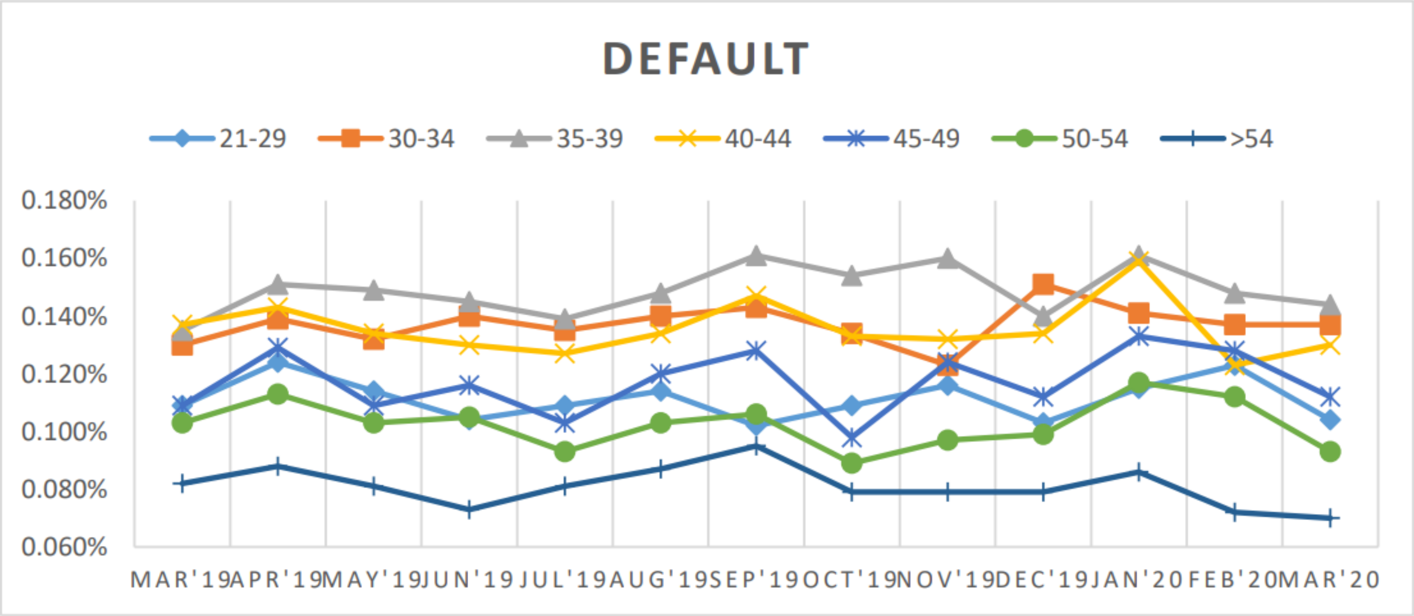
Data Science Capstone Project

1.1 Current Situation

A Report by Credit Bureau Singapore (CBS) for Q1 2020 shows that on average, there are 0.1% of credit card holders in Singapore defaulting on their payment.

(extracted from https://www.creditbureau.com.sg/pdf/CBS-Consumer-Credit-Index-(CCI)-Q1-2020.pdf)



MONTHLY AVERAGE	
21- 29	0.11%
30- 34	0.14%
35- 39	0.15%
40- 44	0.14%
45- 49	0.12%
50- 54	0.10%
>54	0.08%

1.2 Problem Statement

Default is a serious credit card status. It affects defaulter's :-

- standing with that credit card issuer.
- ability to get approved for other credit-based services.

Default loans also cost a lot on a bank due to :-

- Increases the Cost of Funds
- Decreases the Profitability of the Bank
- Decreases the Overall Credit Rating of the Bank

2.1 Data Source

The dataset is from a bank in Taiwan dated Sep 2005.

(extracted from https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset)

- There are 25 variables.
- There are 30,000 customers' data.
- All are numerical data.

```
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 25 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          30000 non-null  int64
 1   LIMIT_BAL                   30000 non-null  float64
 2   SEX                         30000 non-null  int64
 3   EDUCATION                   30000 non-null  int64
 4   MARRIAGE                    30000 non-null  int64
 5   AGE                         30000 non-null  int64
 6   PAY_0                       30000 non-null  int64
 7   PAY_2                       30000 non-null  int64
 8   PAY_3                       30000 non-null  int64
 9   PAY_4                       30000 non-null  int64
 10  PAY_5                       30000 non-null  int64
 11  PAY_6                       30000 non-null  int64
 12  BILL_AMT1                   30000 non-null  float64
 13  BILL_AMT2                   30000 non-null  float64
 14  BILL_AMT3                   30000 non-null  float64
 15  BILL_AMT4                   30000 non-null  float64
 16  BILL_AMT5                   30000 non-null  float64
 17  BILL_AMT6                   30000 non-null  float64
 18  PAY_AMT1                    30000 non-null  float64
 19  PAY_AMT2                    30000 non-null  float64
 20  PAY_AMT3                    30000 non-null  float64
 21  PAY_AMT4                    30000 non-null  float64
 22  PAY_AMT5                    30000 non-null  float64
 23  PAY_AMT6                    30000 non-null  float64
 24  default.payment.next.month  30000 non-null  int64
```

2.2 Data Cleaning

1. Rename and Removing columns:-

- *PAY_0* -> *PAY_1*. *default.payment.next.month* -> *POSSIBLE_DEFAULT*
- Fields for PAY_AMT and BILL_AMT are dropped.

2. Handling Missing Data:-

- There are no missing data

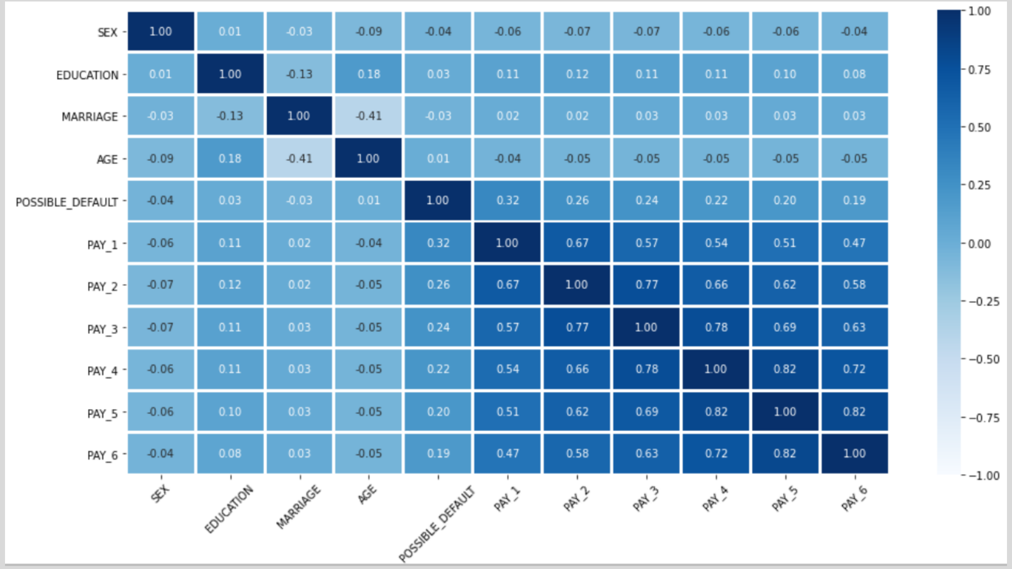
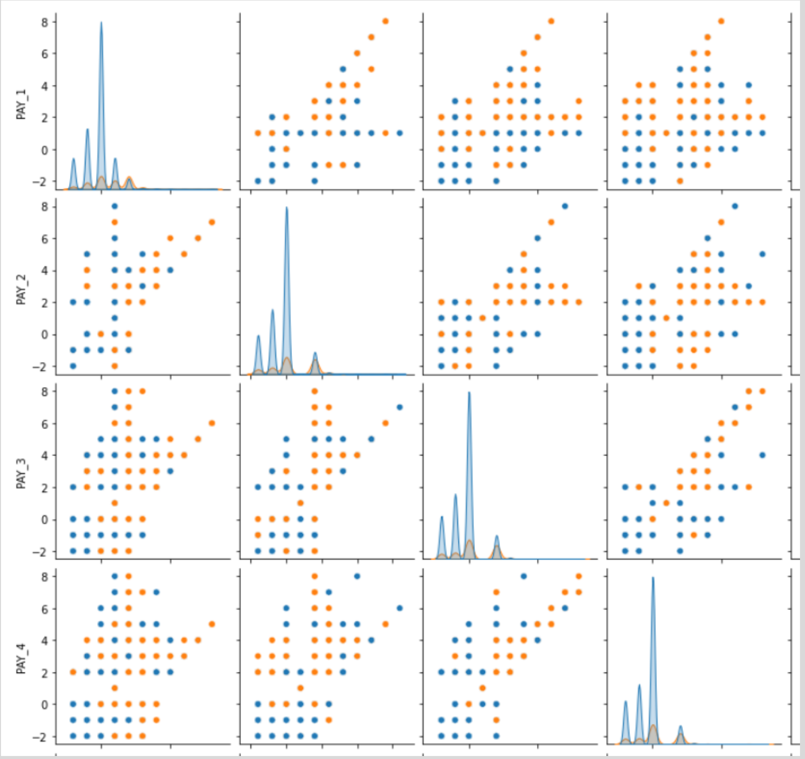
3. Remove/Replace Outliers

- Fields *EDUCATION* and *MARRIAGE* have data out of the nominal range

4. Check Duplicates

- There are no duplicates from original data set.

Correlationships between Fields

Heatmap	Pairplots
	

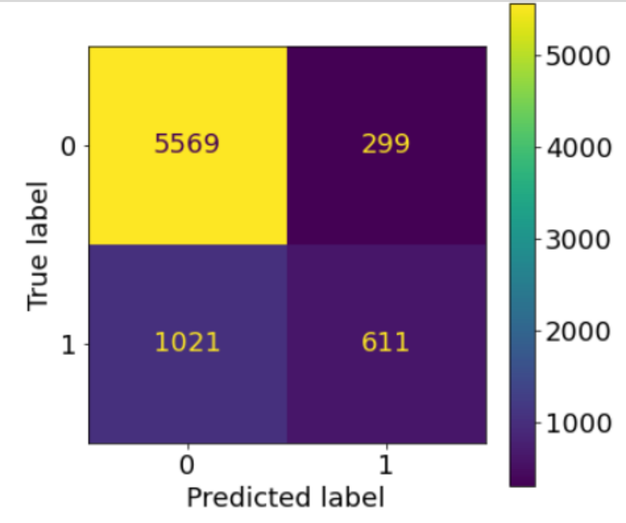
3. Data Mining & Data Analytics

1. Training (75%) and Test (25%) sets
2. Three predictive models are used.
    - Logistic Regression
    - Decision Tree
    - K-Nearest-Neighbors
3. Nine criterion for comparison.

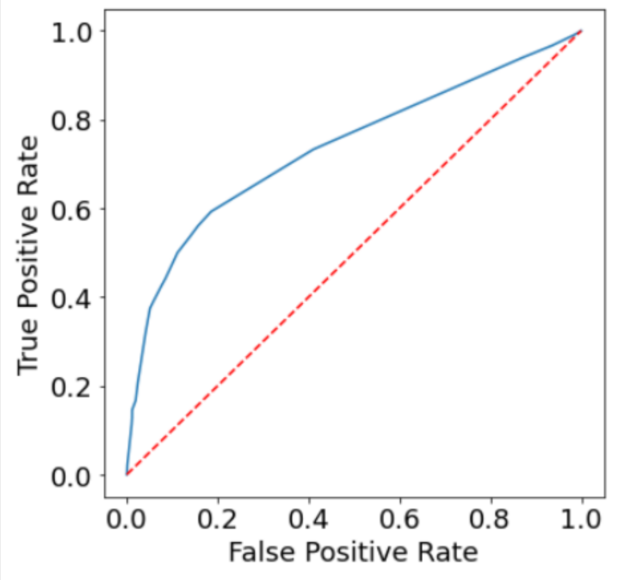
    Training Accuracy, Testing Accuracy, Precision(0), Recall(0), F1(0), Precision(1), Recall(1), F1(1), ROC.

3.1 Generate Criterion for Comparison

Confusion Matrix



ROC



Table

	LR	DT	KNN	GS	Best Score
Train Acc	0.805556	0.830800	0.819511	0.0	DT
Test Acc	0.811467	0.822000	0.824000	0.0	KNN
Precision(0)	0.815887	0.839856	0.845068	0.0	KNN
Recall(0)	0.980232	0.954499	0.949046	0.0	LR
F1(0)	0.890540	0.893515	0.894044	0.0	KNN
Precision(1)	0.742222	0.678700	0.671429	0.0	LR
Recall(1)	0.204657	0.345588	0.374387	0.0	KNN
F1(1)	0.320845	0.457978	0.480724	0.0	KNN
ROC	0.713846	0.730133	0.736687	0.0	KNN

Selected model :- K-Nearest-Neighbors

3.2 Tuning Hyperparameter

- GridSearchCV to tune KNN with the following parameters :-
  - *n_neighbors*: list(range(20,25)),
  - *weights*: ('uniform', 'distance’),
  - *algorithm*: ('ball_tree', 'kd_tree’)
- Optimised KNN fared better in Training Accuracy.
- Original KNN fared better in most other criterion.
- Original KNN is the preferred model.

	KNN	GS	Best Score
Train Acc	0.819511	0.820222	GS
Test Acc	0.824000	0.822400	KNN
Precision(0)	0.845068	0.842081	KNN
Recall(0)	0.949046	0.951431	GS
F1(0)	0.894044	0.893423	KNN
Precision(1)	0.671429	0.672414	GS
Recall(1)	0.374387	0.358456	KNN
F1(1)	0.480724	0.467626	KNN
ROC	0.736687	0.736687	KNN

3.3 Interpretation of Results

1. *Train Acc* -> The model is able to correctly classified 82% of the training data.
2. *Test Acc* -> The model is able to correctly classified 82% of the testing data.
3. *Precision(0)* -> For all the predicted POSSIBLE_DEFAULT=0, the model is able to correctly predict 85% of them
4. *Recall(0)* -> For all the outcome POSSIBLE_DEFAULT=0, the model is able to correctly predict 95% of them
5. *F1(0)* -> Score of 89% shows that the model is able strike good balance between Sensitivity and Precision for POSSIBLE_DEFAULT=0
6. *Precision(1)* -> For all the predicted POSSIBLE_DEFAULT=1, the model is able to correctly predict 67% of them
7. *Recall(1)* -> For all the outcome POSSIBLE_DEFAULT=1, the model is able to correctly predict 37% of them
8. *F1(1)* -> Score of 48% shows that the model is average in balance between Sensitivity and Precision for POSSIBLE_DEFAULT=1
9. *ROC* -> The score of 73% shows that the model is fairly good at classifying POSSIBLE_DEFAULT=0 and POSSIBLE_DEFAULT=1 correctly

4. Conclusion

- 3 models (KNN, LR, DT) are used to predict whether a customer will default on his next credit card payment.
- Heatmaps and charts are used to get a visual feel of how the fields are correlated to each other.
- Nine criterions are used to evaluate which model is more suitable. (Training Accuracy, Testing Accuracy, Precision(0), Recall(0), F1(0), Precision(1), Recall(1), F1(1), ROC)
- KNN has been found to be a more suitable for this dataset.
- Logistic Regression and Decision Tree was also implemented but both did not matched up to KNN.
- KNN with optimised hyperparameter performs better for accuracy of the training set but is not as good for most criterions generated from the test set. This could be due to overfitting.

THE END