**Part 2 Classification on Bank Marketing Dataset**

**Task:** Classify whether it is a yes / no on client’s access to a product using Bank Marketing Dataset

**Dependencies:**

Same as Part 1.

**Dataset per-processing:**

Unlike AirQuality dataset, whose features all are in number format, Bank Dataset contains a mixed of number and string data. So in order for classification to work, we need to somehow transform this dataset to contain only number.

Fortunately, Pandas has built-in function that allows us to do just that (pd.get\_dummies(df)). This will categorize all string data into forms of binary data. For example:

Marital status column has ‘Married’, ‘Single’, and ‘Divorced’. When we transform this column we got Marital\_married, Marital\_single, Marital\_divorced. If the original data is Marital:Married then it will convert to Marital\_married = 1, Marital\_single = 0, Marital\_divorced = 0.

For example,

|  |
| --- |
| **Marital** |
| Marriend |
| Single |
| Divorced |

The above data will become,

|  |  |  |
| --- | --- | --- |
| Marital\_married | Marital\_Single | Marital\_Divorced |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

After finishing pre-processing the data, we can start fitting the data using functions from sklearn.

**Result Analysis**

**Logistic Regression:**

precision recall f1-score support

0 0.92 0.97 0.94 7980

1 0.63 0.35 0.45 1063

avg / total 0.88 0.90 0.89 9043

**Support Vector Machine:**

precision recall f1-score support

0 0.92 0.96 0.94 7980

1 0.57 0.41 0.48 1063

avg / total 0.88 0.89 0.89 9043

**Decision Tree:**

precision recall f1-score support

0 0.93 0.92 0.93 7980

1 0.45 0.47 0.46 1063

avg / total 0.87 0.87 0.87 9043

**Random Forest:**

precision recall f1-score support

0 0.92 0.97 0.94 7980

1 0.61 0.34 0.43 1063

avg / total 0.88 0.90 0.88 9043

**K-NN:**

precision recall f1-score support

0 0.92 0.97 0.94 7980

1 0.60 0.35 0.44 1063

avg / total 0.88 0.90 0.88 9043

|  |  |  |
| --- | --- | --- |
| Algorithm | Accuracy | Execution Time |
| Logistic Regression | **0.89936968** | 0.82156301 |
| Support Vector Machine | 0.89372996 | 5.17533803 |
| Decision Tree | 0.87139224 | **0.29571080** |
| Random Forest | 0.89682627 | 0.31405210 |
| K-NN | 0.89594161 | 0.30589700 |

**-** It seems that accuracy for 0(No) is much better than 1(Yes) across all algorithms. And it is because of the number of No dataset is a lot more than Yes dataset.

Yes: 5289 rows ; No:39922 rows

- Logistic regression seems to fit this dataset the best.

- Decision tree has lowest execution time while also has the lowest accuracy which lets use understand between the accuracy vs algorithm complexity.

- Random forest has the best algorithm complexity to accuracy ratio for this dataset.

- Overall, SVM is the most expensive algorithm to execute as it has the biggest execution time.