INFO 7390 Final Project

Team members:

1. Nidhi Goyal
2. Chaitanya Nakhare
3. Utkarsh Kakkar
4. Aaradhy Sharma

Problem Statement:

With a term deposit, you lock away an amount of money for an agreed length of time (the 'term') – that means you can't access the money until the term is up. In return, you'll get a guaranteed rate of interest for the term you select, so you'll know exactly what the return on your money will be. There has been a revenue decline for the Portuguese bank, and they would like to know what actions to take. After investigation, they found out that the root cause is that their clients are not depositing as frequently as before. Knowing that term deposits allow banks to hold onto a deposit for a specific amount of time, so banks can invest in higher gain financial products to make a profit. In addition, banks also hold better chances to persuade term deposit clients into buying other products such as funds or insurance to further increase their revenues. As a result, the Portuguese bank would like to identify existing clients that have a higher chance to subscribe for a term deposit and focus marketing effort on such clients.

Importance:

The loss of revenue in the bank has declined the bank of deposits to invest into other places and gain profits out of the deals to provide further interests to the customers term deposits. This has created a potential challenge for the bank to function proficiently, providing better interests and to generate revenues. This has increased and given rise to the need of analyzing the data categorically and visualizing the distributions to see what the aspects are wherein the bank should focus and implement the business methods and models to improve the process. The

whole process in turn demands the implementation of the business methods that best fits for the Portuguese Bank.

Objective:

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

Data:

http://archive.ics.uci.edu/ml/datasets/Bank+Marketing#

Portuguese Bank Marketing Data - The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

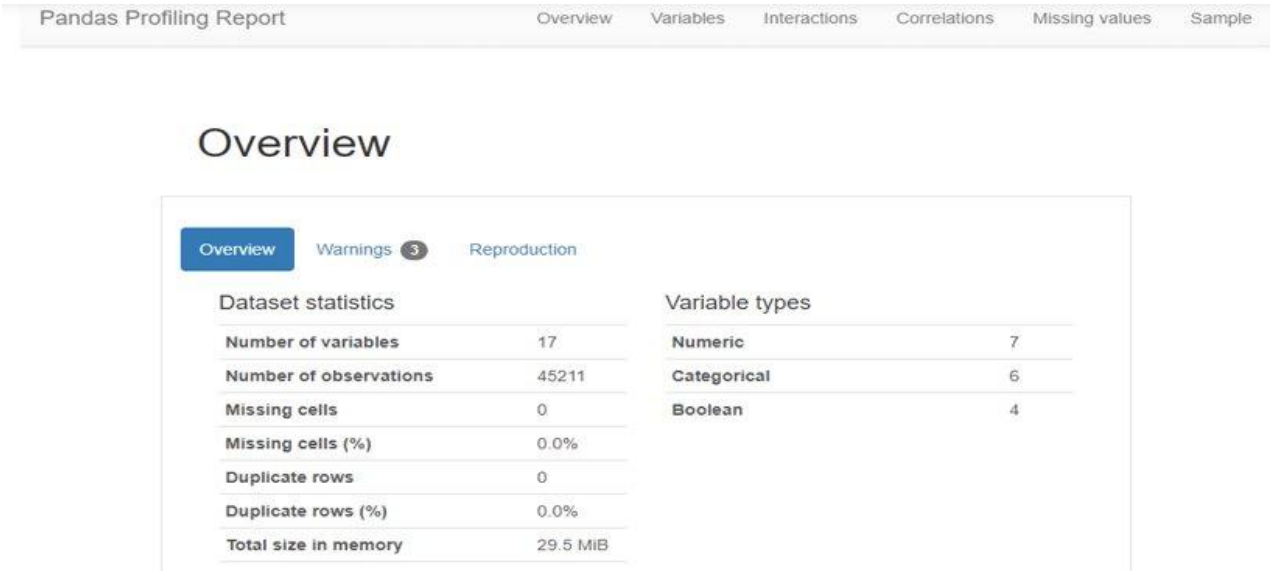
Domain: Banking

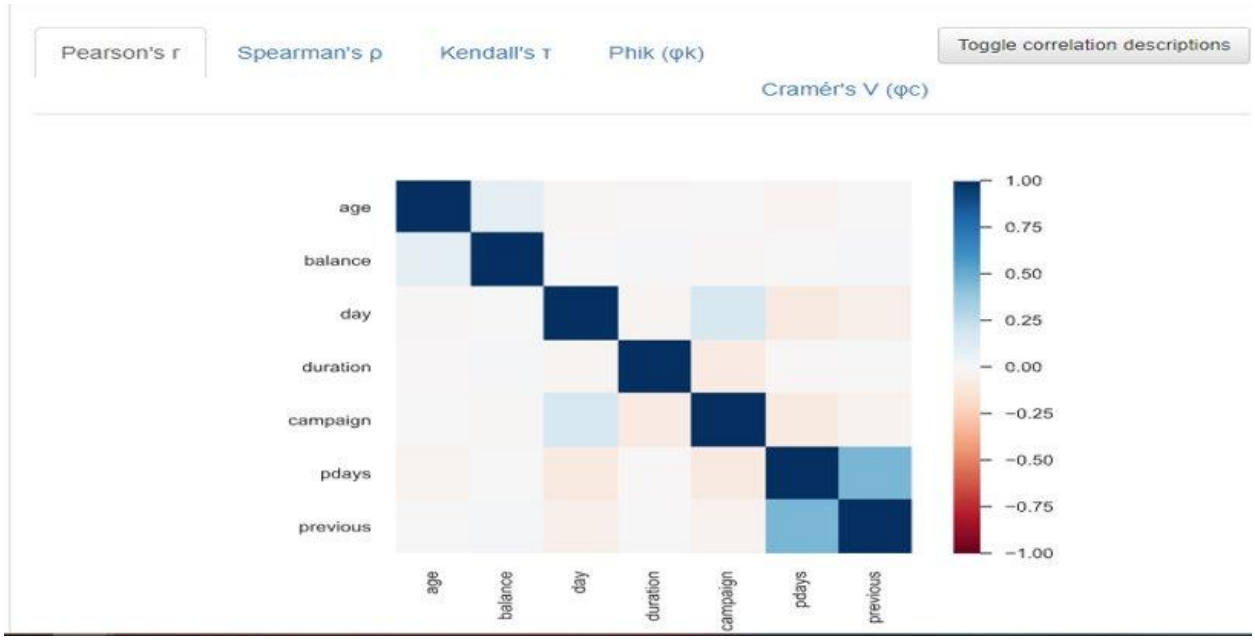
Steps:

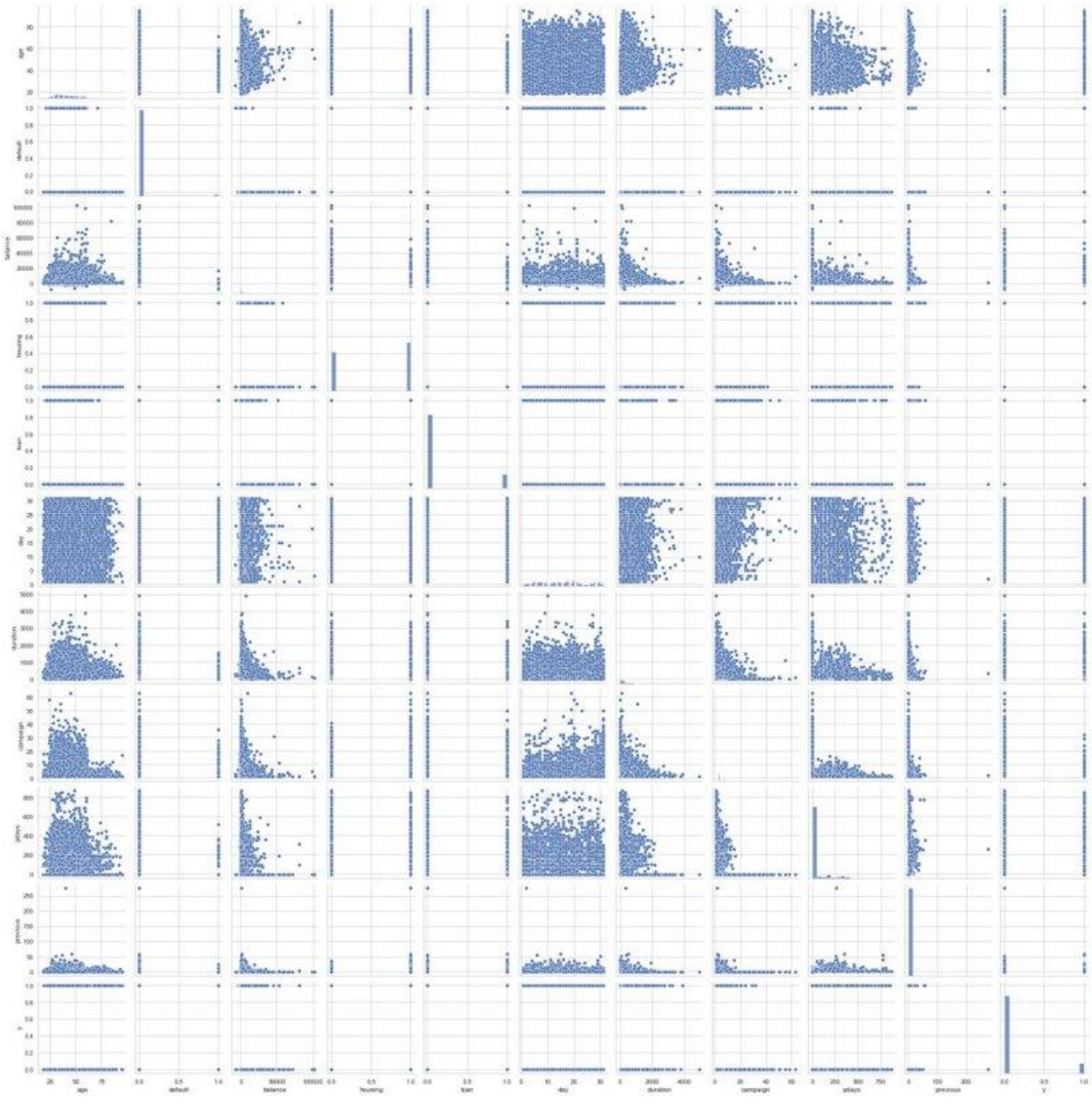
1. Install required libraries
2. Import Required Modules
3. Data Input
4. Exploratory Data Analysis - We are using pandas profiling for data distributions and correlations between the data.
5. Model Implementation - We proposed following algorithms to build a robust business method by finding the accuracy of all the models for the bank:
    - Logistic Regression
    - K-Nearest Neighbors Algorithm - KNN
    - Random Forest Algorithm
    - SVM
    - Decision Tree
    - Neural Networks

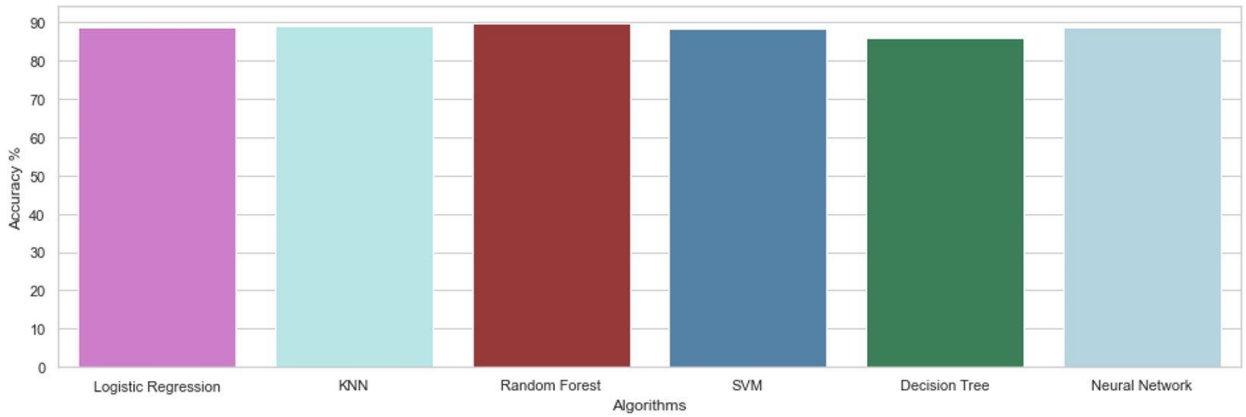
6. Scaling - Understood the model accuracy change with multiple types of scaling. Types of scaling performed are:
    - Standard Scaling
    - Min Max Scaling
    - Robust Scaling
7. Model Comparison
8. Confusion Matrix
9. Model Comparison with PyCaret

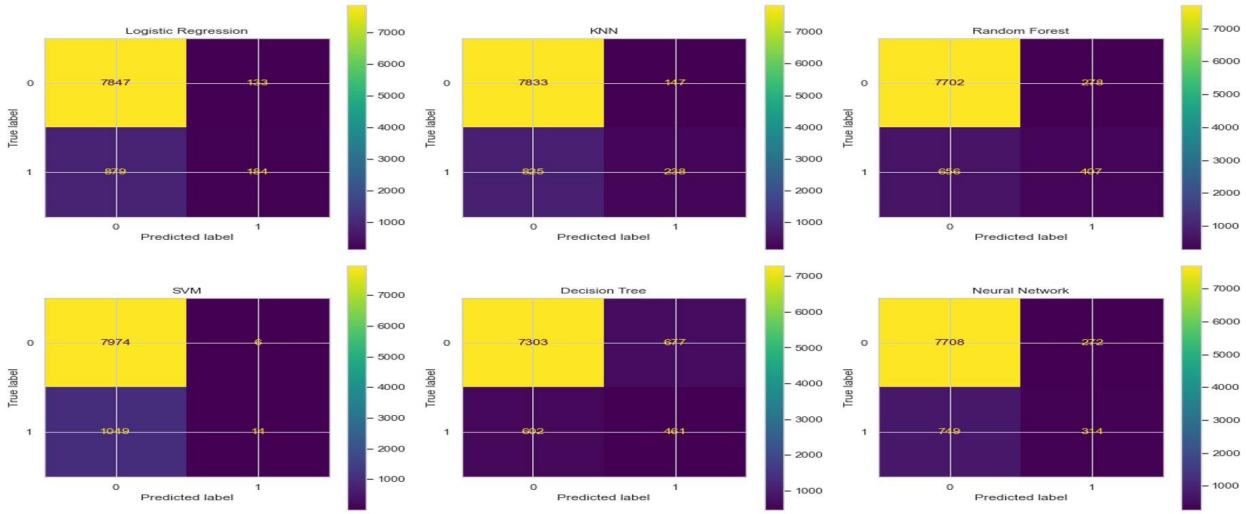
Project Images:



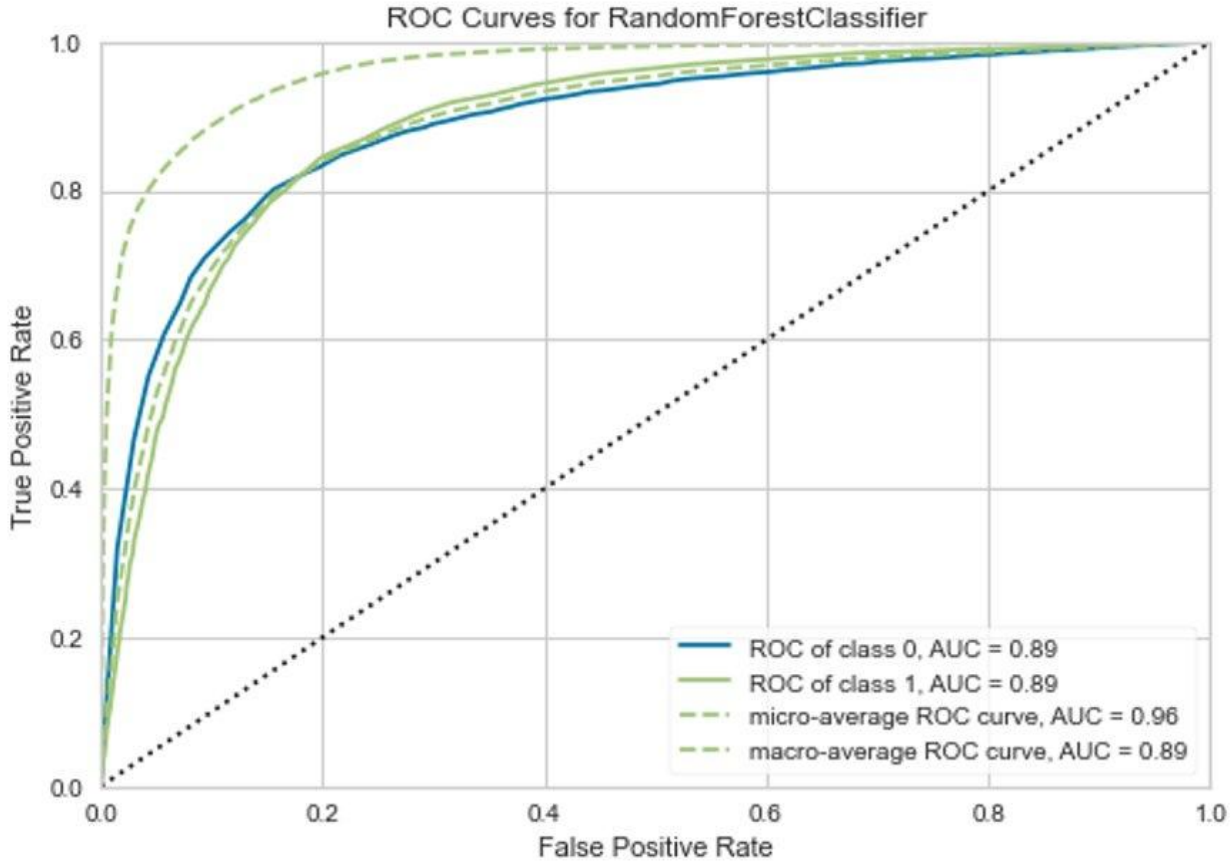

Pearson's r
Spearman's ρ
Kendall's τ
Phik (φk)
Cramér's V (φc)
Toggle correlation descriptions
age
balance
day
duration
campaign
pdays
previous
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


age
default
balance
housing
loan
day
duration
campaign
pdays
previous
y


Accuracy %
90
80
70
60
50
40
30
20
10
0
Logistic Regression
KNN
Random Forest
SVM
Decision Tree
Neural Network
Algorithms


Logistic Regression
7847
133
879
184
KNN
7833
147
835
228
Random Forest
7702
278
656
407
SVM
7974
6
1049
14
Decision Tree
7303
677
602
461
Neural Network
7708
272
749
314
True label
Predicted label
0
1
7000
6000
5000
4000
3000
2000
1000

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.8972	0.8929	0.3739	0.595	0.4588	0.4053	0.4189	1.625
rf	Random Forest Classifier	0.8956	0.8849	0.3476	0.5899	0.4368	0.3836	0.4002	0.885
lr	Logistic Regression	0.8882	0.852	0.1917	0.56	0.2853	0.2402	0.2815	0.416
knn	K Neighbors Classifier	0.8824	0.7576	0.2652	0.4925	0.3442	0.286	0.3028	0.093
dt	Decision Tree Classifier	0.857	0.6611	0.4056	0.3916	0.3982	0.3171	0.3173	0.084
svm	SVM - Linear Kernel	0.775	0	0.4647	0.3089	0.3123	0.2095	0.2413	0.096



Inference:

We achieved following observations after having data analyzed and visualized:

1. Education: Most people have university level education while illiterate people are very less.
2. **Random Forest classification** model has a high accuracy for the above dataset thus it would be apt to understand bank analysis.
3. For our case, **Fall-out or FPR** and **specificity** would be an important metric to choose a model as these will tell us how many customers did not actually subscribe for a term but the model predicted and labeled them as subscribed which would slash down banks business/revenue for that term.
4. **"Lower" specificity and fall-out** is ideal for this USE case. Hence, we choose the model that has better/higher accuracy score which is quintessential for a successful classification model given that it has a lower specificity and fall-out score.
5. Thus, our goal here would be the Fall-out and then implement it as our business model/USE case.
6. We can confidently say that having a high metric say accuracy doesn't necessarily imply a perfect business model. For the bank **XGBoost and Random Forest** would be a good way of classifying the customers.

Citations:

1. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
2. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
3. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
4. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
5. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
6. https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier

7. https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
8. https://seaborn.pydata.org/generated/seaborn.barplot.html
9. https://github.com/pycaret/pycaret
10. https://pycaret.org/compare-models/