IS-603 Decision-Making Support System

Project Final Report

Bank Product's Subscriber Prediction Analysis

Team Quadrons:
Swapnil Sahu
Tamilselvan Gurunathan
Suman Pogul
Smriti Khilnani
Mounika Cheera

Submitted to:
Dr. Sanjay Purushotham

TABLE OF CONTENTS

Abstract:

A Financial institution invests millions of dollars yearly in marketing campaigns to expand its customer base for the new products that it introduces each year. Based on data gathered by their marketing team, this expense is typically used for random customer marketing. The financial institution could save millions of dollars from random marketing and utilize the same for targeted consumer marketing if it had an effective data model that could forecast the precise clients subscribing to their new products. We have built an efficient classification model to predict the precise customer who would subscribe to the bank's new products.

Significance of the Topic:

- To analyze customers' requirements and recommend the relevant product.
- To provide services/products at competitive rates based on market analysis and inflation rate.
- To maximize the financial institution's profit margin.
- To offer other tax-saving options to customers based on their annual income.

Problem Statement:

Finding the right customers is a tedious and significant task for any bank to sell its products and services. To sell its products, the bank needs relevant insight into the customer's needs and some pre-requisite data like their age, occupation, annual income, etc. Even upon gathering the data, it's hard to predict whether a customer will subscribe to their services or not. And it becomes even more difficult when the customer base is in millions. Therefore, to alleviate this problem, our data model will predict whether a potential customer will buy the product or not. This will not only help sales representatives to target customers but also save costs from unwanted promotions. This project intends to use classification data mining tasks to predict whether the customer will subscribe to their product or not.

Overall Objective:

The objective is to perform classification tasks and predict whether the customer will subscribe to the product. Based on the data collected such as age, marital status, education, income, etc. the model will predict the outcome as "yes" or "no". This will help the financial institution to generate targeted marketing campaigns.

Implementation Details:

1. Data Acquisition: The Bank data is acquired from the UCI Data Repository. The link is - https://archive.ics.uci.edu/ml/datasets/bank+marketing

2. Data Preprocessing: The dataset was available in .csv format. As the data was unstructured, we converted the same in .xlsx format to get complete control of formatting. Below are the preprocessing steps we did as a part of data processing.

- Filling the missing data: We used the linear interpolation method to fill in the lost data. In this method, we take the average of the upper and lower numerical values and write the output value. As the empty cell might hinder the functioning of the model. We did the same to get accurate results.
- Finding Duplicate Values and Removing them: Minor duplicate values won't cause many problems but if they are in significant numbers, then they must be removed. In our project, we used conditional formatting to identify and remove duplicate values to save the integrity of the data. For large datasets, dedicated tools are available for serving the purpose, but since our data is manageable, we used MS Excel for the same.
- Changing the dataset format from (.csv) to (.arff): For loading the dataset efficiently to the WEKA tool, we converted the .csv file format to (.arff). The (.arff) is the ideal file format recommended for WEKA. Thus, the data was from the UCI repository, so it was easy for us to perform preprocessing.

Data Modeling:

A classification model was built in WEKA using the J48 algorithm using the training dataset. The data set consisted of 45000 instances. After preprocessing and data cleaning, the dataset dropped to 44600 instances. There are about 20 fields for an instance such as job, salary, previous credit, campaign status, etc. We have labeled the target class variable as “Subscribe”. This variable will be used to predict whether the customer will subscribe to the new product or not. The data model correctly classified 38728 instances, keeping its accuracy at 94.02 %. The confusion matrix is considered for further improving the accuracy of the model. We then tested the models using an actual dataset consisting of 12 instances. The model predicted the class label for the actual instances fed into it as represented in the below diagram fig 1.2. We then created multiple classification models using K-NN, Naïve Bayes, and logistic regression algorithms to compare the classification accuracy of the models. Below are the results of each model that were tested under two methods: the held-out data set method and the cross-validation method.

Model Results:

Correctly Classified Instances	38728	94.0274 %
Incorrectly Classified Instances	2460	5.9726 %
Kappa statistic	0.6836	
Mean absolute error	0.0918	
Root mean squared error	0.2142	
Relative absolute error	45.9028 %	
Root relative squared error	67.7541 %	
Total Number of Instances	41188	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.974	0.329	0.959	0.974	0.967	0.686	0.950	0.991	no
	0.671	0.026	0.769	0.671	0.717	0.686	0.950	0.761	yes
Weighted Avg.	0.940	0.295	0.938	0.940	0.938	0.686	0.950	0.965	

Fig 1.1 Classification accuracy of the model built using the training dataset.

The same model was used to test new data that hasn't been used to train the model and the model was able to predict the class variable. When compared with the actual value of the target variable to the predicted variable, the performance accuracy of the model stood at 95% classifying 12 instances that will subscribe to the bank's new product. Below are the predicted results.

=== Predictions on user test set ===

inst#	actual	predicted	error	prediction
1	1:?	2:yes		0.692
2	1:?	1:no		0.996
3	1:?	1:no		0.996
4	1:?	1:no		0.825
5	1:?	1:no		0.996
6	1:?	2:yes		0.6
7	1:?	1:no		0.996
8	1:?	1:no		0.996
9	1:?	1:no		0.996
10	1:?	1:no		0.996
11	1:?	1:no		0.825
12	1:?	2:yes		0.577

Fig 1.2 Predictions on the actual data using the classification model.

Similarly, below are the results for other models that were built to compare each model's performance accuracy, which will help us finalize the efficient model for the financial institution's objective.

Below are the results of the J48 algorithm under the cross-validation test method.

```
Correctly Classified Instances       37563               91.1989 %
Incorrectly Classified Instances      3625                8.8011 %
Kappa statistic                          0.5307
Mean absolute error                      0.1132
Root mean squared error                  0.2584
Relative absolute error                 56.6101 %
Root relative squared error             81.7357 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.959    0.462    0.942      0.959    0.951      0.533    0.884     0.967     no
                 0.538    0.041    0.627      0.538    0.580      0.533    0.884     0.539     yes
Weighted Avg.    0.912    0.414    0.907      0.912    0.909      0.533    0.884     0.918

=== Confusion Matrix ===

     a     b   <-- classified as
 35065  1483 |     a = no
  2142  2498 |     b = yes
```

Fig 1.3 Classification accuracy of Decision Tree classifier using cross-validation

The k-nearest neighbor's algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. Below are the results of the KNN algorithm under cross-validation and training set methods respectively.

```
Correctly Classified Instances       35944               87.2681 %
Incorrectly Classified Instances      5244               12.7319 %
Kappa statistic                          0.3152
Mean absolute error                      0.1273
Root mean squared error                  0.3568
Relative absolute error                 63.6876 %
Root relative squared error            112.8533 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.938    0.645    0.920      0.938    0.929      0.317    0.645     0.918     no
                 0.355    0.062    0.422      0.355    0.386      0.317    0.645     0.225     yes
Weighted Avg.    0.873    0.580    0.864      0.873    0.868      0.317    0.645     0.840

=== Confusion Matrix ===

     a     b   <-- classified as
 34298  2250 |     a = no
  2994  1646 |     b = yes
```

Fig 1.4 Classification accuracy of K-NN classifier using cross-validation

```
Correctly Classified Instances       41188              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0.0121 %
Root relative squared error              0.0077 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     no
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     yes
Weighted Avg.    1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000

=== Confusion Matrix ===

     a     b   <-- classified as
 36548     0 |     a = no
     0  4640 |     b = yes
```

Fig 1.5 Classification accuracy of K-NN classifier using the training set

Naïve Bayes Classifier is one of the simplest and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts based on the probability of an object. Below are the results of the naive Bayes algorithm under cross-validation and training set methods respectively.

```
Correctly Classified Instances       35951               87.2851 %
Incorrectly Classified Instances      5237               12.7149 %
Kappa statistic                          0.451
Mean absolute error                      0.1405
Root mean squared error                  0.3324
Relative absolute error                 70.2816 %
Root relative squared error            105.1481 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.905    0.383    0.949      0.905    0.927      0.458    0.871     0.978     no
                 0.617    0.095    0.453      0.617    0.522      0.458    0.871     0.481     yes
Weighted Avg.    0.873    0.350    0.893      0.873    0.881      0.458    0.871     0.922

=== Confusion Matrix ===

     a     b   <-- classified as
 33087  3461 |     a = no
  1776  2864 |     b = yes
```

Fig 1.6 Classification accuracy of Naive Bayes classifier using training Set

```
Correctly Classified Instances       35950               87.2827 %
Incorrectly Classified Instances      5238               12.7173 %
Kappa statistic                          0.4511
Mean absolute error                      0.1406
Root mean squared error                  0.3324
Relative absolute error                 70.3035 %
Root relative squared error            105.1389 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.905    0.383    0.949      0.905    0.927      0.458    0.871     0.978     no
                 0.617    0.095    0.453      0.617    0.522      0.458    0.871     0.482     yes
Weighted Avg.    0.873    0.350    0.893      0.873    0.881      0.458    0.871     0.922

=== Confusion Matrix ===

     a     b   <-- classified as
 33085  3463 |     a = no
  1775  2865 |     b = yes
```

Fig 1.7 Classification accuracy of K-NN classifier using cross-validation

Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of certain classes based on some dependent variables. In short, the logistic regression model computes a sum of the input features (in most cases, there is a biased term), and calculates the logistic of the result. Below are the results of the logistic regression algorithm under cross-validation and training set methods respectively.

```
Correctly Classified Instances       37535               91.1309 %
Incorrectly Classified Instances      3653                8.8691 %
Kappa statistic                          0.473
Mean absolute error                      0.122
Root mean squared error                  0.25
Relative absolute error                 60.9945 %
Root relative squared error             79.0595 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.973    0.575    0.930      0.973    0.951      0.488    0.936     0.991     no
                 0.425    0.027    0.667      0.425    0.519      0.488    0.936     0.601     yes
Weighted Avg.    0.911    0.513    0.901      0.911    0.902      0.488    0.936     0.947

=== Confusion Matrix ===

     a     b   <-- classified as
 35562   986 |     a = no
  2667  1973 |     b = yes
```

Fig 1.8 Classification accuracy of Logistic classifier using a test set

```
Correctly Classified Instances       37511               91.0726 %
Incorrectly Classified Instances      3677                8.9274 %
Kappa statistic                          0.4698
Mean absolute error                      0.1223
Root mean squared error                  0.2508
Relative absolute error                 61.1742 %
Root relative squared error             79.3332 %
Total Number of Instances            41188

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.973    0.577    0.930      0.973    0.951      0.484    0.935     0.991     no
                 0.423    0.027    0.663      0.423    0.516      0.484    0.935     0.597     yes
Weighted Avg.    0.911    0.515    0.900      0.911    0.902      0.484    0.935     0.947

=== Confusion Matrix ===

     a     b   <-- classified as
 35548  1000 |     a = no
  2677  1963 |     b = yes
```

Fig 1.9 Classification accuracy of Logistic classifier using cross-validation

Multiple classification models were built using KNN, Naive Bayes, and logistic regression to compare the performance and identify the efficient model. The receiver Operating Characteristic (ROC) curve is used for the graphical representation of the model developed. We have chosen the ROC curve as it is independent of the cost-benefit matrix and class priors. Varying the threshold value for the classifier creates a curve. As we can see the curve depicted illustrates that area under the ROC curve which means perfect performance accuracy. Each classifier has a separate curve and to analyze the performance of an efficient model, these curves must be plotted on a single graph. We used the WEKA tool to achieve the same. The WEKA tool has a knowledge flow environment that can be used to construct the process of building a model to visualize the same on the performance curve. This environment has built-in nodes that process the dataset and analyze the performance of each model. Below is the figure that depicts the knowledge flow used in this project to produce a ROC curve.

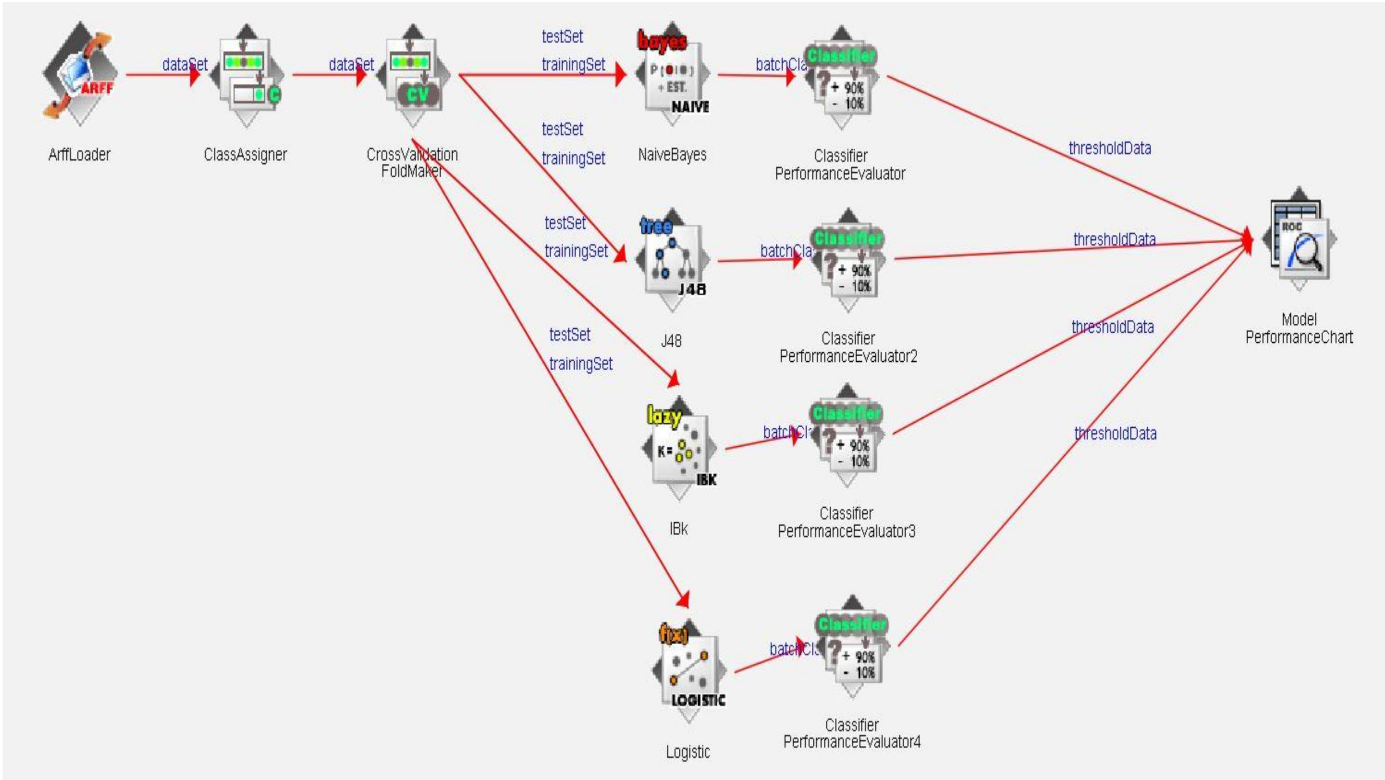

Fig 2.0 Knowledge Flow component Diagram

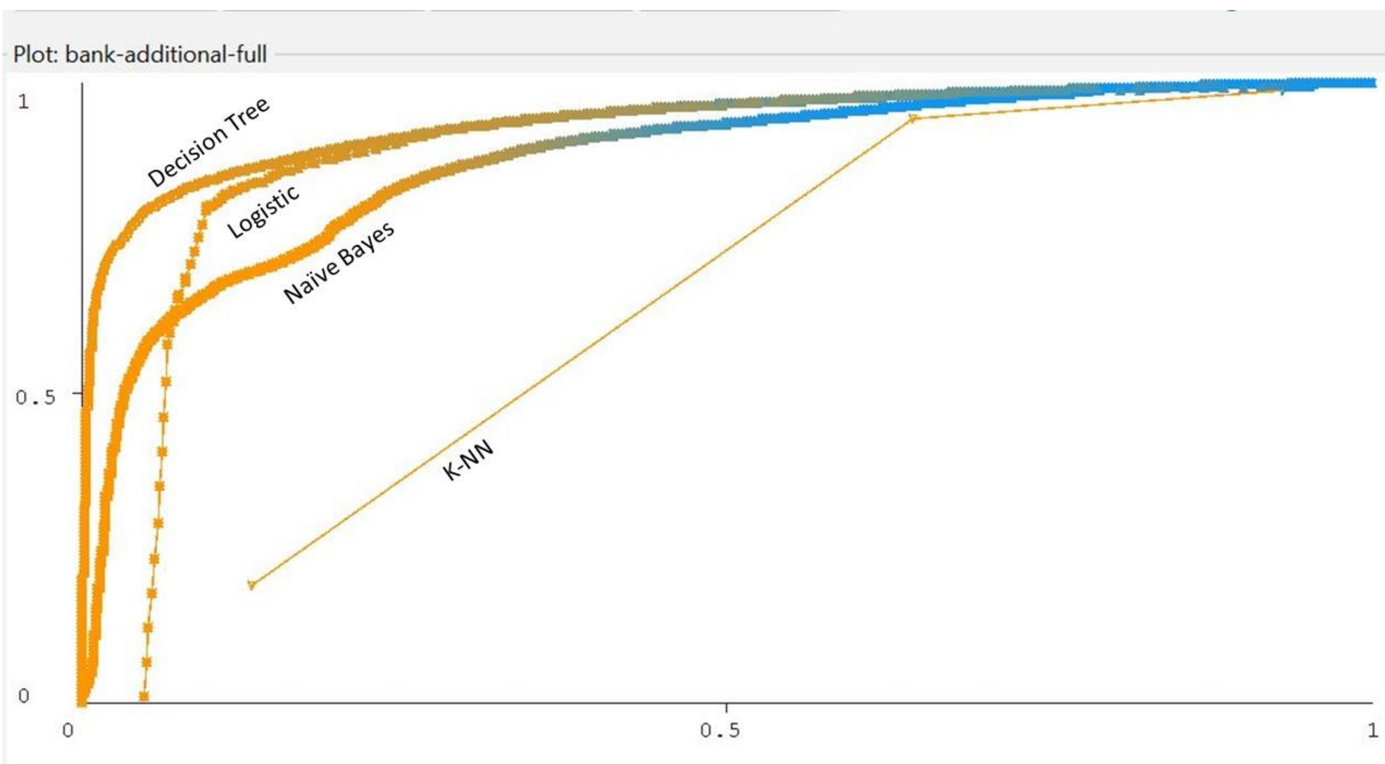

Fig 2.1 Receiver operating characteristic (ROC) curve

Conclusion:

Based on the analysis of the performance accuracy out of the four data models (Decision Tree, Naive Bayes, Logistic, KNN) used, we conclude that the Decision Tree has the highest accuracy percentage of 94% which would be a more efficient model to predict whether a customer will subscribe the bank's product or not.

Individual Contributions:

- *Data Acquisition -*
  Suman Pogul: Data research on various platforms like UCI Repository, Kaggle, and CERN. Ideal data set to meet the project goal. Minor modifications to make the data more efficient and useful. Tested sample dataset before finalizing.
  Swapnil Sahu: Data preprocessing and cleaning using MS-excel. Using various features to make the data organized and consistent. Performed some pre-checks to make sure data will show accurate results. Data Analysis using charts and graphs.
- *Data Modelling -*
  Tamilselvan Gurunathan: Analyzing ideal data models for the project. Tested different data models to check performance and accuracy. Based on the same, helped us to finalize the model for the project. Performed data loading and modeling in WEKA.
  Mounika Cheera: Performed model analysis and visualization. Analyzed various performance metrics via a knowledge base and ROC curves. Identified ideal models for the project and developed their performance analysis chart for more insight.
- *Documentation -*
  Smriti Khilnani: Performed the Project Progress Evaluation and Documentation. Identified the KPIs and roadblocks in the project. Managed resources online/offline to help other team members achieve their goals. Report completion and structuring.

References:

1. Data Repository: https://archive.ics.uci.edu/ml/datasets/bank+marketing
2. Foster Provost & Tom Facet, *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking, 1st Edition*
3. Pratyush Bharati, Abhijit Chaudhury, *An empirical investigation of decision-making satisfaction in web-based decision support systems,* Decision Support Systems, Volume 37, Issue 2, 2004, Pages 187-197, https://doi.org/10.1016/S0167-9236(03)00006-X.
   (https://www.sciencedirect.com/science/article/pii/S016792360300006X)