Project content:

1. Summary

1.1. Purpose: The main objective of this project is to implement the CIND 119 course learning outcomes regarding classification techniques of *Decision Tree* and *Naïve Bayes* on the given dataset (from a Portuguese Bank). Through classification we would like to know that a certain customer with a set of demographics fall into what final class attributes (of subscribing to a long-term accounts or not) and compare the performance metrics and results of various supervised methods on both the original and filtered dataset.

1.2. Dataset Selection and Summary of Dataset: A dataset from a Portuguese Bank was chosen for this project. The bank dataset contains 17 attributes including 10 categorical/qualitative attributes (job, marital status, education, default, housing, loan, contact, month, poutcome, and y) and 7 quantitative attributes (age, balance, day, duration, campaign, pdays, previous). The class attribute is y which is binary in forms of yes and no.

1.3.Tools/Methodology: The machine learning (ML) project was conducted using python notebook in the google colab platform. Various libraries/modules in python (pandas, scipy, numpy, sklearn, matplotlip, seaborn) were utilized for the preprocessing techniques (checking for null and extreme values, balancing the unbalanced dataset, normalizing quantitative values, encoding qualitative attributes, and suppressing the negative values). Then, the python script was implemented on the filtered data. Among the options available for classification techniques (supervised learning), four methods (Decision Tree, Random Forest, Naïve Bayes, K-Nearest Neighbor) were selected and their performance metrics were compared together to distinguish the best method. Finally, the parameters of the best technique were optimized (tuned) and the model was saved. Data visualization were supplemented to provide a sense of character for attributes and their values as well. Parallel to this process, the same python commands were applied on the original data to highlight the importance of preprocessing steps and illustrates the differences in the outcomes (section 5.7).

1.4. Findings: Based on comparison the Random Forest technique best represented the classification with the best performance metrics. After that, Decision Tree and K-Nearest

Neighbor were the best models and Naïve Bayes showed the lowest performance metrics for this classification problem. After tuning the hyper parameters for the Random Forest model, the outcomes from the original Random Forest model was compared with the optimized model and stipulated that the original Random Forest model can predict the attribute class more precisely.

2. Workload Distribution

Member Name (Group 9)	**List of Tasks Performed**
Zeinab Khansari	We have conducted all stages of this project together!
Fatemeh Kamyabkalantari	

3. Introduction and Project Framework

Because of the competitive nature of consumer markets, applications of machine learning and data mining have attracted a significant attention from around the world. Making sense of data/data mining in direct marketing such as phone marketing and email marketing has been the subject oof researches extensively for the last two decades (Viaene et al., 2001; Baesens et al., 2003; Kim et al.,2005; Crone at al., 2005). The main goal of data mining and machine learning in direct marketing projects is to detect the costumers that most likely end up purchasing product and/or services offered by the company. So, the company can optimize their effort for maximizing its profits. Based on the similar philosophy, in this project the main objective is to classify the costumers based on the recorded observations and predict the response of the prospective costumer based on already available demographics of other costumers that have been analysed and modeled. Figure 1. Illustrates the schematic structure of this project for each round of modeling.

As shown in Figure 1, after data selection, the preprocessing techniques including checking for null and extreme values, balancing the unbalanced dataset, normalizing quantitative values, encoding qualitative attributes, and suppressing the negative values were implemented on dataset. Then, the filtered dataset was split into two classes of train set and test to later on be used for the classification models including Decision Tree, Random Forest, Naïve Bayes, K-Nearest Neighbor.

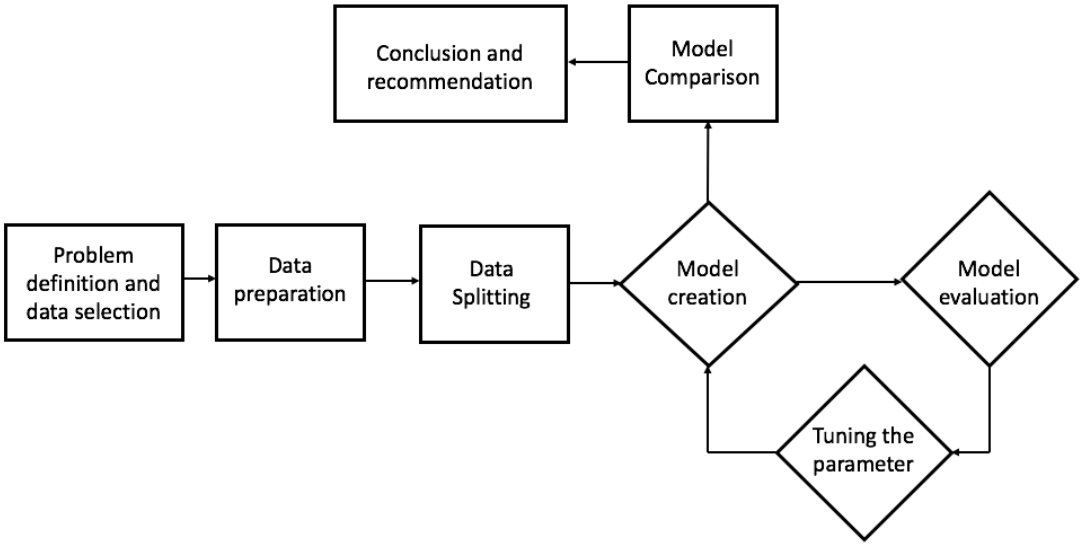

Figure 1. Schematic structure of the project flow.

4. Data Preprocessing

Datamining techniques are being used to turn the raw data into useful information. Data preprocessing is a prerequisite in efficient knowledge discovery and thus effective data mining (Crone et al., 2006). There is evidence in the extant literature representing that the datasets associated with telecommunication and phone-marketing are highly unbalanced with the efficiency of around 5% (Yang et al., 2005). Hence, the preprocessing is required to transform these unbalanced datasets and make them appropriate for conventional classification approaches (Wang et al., 2005; Yang et al., 2005). However, the selection of the preprocessing techniques (noise reduction, normalization, random sampling for highly skewed attributes, etc.) depends on the nature of the dataset and varies from one dataset to another (Crone et al., 2006; Japkowicz and Stephen, 2002).

The dataset that has been considered as the input for this project, has been integrated from a Portuguese bank. This bank would like to sell long-term deposit accounts such as bonds and saving accounts through impactful and efficient telemarketing and phone communications. The dataset encompasses the following costumers' demographics along with their data class/type as represented in Table 1.

Table 1. Dataset for costumers' demographics and data type

No.	Attribute	Comment	Data Type
1	Age	Age of the customer	Numeric
2	Job	Type of job	Qualitative
3	Marital	Marital status	Qualitative
4	Education	Education of the customer	Qualitative
5	Default	Shows whether the customer has credit in default or not	Qualitative
6	Balance	Average yearly balance in Euros	Numeric
7	Housing	Shows whether the customer has housing loan or not	Qualitative
8	Loan	Shows whether the customer has personal loan or not	Qualitative/categorical
9	Contact	Shows how the last contact for marketing campaign has been made	Qualitative
10	Day	Shows on which day of the month last time customer was contacted	Numeric
11	Month	Shows on which month of the year last time customer was contacted	Qualitative
12	Duration	Shows the last contact duration in seconds	Numeric
13	Campaign	Number of contacts performed during the marketing campaign and for this customer	Numeric
14	Pdays	Number of days that passed by after the client was last contacted from a previous campaign	Numeric, -1 means client was not previously contacted
15	Previous	Number of contacts performed before this campaign and for this client	Numeric
16	Poutcome	Outcome of the previous marketing campaign	Qualitative
17	Y	Class attribute showing whether the client has subscribed a term deposit or not	Binary: "yes","no"

To see which one of these variables has the highest impact on the class attribute and find the sequence for the strength of their interrelationships the attribute analysis was performed on data using tableau software and weka software. Consequently, duration, poutcome and age were the most impactful attributes on the targeted class attribute: 'y'.

4.1. Checking for Null Values and Extreme Values

To check if there are any null values, and check the datatype in the dataset, the below code was run and as illustrated it showed that there are no missing values in the data frame. The dataset (bank.arff) was uploaded as 'df'. The result also represents that there are 7 numerical/qualitative attributes and 10 categorical/qualitative attributes in dataset.

Table 2. Checking for null values and evaluating datatype

Code: **df.info()** Alternatively, df.isnul().sum() can check the number of null values.		
Attribute	**No of non-null records**	**Data type**
Age	4521 non-null	float 64
Job	4521 non-null	object
Marital	4521 non-null	object
Education	4521 non-null	object
Default	4521 non-null	object
Balance	4521 non-null	float 64
Housing	4521 non-null	object
Loan	4521 non-null	object
Contact	4521 non-null	object
Day	4521 non-null	float 64
Month	4521 non-null	object
Duration	4521 non-null	float 64
Campaign	4521 non-null	float 64
Pdays	4521 non-null	float 64
Previous	4521 non-null	float 64
Poutcome	4521 non-null	object
Y	4521 non-null	object
dtype: float64 (7), object (10)		

Also, to examine the quartiles and statistical metrics such as min, max and standard deviation for quantitative values, the below line of code was executed:

Table 3. Checking extreme values

Code: **df.describe()**							
	age	**balance**	**day**	**duration**	**campaign**	**pday**	**previous**
Count	4521	4521	4521	4521	4521	4521	4521
Mean	41.17	1422.65	15.91	263.96	2.79	39.76	0.54
Std	10.57	3009.63	8.24	259.85	3.1	100.12	1.69
Min	19	-3313	1	4	1	-1	0
25%	33	69	9	104	1	-1	0
50%	39	444	16	185	2	-1	0
75%	49	1480	21	329	3	-1	0
Max	87	71188	31	3025	50	871	25

The results indicate that the number of datapoint is equal to 4,521, the mean value for age equals 41.17 years, with the standard deviation of 10.58, and minimum of 19 years old and maximum of 87 years old, the 25% quartile shows the 33 years old costumer with the median of 39 years old and third quartile (75%) of equal to 49 years old. The similar interpretations apply to other numerical attributes in the above result.

4.2. Balancing the Highly Skewed/Unbalanced Data and Altering Categorical Values

As mentioned earlier and was expected, the attribute class "y" is highly unbalanced/skewed. Therefore, the resampling was conducted to generate the balance set of data for further analysis. We checked the balance or unbalanced status of the dataset and found that there are 4000 no and only 521 in our class attribute (y). So, using the SMOTE library we resampled the dataset in 'y'. Moreover, since the SMOTE doesn't work with categorical data, besides altering the class attribute to 0 and 1 for yes and no respectively, we converted all categorical data (job, marital, education, default, housing, loan, contact, month, poutcome) to dummy variables as well.

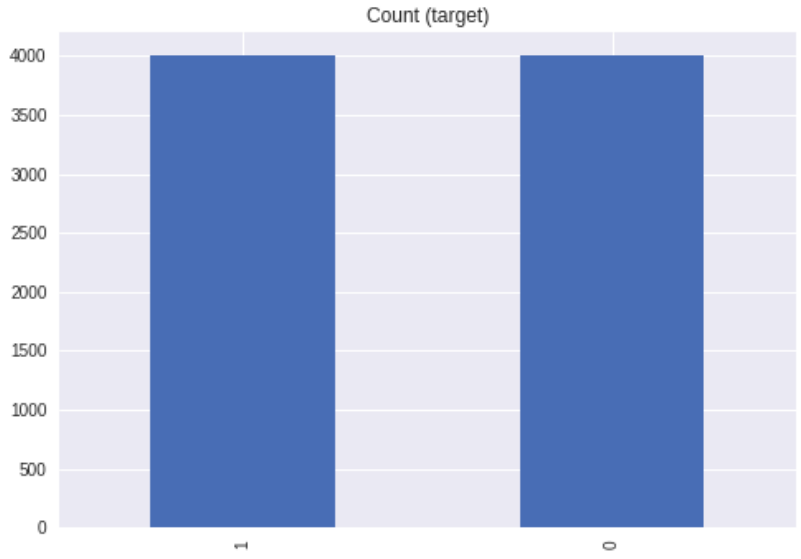

Figure 2. The results of resampling for class attribute ('y') and generating a balanced dataset

4.3. Scaling Values

To avoid the detrimental impact of the presence of outliers, we scaled our numeric values in the dataset using the sklearn.preprocessing module (MinMaxScaler). And then split our dataset into two sections: training dataset (70% of original dataset) and testing dataset (30% of the original dataset). Hence, the shape and the label for taring and testing datasets are as below:

Shape of training feature: (5600, 51)
Shape of testing feature: (2400, 51)
Shape of training label: (5600,)
Shape of training label: (2400,)

4.4. Suppressing Negative Values

Since one of the models that we will use for modeling is the Naïve Bayes and this model won't work with negative values we used lambda to suppress negative values and change them to zero in a copy of dataset only for the Naïve Bayes model. Table 4 represents the section of the data frame:

Table 4. the data frame after preprocessing

	Age	Balance	Day	Duration	Campaign	Pdays	Previous	Job_admin.
0	0.161	0.025	0.60	79	0	0	0	0
1	0.205	0.067	0.33	220	0	0.389	0.16	0
2	0.235	0.018	0.50	185	0	0.378	0.04	0
3	0.161	0.020	0.06	199	0.061	0	0	0
4	0.588	0	0.13	226	0	0	0	0

5. Predictive Modeling/Classification

After preprocessing techniques including:

(1) checking for null values,

(2) evaluating extremes values and quartiles,

(3) balancing the unbalanced data,

(4) altering categorical values to dummy values,

(5) scaling values to avoid the impact of the outliers in modeling, and finally,

(6) suppressing negative values,

now, we use four different modeling techniques encompassing:

(1) Decision Tree,

(2) Random Forest,

(3) Naïve Bayes, and,

(4) K-Nearest Neighbor

in order to classify our dataset and be able to predict the attribute class for training portion of the data frame.

5.1. Classification via Decision Tree

After implementing the Decision Tree model (training the model with training dataset and testing the model with testing portion of the dataset), the below metrics were calculated as shown in Table 5.

Table 5. Metrics for Decision Tree model

Accuracy	0.913
Precision	0.916
Recall	0.910
F1 score	0.913
Cohens Kappa score	0.826
Area under curve	0.913
Confusion matrix	1093 100 108 1099

5.2. Classification via Random Forest

Similar to the execution procedure for the Decision Tree model, we used sklearn.ensemble library (RandomForestClassifier) to implement the Random Forest model on the dataset. Table 6 illustrates the metrics for this model:

Table 6. Metrics for Random Forest

Accuracy	0.941
Precision	0.981
Recall	0.900
F1 score	0.939
Cohens Kappa score	0.883
Area under curve	0.987
Confusion matrix	1173 20 120 1087

5.3. Classification via Naïve Bayes

After applying Naïve Bayes model on the dataset, the following metrics were calculated to evaluate the model as depicted in Table 7:

Table 7. Metrics for Naïve Bayes model

Accuracy	0.776
Precision	0.815
Recall	0.719
F1 score	0.764
Cohens Kappa score	0.553
Area under curve	0.874
Confusion matrix	996 197 339 868

5.4. Classification via K-Nearest Neighbor

Also, we used K-Nearest Neighbor model for further comparision and discussion and as illustrated in Table 8 the below metrics were generated:

Table 8. Metrics for K-Nearest Neighbor model

Accuracy	0.791
Precision	0.730
Recall	0.927
F1 score	0.817
Cohens Kappa score	0.582
Area under curve	0.847
Confusion matrix	780 413 87 1120

5.5. Discussion on Performance Metrics

Usually the performance of a model is evaluated using accuracy, precision, recall and F1 score. Confusion matrix provides series of values, namely: True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). Figure 3 represents the features of a typical confusion

matrix. Using this values, accuracy, precision, recall and F1 score for classification models can be calculated.

Predicted Class

Actual Class		Class = Yes	Class = No
	Class = Yes	TP	FN
	Class = No	FP	TN

Figure 3. A typical confusion matrix

Although accuracy perhaps seems the most intuitive performance measure, however, it is very determinative only when we have a symmetrical dataset with near equivalent in terms of false positive and false negative. Hence, for most of dataset when there is no symmetry, other performance metrics should be considered as well.

$$\text{Accuracy} = (TP+TN)/(TP+FP+FN+TN) \qquad \text{Equation 1}$$

Precision shows the correctly predicted positive observations portion of the total predicted positive observations.

$$\text{Precision} = TP/(TP+FP) \qquad \text{Equation 2}$$

Recall or Sensitivity represents the ratio of correctly predicted positive observations to the all observations in actual class being yes.

$$\text{Recall} = TP/(TP+FN) \qquad \text{Equation 3}$$

To consider the impact of both precision and recall, the F1 score and is usually more considered (compared to accuracy) to judge and evaluate the performance of the model as it combines the impact of both recall and precision.

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision}) \qquad \text{Equation 4}$$

Kappa score measures the agreement between the classification by machine learning model and the manually established model. Kappa score changes between 0 and 1. The more the Kappa score is close to 1 the more the machine learning model is in agreement with the true values in real classification and vice versa.

The ROC (Receiver Operating Characteristics) curve, is a probability curve, representing how much the machine learning model is capable of distinguishing between the predicted classes. AUC or Area Under the Curve, demonstrates the area under the ROC curve that means the more the

AUC (probability of correctly distinguishing between classes) goes close to 1, the higher the performance of the machine learning model.

Comparing the evaluation results for all of our four models reveals that

- The highest accuracy belongs to the Random Forest model, and the lowest accuracy belongs to Naïve Bayes model.
- The highest precision is for the Random Forest model and the lowest precision belongs to K-Nearest Neighbor.
- The highest recall belongs to the K-Nearest Neighbor and the lowest recall was observed for the Naïve Bayes model.
- The highest F1 score is for the Random Forest model and the lowest F1 score belongs to Naïve Bayes model.
- The highest Kappa score was observed for the Random Forest model and the lowest Kappa score was observed for the Naïve Bayes model.
- The highest AUC belongs to Random Forest and the lowest AUC belongs to K-Nearest Neighbor model.

Based on the observations summarized above, the Random Forest Model seems the best model for this particular dataset and Naïve Bayes and K-Nearest Neighbor models have the lowest performance. Thus, the sequence of performance from highest to lowest is Random Forest, Decision Tree, and then both K-Nearest Neighbor and Naïve Bayes model.

The next section will examine, evaluate and compare these four models by plotting their ROC curves and comparing the performance metrics in a bar graph.

5.6. Model Comparison (Visual Comparison)

Using matplotlib and seaborn libraries in python, we visually showed the differences between evaluated performance metrics for all four machine learning models by comparing their ROC (Receiver Operating Characteristics) curves and bar diagrams of their performance metrics as shown in Figure 4.

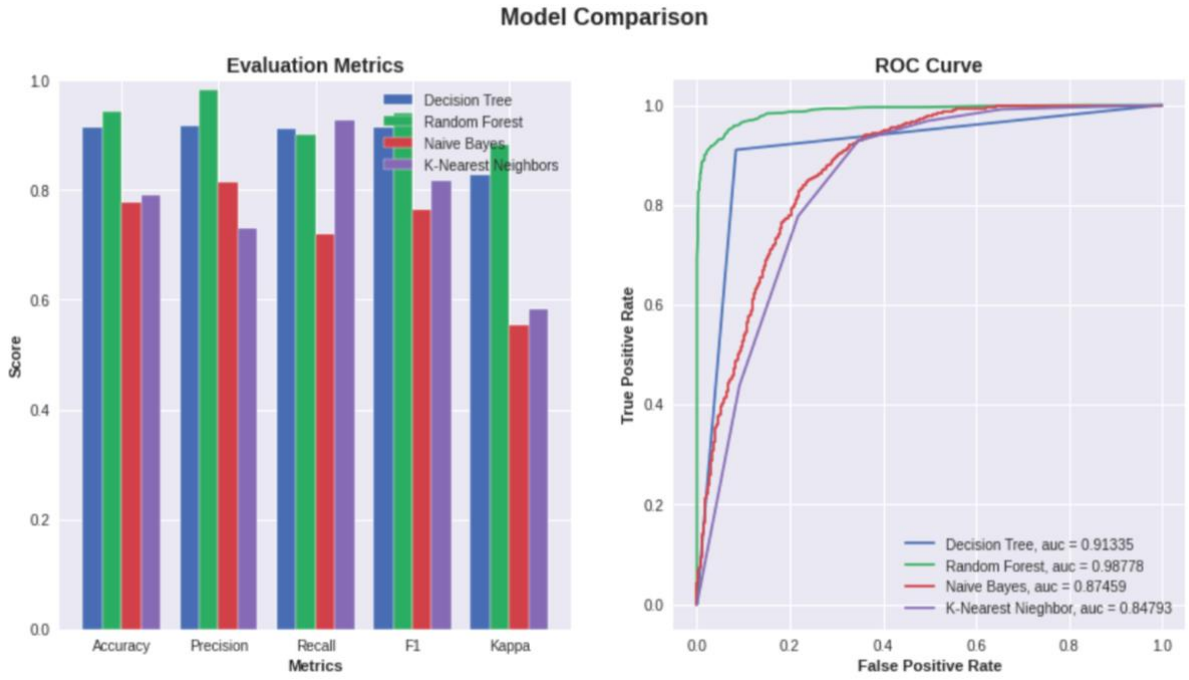

Figure 4. The differences between various machine learning techniques

The visual representation of the ROC curves and the performance metrics for different machine learning techniques that we used confirms and validates the observations we mentioned from the data represented in the previous section. By looking at these curves and bar charts, we can conclude that the Random Forest model is the best in representation the actual classification for the dataset and the lowest performance (lowest probability of correctly classify the data) belongs to both Naïve Bayes and K-Nearest Neighbor models.

5.7. Comparison between optimized model and original Random Forest

Since we figured out the highest performance belongs to the Random Forest model, we tried to optimize the model by tuning the hyper parameter to make the model even more precise. We used GridSearchCV in sklearn library and plotted the optimized Random Forest model against the baseline Random Forest model as represented in Figure 5.

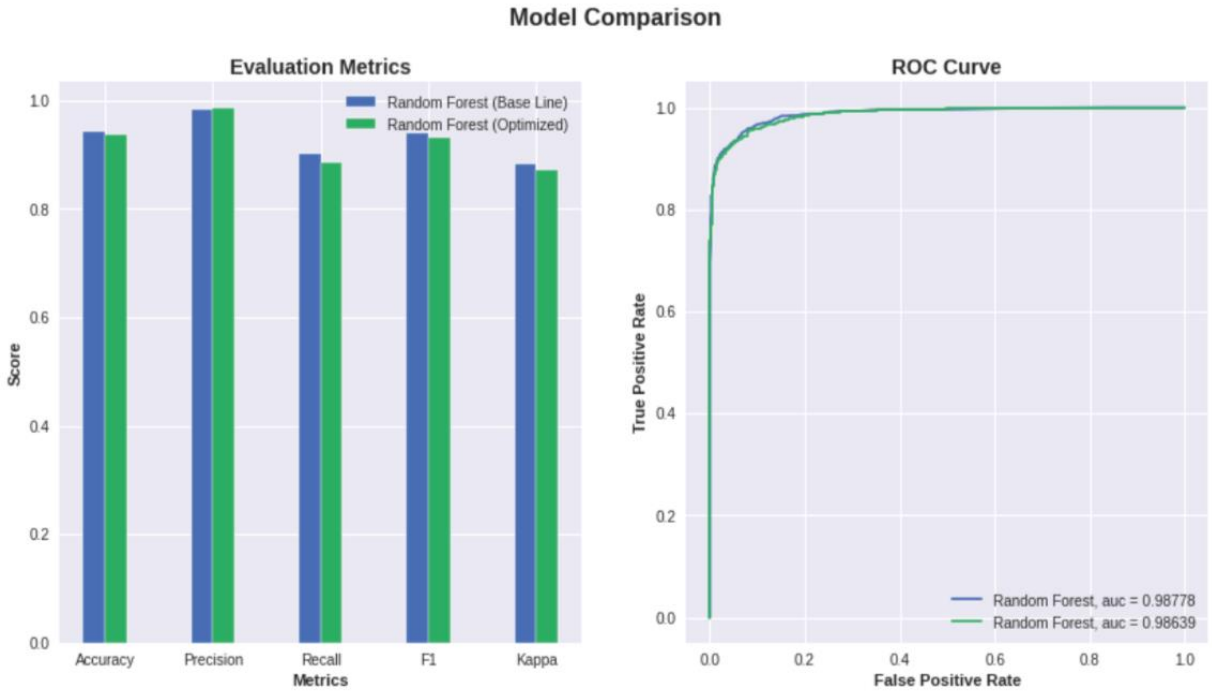

Figure 5. comparison between the optimized Random Forest and non-optimized Random Forest model

The outputs demonstrate that there is a negligible differences between optimized and non-optimized version of the Random Forest model and basically no improvement observed in the optimized model due to tuning the parameters.

5.8. Data saving

Finally, we saved the data by importing dump and load functions from joblib module in python.

6. Highlighting the Importance of preprocessing

To highlight the importance of preprocessing, we performed all the procedure except the preprocessing section for our four machine learning models. Table 9 shows the performance matric of all models without preprocessing and Figure 6 visually represents their differences.

Table 9. Performance metrics for all models without data preprocessing

Metric	Decision Tree	Random Forest	Naïve Bayes	K-Nearest Neighbor
Accuracy	0.86	0.88	0.47	0.87
Precision	0.45	0.67	0.15	0.47
Recall	0.49	0.20	0.72	0.18
F1 score	0.47	0.31	0.25	0.26
Cohens Kappa score	0.39	0.27	0.06	0.21
Area under curve	0.70	0.89	0.59	0.72
Confusion matrix	1087 101 86 83	1171 17 134 35	516 672 46 123	1154 34 138 31

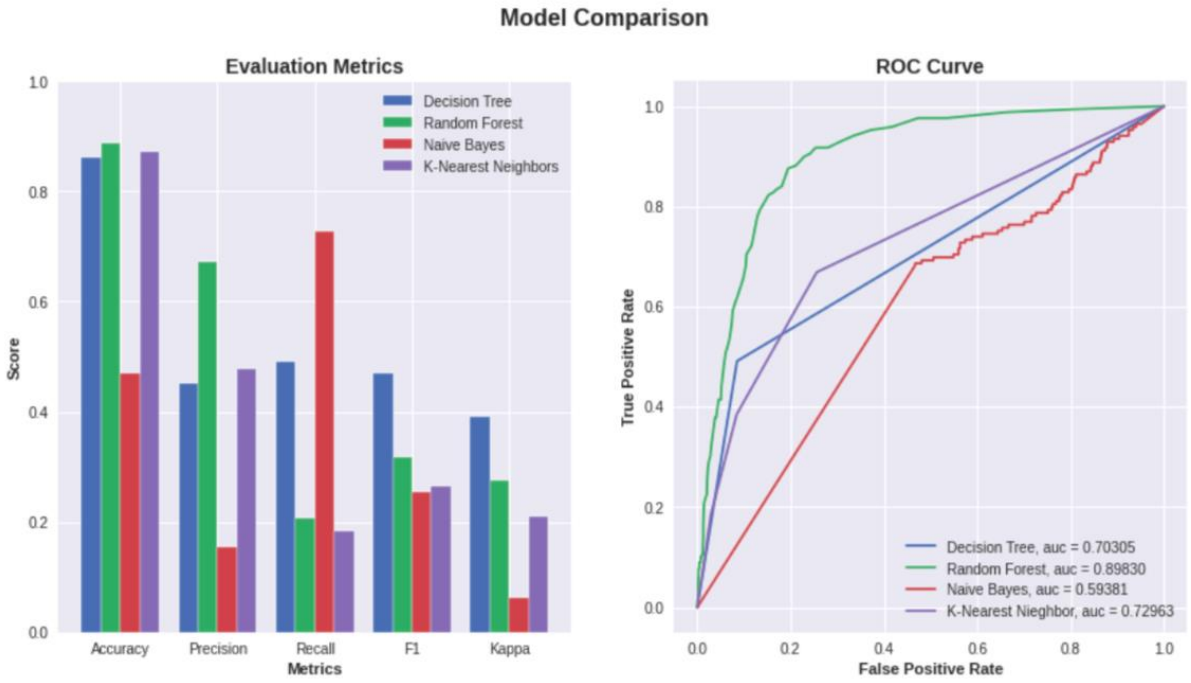

Figure 6. The ROC curves and performance metrics for different models without data preprocessing

From Figure 6 and table 9, we can conclude that preprocessing enhances the performance of a machine learning model significantly and remarkably distinguishes between the ROC curves and

probabilities of correctly distinguishing the class. Also, we can conclude that although the precision and recall were hugely influenced by preprocessing (comparing Figure 6 and Figure 4), however, the accuracy was affected the least by the preprocessing process.

7. Data Visualization with Tableau

To compare various attributes and see the visual impact of them on the attribute class, we used Tableau. Figure 7 shows the influence of being 'married' and 'account balance' on the class attribute and suggests that the success (yes – meaning the customer opening account) in married customer is almost 7 times higher than the divorced group and two times higher than the single group. Also, Figure 7 represents that the married group has the highest account balance and the divorced group has the lowest balance.

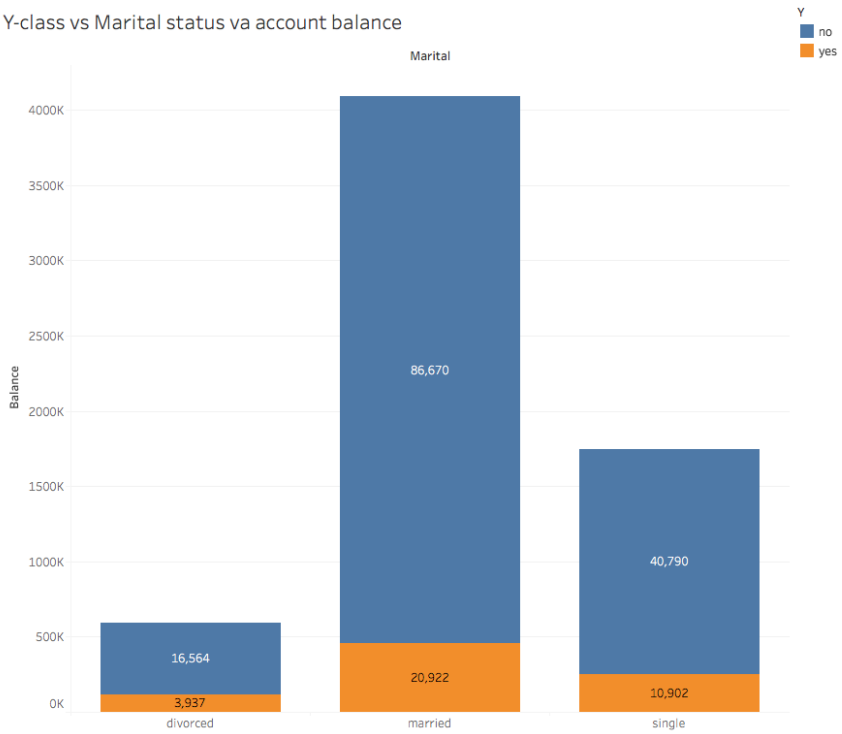

Figure 7. the impact of marital status and account balance on attribute class

Figure 8 illustrates the relationship between education and call duration through the telemarketing process. As seen in Figure 8, the highest duration belongs to the group with secondary education. As discussed earlier the call duration has a great impact on class attribute and the higher the duration, the higher the chance of having 'yes' as a class attribute. Hence, it seams most people who have a 'yes' as their class attribute belongs to the group with secondary education.

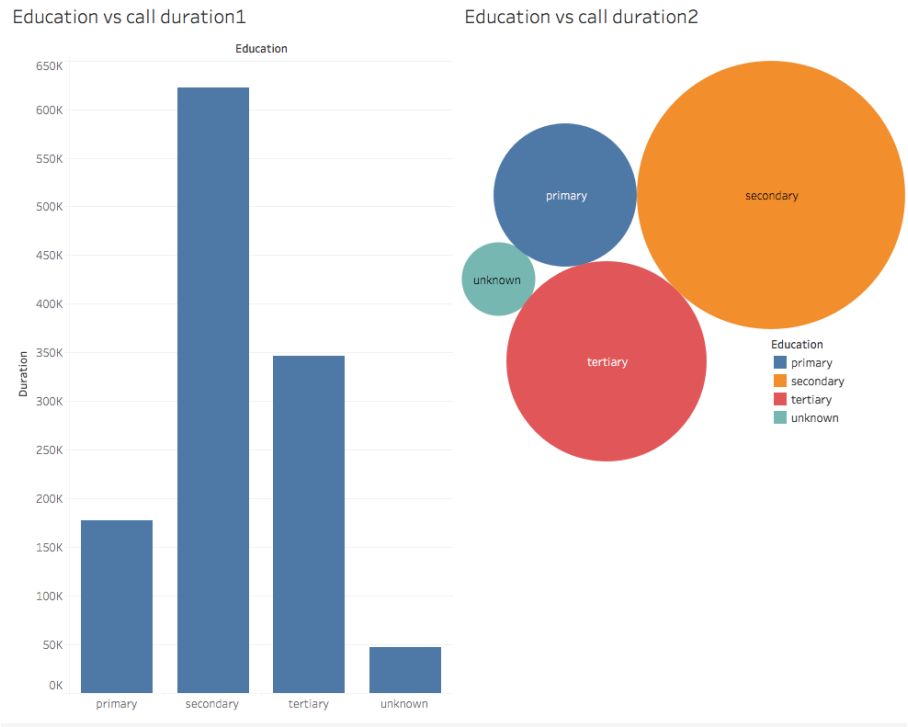

Figure 8. Education and call duration relationship

Figure 9 represents the impact of 'previous' and 'loan' on attribute class and shows people who don't have loan are more willing to open long-term account than people who are having loans. Figure 9 also represents the more the number of contacts during previous campaign to a specific customer the less the customer is willing to open an account. Therefore, the number of calls has an adverse effect on the attribute class.

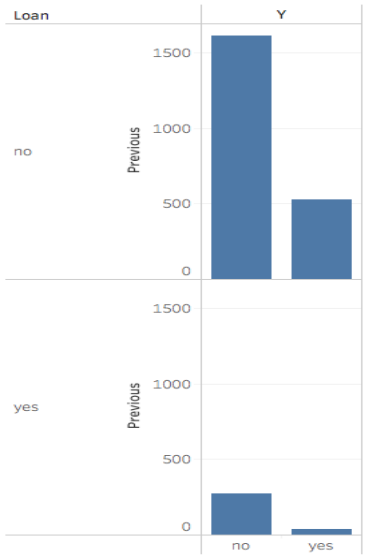

Figure 9. The impact of the number of contact for a specific customer during previous campaign and the existence of loan on the class attribute

Figure 10 represents the impact of the customers' job on the attribute class and as shown, the highest probability for having a yes attribute class is for management, technician, and then admin jobs.

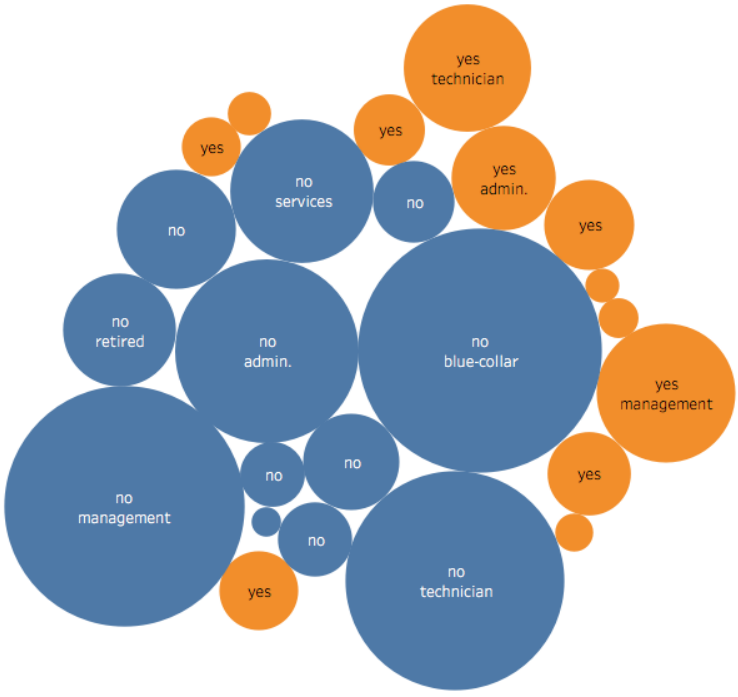

Figure 10. The influence of jobs on the attribute class.

8. Conclusions and Recommendations

The machine learning and data mining can brings raw data to life, somehow that numbers can communicate with people during the decision making process and provide valid information and predict the outcomes of various probable decisions or attitude. The worlds of telemarketing and other means for direct marketing (email, mail, etc.) can immensely benefit from machine learning as they can allocate their effort on attracting customers with the highest chance of making ties for purchasing a product or committing to a service.

In this project we want to identify the customers who open a long term account (class attribute = yes) based on various demographics that were provided to us. We implemented six preprocessing techniques including (1)checking for null values, (2) evaluating extremes values and quartiles, (3) balancing the unbalanced data, (4)altering categorical values to dummy values, (5) scaling values to avoid the impact of the outliers in modeling, and finally, (6) suppressing negative values and four different modeling techniques encompassing (1) Decision Tree, (2)Random Forest, (3)Naïve Bayes, and, (4) K-Nearest Neighbor and evaluated the results. According to the outcomes of our project, for this specific Portuguese Bank, the Random Forest model provided the best performance in terms of correctly predicting the class. Naïve Bayes and K-Nearest Neighbor depicted the least accuracy and performance in classification of this dataset. We also figured out the data preprocessing hugely impacts the performance of all four techniques of classification that we used. Hence, executing a thorough and precise preprocessing methods can immensely increase the reliability and performance of the classification model and thus is of great importance.

Also, we would like to present the following recommendations for more efficient telecommunication/telemarketing:

- Focusing more on married costumers
- Focusing on people with secondary education
- Focusing on customers with managerial job or in technician group and admin groups
- Keep the number of contact in an optimum range and avoid calling a specific customer many times
- Focus on customers who don’t have loan.

9. Reference:

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring, Journal of the Operational Research Society 54 (6), pp. 627–635.

Crone, S.F., Lessmann, S., and Stahlbock, R. (2006). The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing, *European Journal of Operational Research* (173), pp. 781–800.

Japkowicz, N., and Stephen, S. (2002). The class imbalance problem: A systematic study, *Intelligent Data Analysis* 6 (5), pp. 429–450.

Kim, Y.S., Street, W.N., Russell, G.J., Menczer, F. (2005). Customer targeting: A neural network approach guided by genetic algorithms, *Management Science* 51 (2), pp. 264– 276.

Viaene, S., Baesens, B., Van Gestel, T., Suykens, J.A.K., Van den Poel, D., Vanthienen, J., De Moor, B., Dedene, G. (2001). Knowledge discovery in a direct marketing case using least squares support vector machines, *International Journal of Intelligent Systems* 16 (9), pp.1023–1036.

Wang, K., Zhou, S., Yang, Q., and Yeung, J.M.S. (2005). Mining customer value: from association rules to direct marketing, *Journal of Data Mining and Knowledge Discovery*.

Yang, Y., Yang, Q., Lu, W., Pan, J., Pan, R., Lu, C., Li, L., and Qin, Z. (2005). Preprocessing Time Series Data for Classification with Application to CRM, in *AI 2005*, LNAI 3809, Zhang, S. and Jarvis, R. (Eds.), pp. 133–142.