Bank Telemarketing Success Prediction

IE7300: Statistical Learning for Engineering

Spring 2023

Project Group 10

Narayana Sudheer Vishal Basutkar

Aryan Fernandes

Phani Bhavana Atluri

Manasvi Mittal

1. Abstract:

This project aims to develop predictive models for forecasting whether a client is likely to subscribe to a term deposit using the Bank Telemarketing dataset from UCI. The project also involves evaluating various classification models to determine the most efficient algorithm for predicting the results of telemarketing campaigns within the banking industry. By comparing and contrasting the performance of different models, the project seeks to establish the most effective approach for improving the efficiency of telemarketing campaigns. The dataset encompasses data collected from a Portuguese banking institution's telemarketing campaigns conducted between 2008 and 2013. The project evaluates the effectiveness of telemarketing as a marketing strategy for promoting banking services to clients during this period.

2. Introduction:

The banking industry has been using telemarketing campaigns to promote its services to clients for many years. With the advent of new technologies and increased competition in the industry, there is a need to evaluate the effectiveness of telemarketing as a marketing strategy. This project aims to leverage the Bank Telemarketing dataset from UCI to develop predictive models that can accurately forecast whether a client is likely to subscribe to a term deposit. The project also involves evaluating various classification models to determine the most efficient algorithm for predicting the results of telemarketing campaigns conducted by the banking institution.

The dataset encompasses data collected from a Portuguese banking institution's telemarketing campaigns conducted between 2008 and 2013, a period coinciding with the 2008 financial crisis. Since the banking industry underwent significant changes during this period, the project seeks to evaluate the effectiveness of telemarketing as a marketing strategy for promoting banking services to clients. By comparing and contrasting the performance of different models, the project seeks to establish the most effective approach for improving the efficiency of telemarketing campaigns within the banking industry.

The primary goal of this project is to gain a comprehensive understanding of the Bank Telemarketing dataset by carrying out data preprocessing, univariate and bivariate analysis, and determining the usefulness of the dataset columns. Furthermore, the project aims to test and compare three different classification models - Logistic Regression, Gaussian Naive Bayes, and Neural Networks - to identify the most effective algorithm for predicting the outcome of telemarketing campaigns conducted by the banking institution. By achieving these goals, the project seeks to improve our understanding of the effectiveness of telemarketing as a marketing strategy within the banking industry and provide insights for future telemarketing campaigns.

3. Dataset Description

The Bank Telemarketing dataset is available on the UCI Machine Learning Repository, accessible through the link https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. We chose the bank-additional-full dataset. (Direct Link). It contains 45211 records, consisting of 20 input variables and 1 output variable. The output variable is binary and serves as the target variable. Class 1 denotes that the client has subscribed to a term deposit, while class 0 indicates that the client has not subscribed. The input variables are a mix of numeric and categorical variables. However, the dataset exhibits a bias towards a higher number of records indicating that clients did not subscribe compared to those showing successful subscriptions. This imbalance in the data distribution could potentially impact the model's performance and may require addressing through techniques such as oversampling, undersampling, or synthetic data generation.

4. Exploratory Data Analysis

The dataset consists of 41188 rows and 21 columns. Upon analyzing the target variable, it was observed that the count for 'no' is almost 9 times more than the count of 'yes'. This indicates a higher likelihood of bias-variance tradeoff in the model's predictions. We dropped emp.var.rate because of its high correlation with nr_employed and euribor3m columns. We labeled encoding for the 'month' because there was an order and march was the month when there was a high probability of success. For the rest of all the categorical columns we did one hot encoding and for the rest of the numerical columns we did standard normalization.

The Bank Telemarketing dataset includes several variables related to both the bank client data and the last contact of the current marketing campaign.

The client data variables comprise of age, job type, marital status (divorced means divorced or widowed), education, default status, housing loan, and personal loan. On the other hand, the variables related to the last contact of the current marketing campaign include contact communication type (telephone or cellular), the last contact month of the year, the last contact day of the week, the last contact duration in seconds, and the number of days that passed (pdays) by after the client was last contacted from a previous campaign (numeric; 999 means the client was not previously contacted).

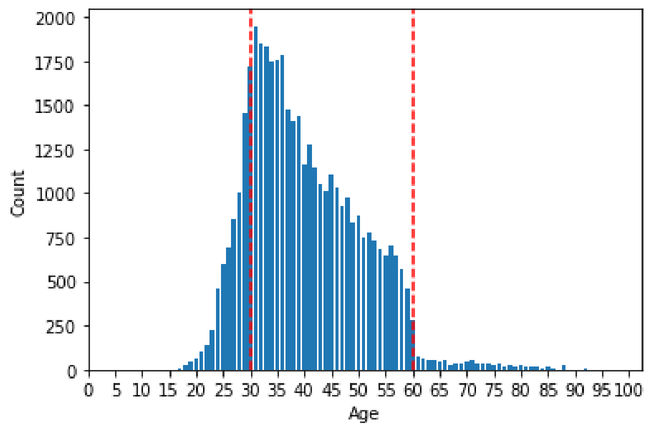
Additionally, the dataset includes the number of contacts performed before this campaign and for this client (numeric) and the outcome of the previous marketing campaign (categorical: 'failure', 'nonexistent', 'success'). Finally, the dataset also includes social and economic context attributes such as the employment variation rate - quarterly indicator (numeric) and the consumer price index - monthly indicator (numeric).

Output variable (desired target):

y - Has the client subscribed to a term deposit? Please provide a binary response: 'yes' or 'no'

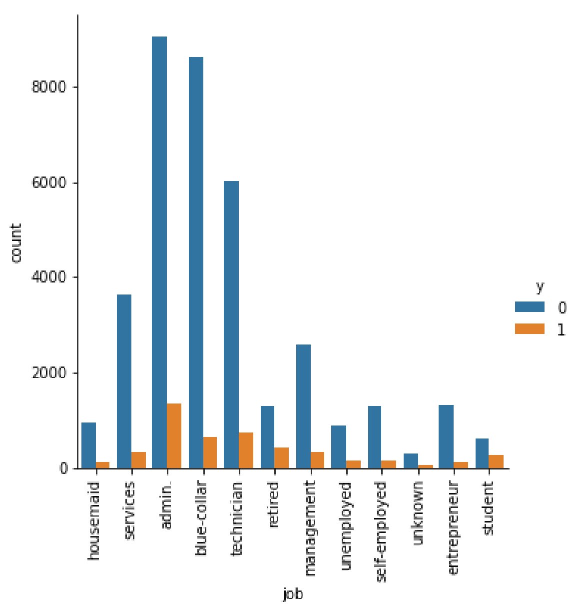
Column-wise data preprocessing

1) Age



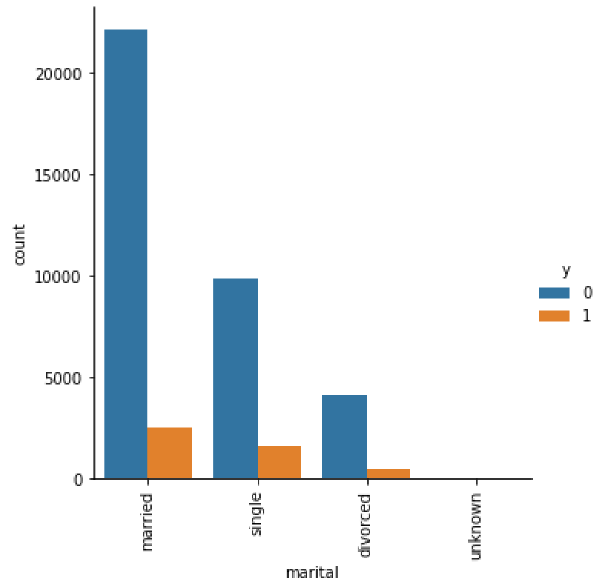
- We categorized the 'age' column to investigate the correlation between age and subscribing to a term deposit. This was done due to the bank targeting people between 30 and 60, but having low subscription rates among them. However, those over 60 tend to subscribe more, despite not being the primary target. Categorizing the column also helps to reduce the impact of outliers.

2) Job



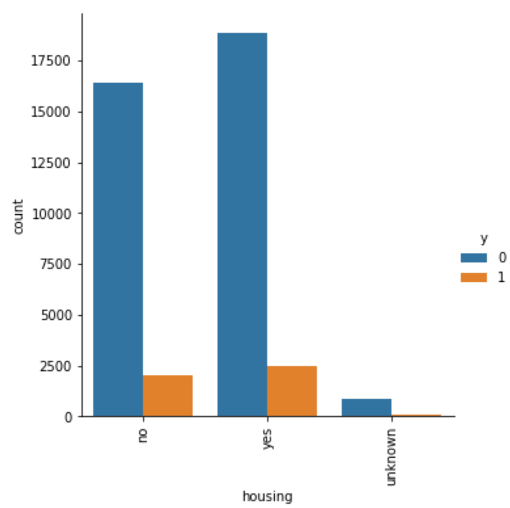
- The job feature is a categorical feature with very few unknown value fields. We dropped the entire rows that had unknown values.

3) Marital



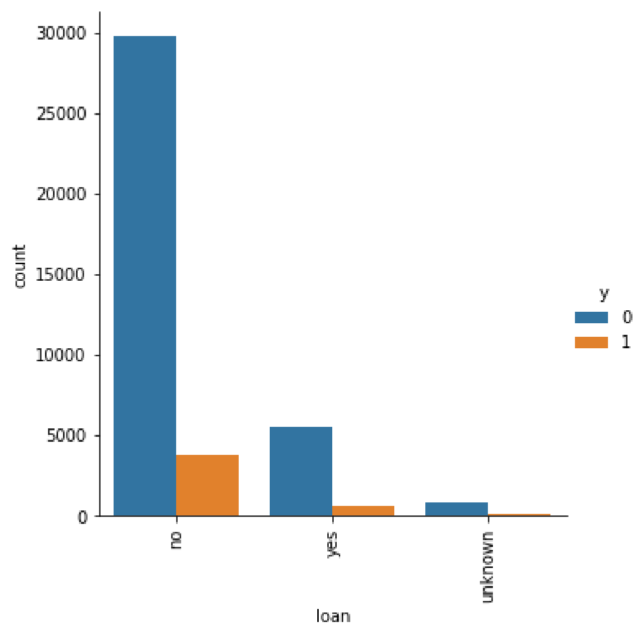
- Marital is also a categorical feature for which we again dropped the unknown value rows.

4) Housing



- After analyzing the 'housing' variable, it appears that having a housing loan may not have a significant impact on the likelihood of subscribing to a term deposit. This is based on the observation that the subscription rate for people without housing loans is approximately 10%, while the subscription rate for those with housing loans is 11%. Therefore, we have decided to drop this column from further analysis.

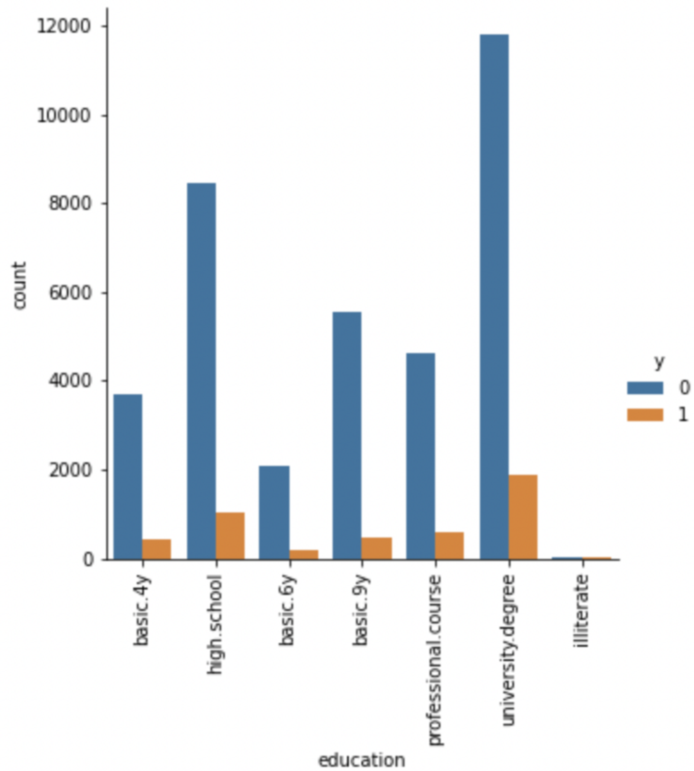
5) Loan



- Similar to housing, this feature doesn't make much significance so we dropped this as well.

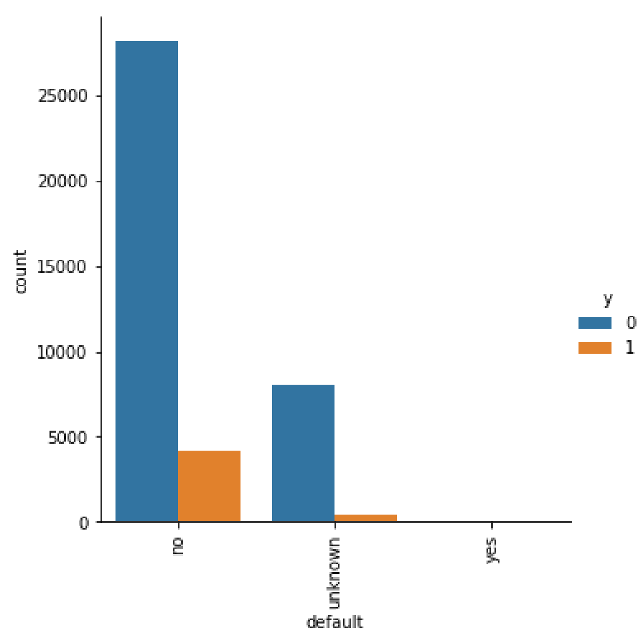
6) Education

- Moving on to the 'education' variable, we observed that there are a significant number of unknown values. However, the probability distribution of the known values is similar to that of the 'university degree' category, which also has the highest number of records. Therefore, we have imputed the unknown values in the 'education' variable with the 'university degree'.

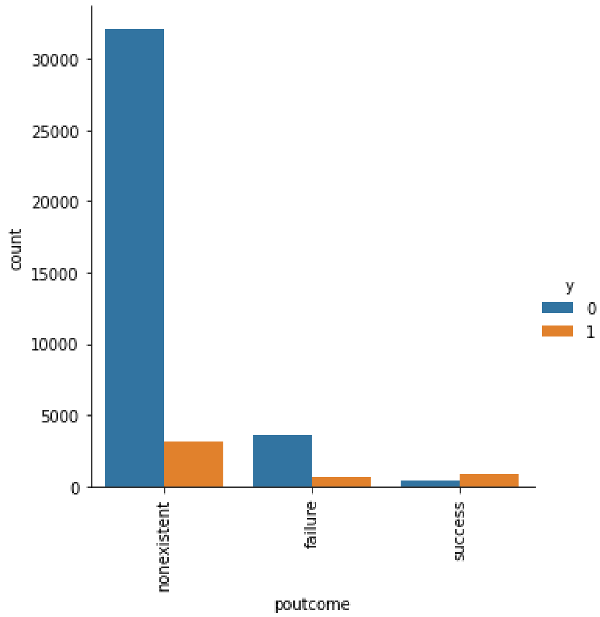


7) Default

- As a preprocessing step, the 'default' column was dropped due to a high number of unknown values.

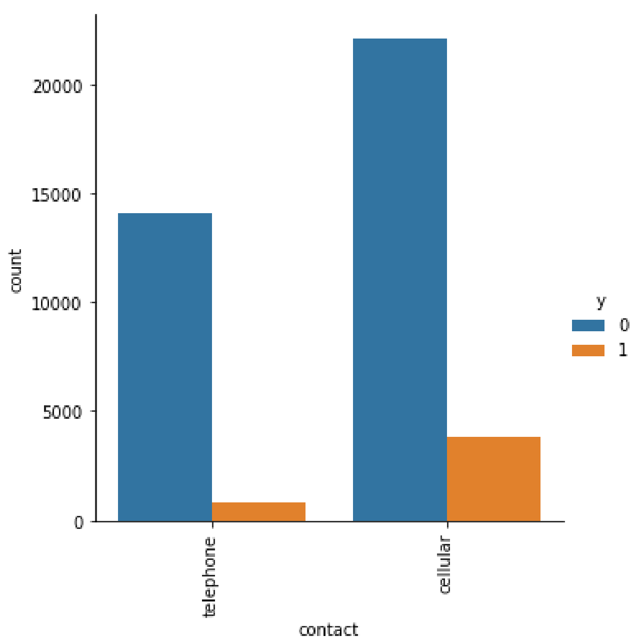


8) Poutcome



- We kept this feature as it is. (Later one-hot-encoded)

9) Contact



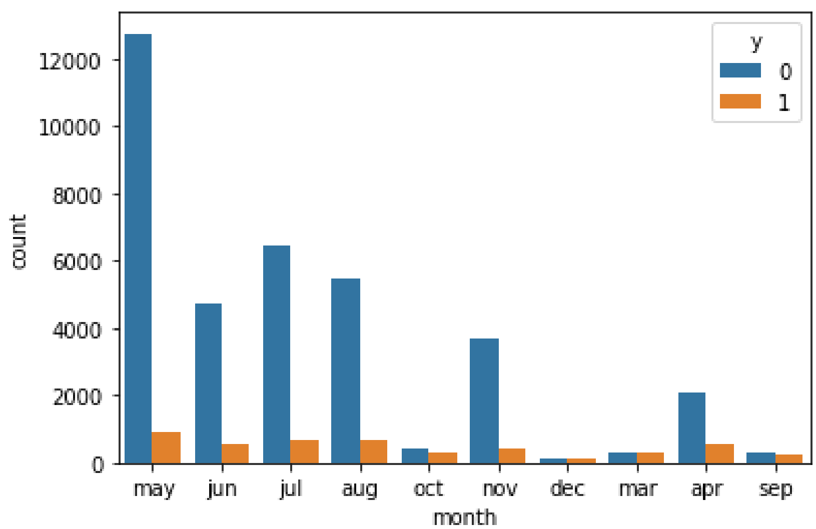
- Even this feature was untouched. (Later one-hot-encoded)

10) Previous

y	0	1
previous		
1_call	5	13
2+_calls	4085	1467
first_call	32103	3114

- Previous is a numeric feature. Since the number of calls after the second didn't matter much about the success, we categorized the numbers in the above format. We later one-hot encoded this feature.

11) Month



y	0	1
month		
apr	0.795038	0.204962
aug	0.894237	0.105763
dec	0.511111	0.488889
jul	0.909806	0.090194
jun	0.895520	0.104480
mar	0.493530	0.506470
may	0.935238	0.064762
nov	0.898948	0.101052
oct	0.560113	0.439887
sep	0.549822	0.450178

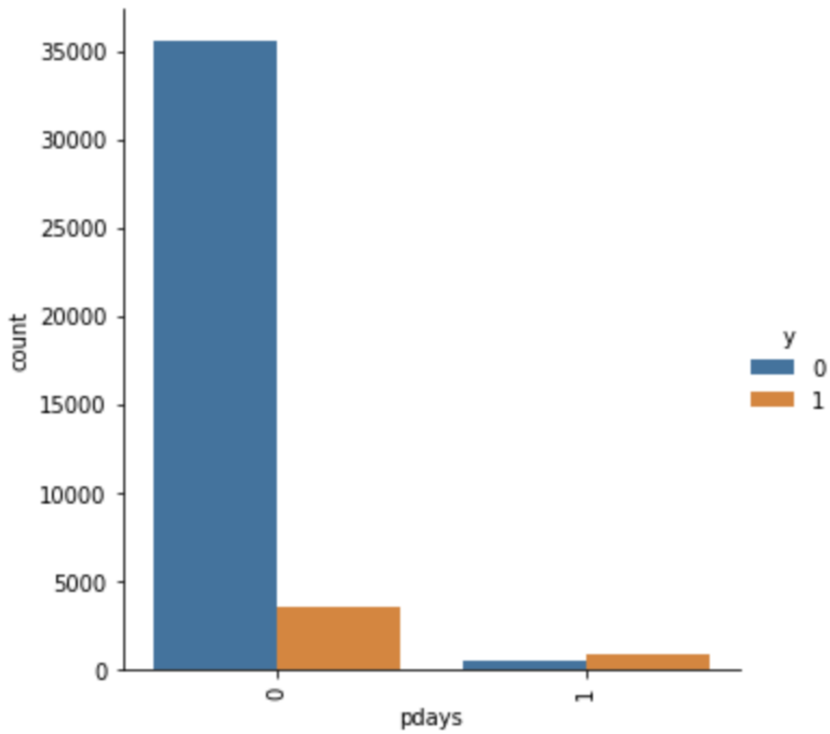
- We can see that there is a high chance of success in the month of march, then december, then september and so on.. Hence, instead of one hot encoding this feature, we label encoded it while giving more value to the months that gave more success probability and vice versa.

12) Day of week

```
y               0     1
day_of_week
fri          6940   839
mon          7579   841
thu          7497  1033
tue          7060   946
wed          7117   935
y                   0         1
day_of_week
fri          0.892146  0.107854
mon          0.900119  0.099881
thu          0.878898  0.121102
tue          0.881839  0.118161
wed          0.883880  0.116120
```

- Upon analyzing the 'day of week' variable, we observed that the distribution of values is quite even across all days of the week. This suggests that the day of the week may not have a significant impact on the likelihood of subscribing to a term deposit. Therefore, we have decided to drop this column from further analysis.

13) Pday

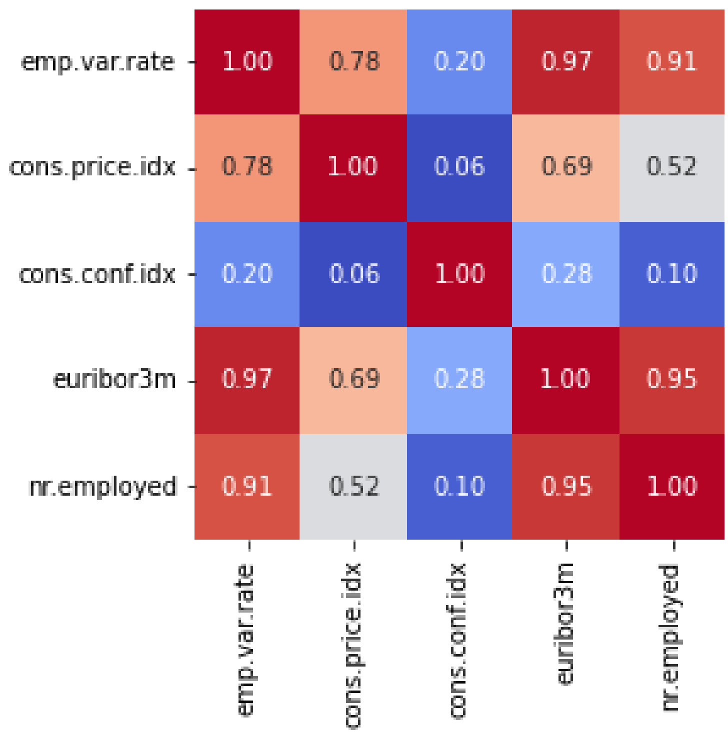


- If clients weren't contacted previously, i.e. their value was 999. We renamed them to 0 and rest of those that were contacted before, we cumulated them to one value - 1.

14) Duration and Campaign

- These features were also untouched and in the preprocessing.

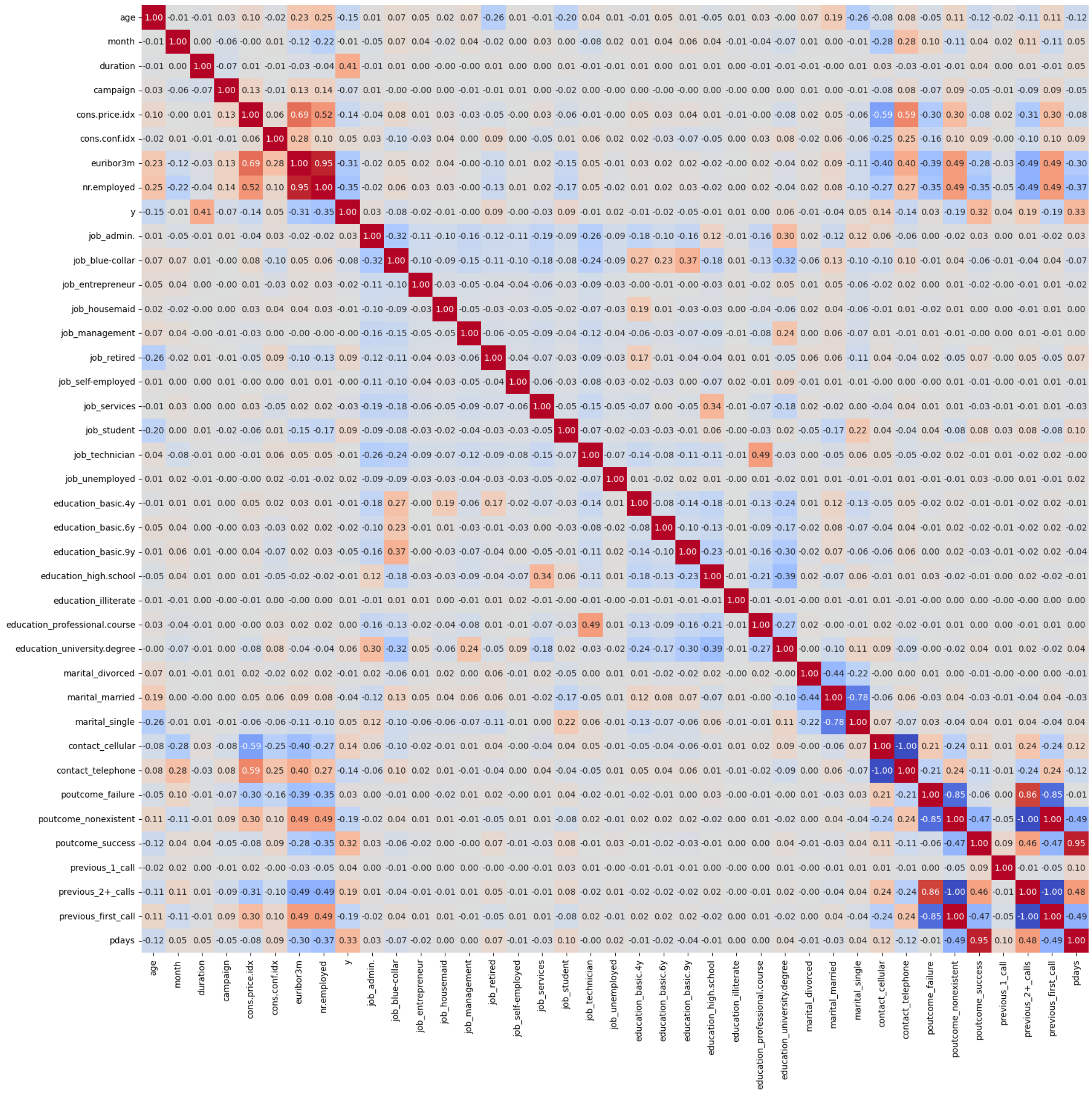
Bivariate analysis of socio-economic features



- As we can see, the emp.var.rate and euribor3m and nr.employed are highly correlated. We dropped the emp.var.rate feature.

Once the column-wise analysis was done, we performed one hot encoding on the categorical features and standardization on the numerical features.

Finally, we created the correlation matrix of the entire dataframe to see the correlations of all columns with each other. We found that 'poutcome_nonexistent' and 'previous_first_call' had a correlation coefficient of 1, which makes sense as if it is the first call, the previous outcome will be non-existent. We removed the poutcome_nonexistent from the dataframe. Below is the image of the correlation matrix.



5. Machine Learning Models

Test/Train Split

The code first splits the original dataframe ‘df_final’ into two datasets - a training set (train_df) and a test set (test_df). This is done using the train_test_split() function from the scikit-learn library, which randomly splits the data into two sets based on a specified test size (in this case, 0.2 or 20% of the data).

We performed the SMOTE function on the training dataset because of the imbalance in the dataset. We resampled the success-indicating rows to a size that is 40% of the failure.

1) Logistic Regression

We performed Logistic regression with the L2 Regularization and Gradient Descent algorithm from scratch and tested the model for precision, recall, f1_score, and accuracy. We tuned the learning rate hyperparameter by running over a loop of learning rates and picked the best one.

We also compared our performance metrics with the metrics achieved from applying logistic regression using sklearn library and we got very similar results. The results are tabulated in the results section.

2) Gaussian Naive Bayes

We trained our own Gaussian Naive Bayes model which we have written from scratch with the final preprocessed dataset as the columns follow Gaussian distribution.

We also added the Laplace smoothing method as the Naive Bayes classifier will be more robust to outliers and small sample sizes, and provide more accurate predictions on test data. Correspondingly, we have seen an increase in accuracy when the Laplace smoothing method is added.

3) Decision Tree

We also tested our dataset against SKLearn’s Decision Tree classifier as well. This classifier gave us the worst results compared to our other models.

4) Neural Network

We created a Deep Neural Network using Tensorflow and added L2 regularization and Batch Normalization after each layer to eradicate the vanishing gradient problem. Below is the summary of the model-

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 dense (Dense)               (None, 1024)              39936     
                                                                 
 dense_1 (Dense)             (None, 256)               262400    
                                                                 
 batch_normalization (BatchN  (None, 256)              1024      
 ormalization)                                                   
                                                                 
 dense_2 (Dense)             (None, 64)                16448     
                                                                 
 batch_normalization_1 (Batc  (None, 64)               256       
 hNormalization)                                                 
                                                                 
 dense_3 (Dense)             (None, 32)                2080      
                                                                 
 batch_normalization_2 (Batc  (None, 32)               128       
 hNormalization)                                                 
                                                                 
 dense_4 (Dense)             (None, 1)                 33        
                                                                 
=================================================================
Total params: 322,305
Trainable params: 321,601
Non-trainable params: 704
_________________________________________________________________
```

6. Results

The results of the above-mentioned ML models are tabulated below.

S.no.	Model	Training Accuracy	Test Accuracy	Precision	Recall	F1 Score
1	Logistic Regression	86.15%	90.46%	56.01%	71.49%	62.81%
2	Logistic Regression Sklearn	86%	89.9%	71.3%	54.11%	62.01%
3	Gaussian Naive Bayes	77.29%	89.56%	57%	32%	41%
4	Decision Tree Sklearn	75%	80.9%	34.7%	79.1%	48.2%
5	Deep Neural Network	91%	91.38%	59.48%	57.6%	55.6%

7. Conclusion

In conclusion, this project aimed to explore the Bank Telemarketing dataset and develop predictive models to determine the most efficient algorithm for forecasting whether a client is likely to subscribe to a term deposit. The dataset contained 45211 records with 20 input variables and 1 output variable. The project used three different classification models - Logistic Regression, Gaussian Naive Bayes, and Neural Networks - to identify the most effective algorithm for predicting the outcome of telemarketing campaigns conducted by the banking institution. The project also carried out data preprocessing, univariate and bivariate analysis to gain a comprehensive understanding of the dataset.

The project observed that the dataset exhibited a bias toward a higher number of records indicating that clients did not subscribe compared to those showing successful subscriptions. This imbalance in the data distribution could potentially impact the model's performance and may require addressing through techniques such as oversampling, undersampling, or synthetic data generation.

Overall, the project identified that Logistic Regression was the most efficient algorithm for forecasting the outcome of telemarketing campaigns conducted by the banking institution. The insights gained from this project can help to improve the efficiency of telemarketing campaigns within the banking industry and provide valuable insights for future telemarketing campaigns.