ITM618 Final Course Project

Paul Nguyen, Faraz Ali, Aidan Ranjitsingh, Nehal Patel

Project Objective

- Determine if a client would subscribe to a term deposit or not by implementing an algorithm on the target class “Subscribed”.
- Explored the shape and size of the training dataset.
- Cleaned the values of the training set and used it on the test set.
- Implemented classification models to visualize data.
- Applied classification models to get enhanced results.

Data Exploration

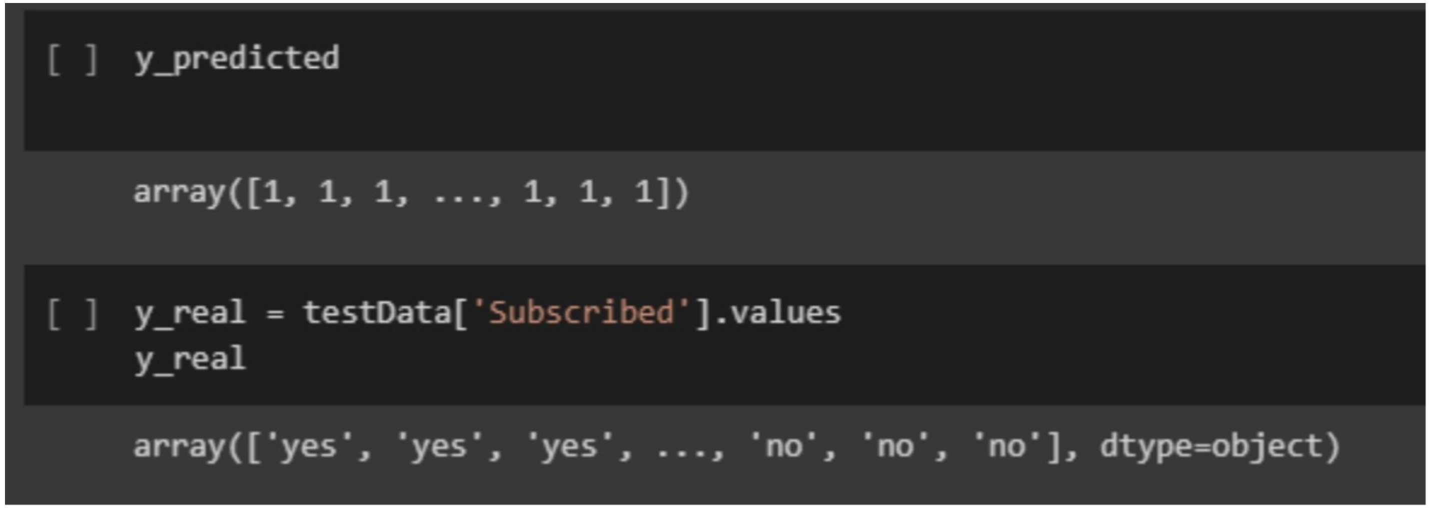
- There were 14 attributes + 1 target attribute (Subscribed):
  - 5 numeric (age, duration, campaign, pdays, nr.employed).
  - 6 nominal.
    - E.g., job, marital, education, housing, loan, contact,
  - 3 ordinal
    - E.g., month, day_of_week, poutcome.
- Number of elements present in the dataset was 439,065 items.
- The dimensions of the dataset represented in the form of a tuple was (29271, 15).

Data Cleaning

- Found partial, noisy, and duplicate data in the training set.
- Implemented a df.dropna function on the flawed data:
  - Identified 2,964 unknown values in the train set.
  - Identified 1,157 unknown values in the test set.
- Dropped rows with errors in:
  - “job”
  - “marital”
  - “education”
  - “housing”
  - “loan”

Learning Method 1

- Implemented KNN model.
  - Helped to predict if most of the target attribute was subscribed or not.
  - The use of y_predicted and y_real functions helped to detail the number of subscriptions.



```
[ ]  y_predicted

     array([1, 1, 1, ..., 1, 1, 1])

[ ]  y_real = testData['Subscribed'].values
     y_real

     array(['yes', 'yes', 'yes', ..., 'no', 'no', 'no'], dtype=object)
```

Learning Method 2

- Implemented decision tree.
  - Gave a visualization of how the prediction algorithm worked.
  - It helped to map out all the attributes and visualized the outcome of each attribute.
  - Provided a generalization of whether a client will subscribe or not.

Evaluation

- The following are a representation of values comparing the test results to the predicted results using the confusion matrix.

	p	n
Y	282	146
N	158	2990

- **Accuracy**
  - **91.45%**
- **Error rate**
  - **8.5%**
- **Precision**
  - **65.89%**
- **Recall**
  - **64.1%**

Evaluation

- The accuracy rate of 91.45% indicated the correctness of the test predictions to the test results.
- The error rate of 8.5% is an inverse of the accuracy rate demonstrating the errors the prediction model had compared to the test results.
- The precision rate of 65.89% represents the positivity of the values that were labelled as actually positive.
- The recall rate of 64.1% signifies that the values that were selected as positive were classified as positive.

Discussions

- What did you find about your models?
  - The KNN model was more effective than the decision tree as it generated the same results with less resources.
  - The decision tree helped to visualize how predictions were made.
  - The use of a confusion matrix was inefficient with a decision tree.

Conclusion and Future Work

- Summary
  - Both methods drew similar prediction accuracies. The decision tree gave a visual representation whereas KNN put all the data into a readable array.
- What would you do in future to improve the models?
  - We would reduce redundancy in the data. Removing this data would allow for a more accurate result based off the actual data.