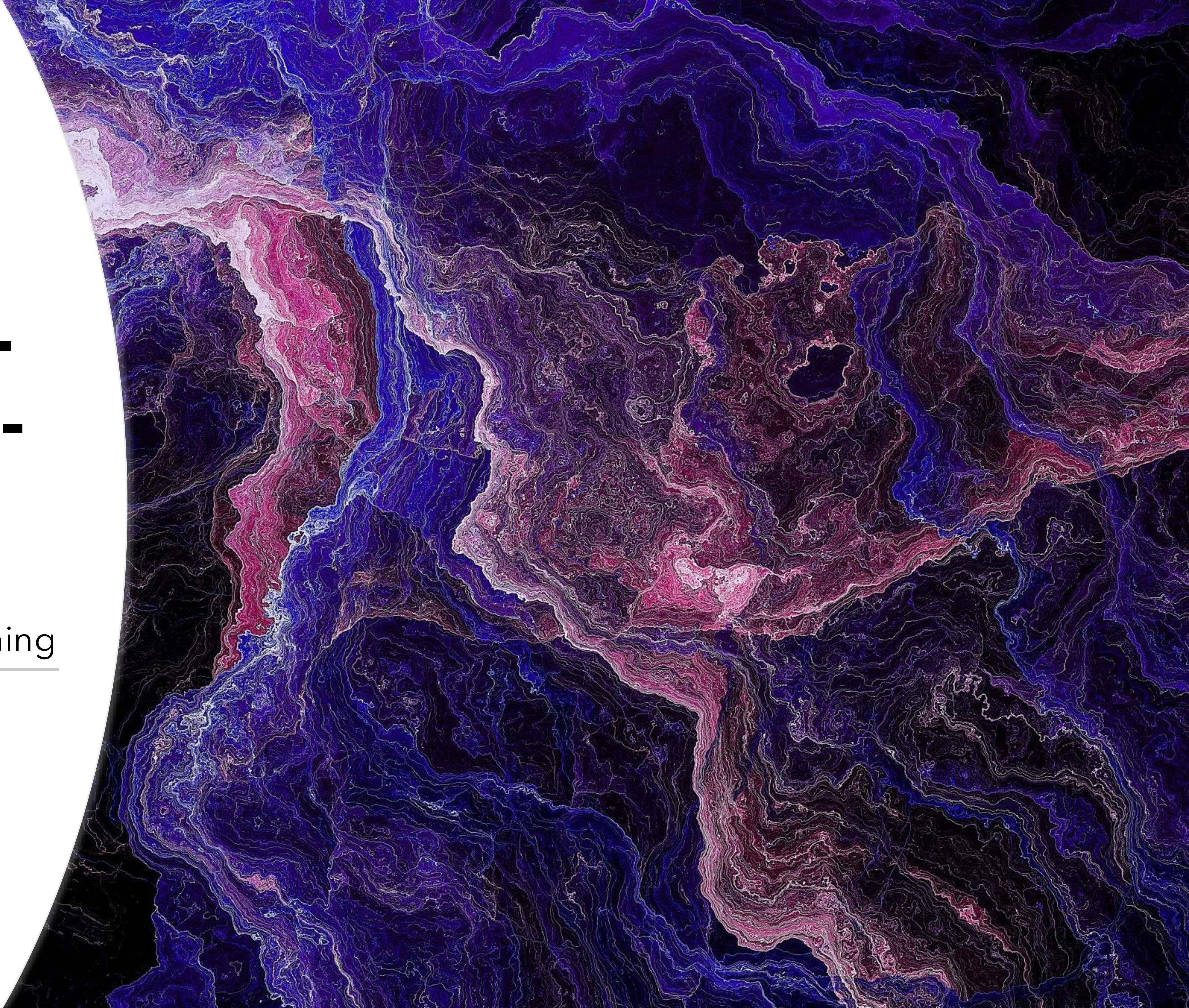
Evaluating the CO2 Emission from Gasoline-Powered Light-Duty Vehicles

Group 5 – Applied Machine Learning

Han Chau - 20012654

Randy Duong - 20019147

Nihar Dugade - 20018679



Content

- Problem and research question
- Data description and insights
- Data pre-processing
- Method
- Analysis and results
- Conclusion

Problem and research question

Can we accurately predict whether the CO_2 emissions of a car exceed the allowed threshold using machine learning models?

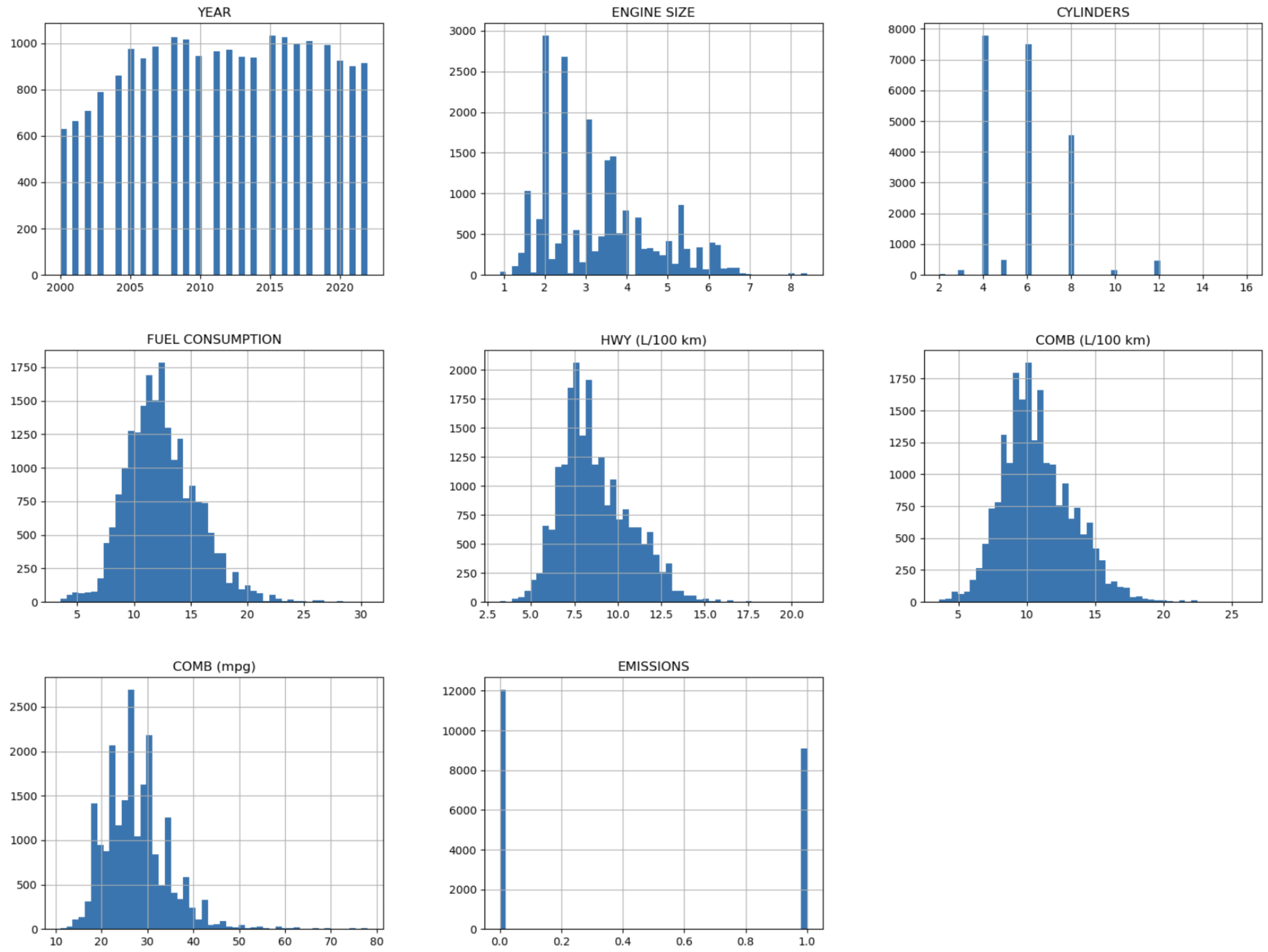
Can the chosen model be used as a practical tool for monitoring and controlling CO_2 emissions in real-time scenarios?

Data description and insights

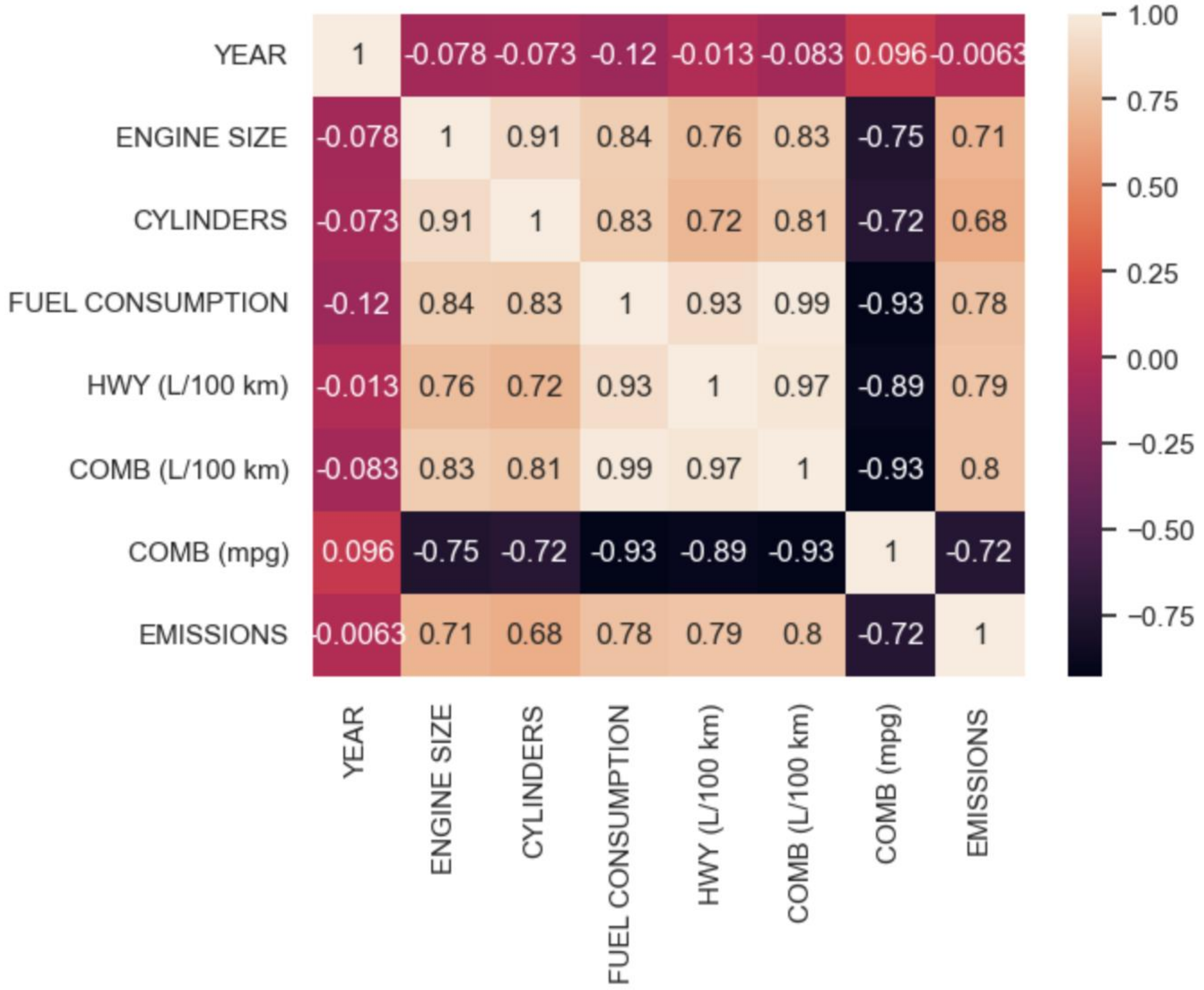
- Taken from Kagle with 22,556 examples and 13 columns.
- Datasets provide model-specific fuel consumption ratings and estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada.

	YEAR	MAKE	MODEL	VEHICLE CLASS	ENGINE SIZE	CYLINDERS	TRANSMISSION	FUEL	FUEL CONSUMPTION	HWY (L/100 km)	COMB (L/100 km)	COMB (mpg)	EMISSIONS
0	2000	ACURA	1.6EL	COMPACT	1.6	4	A4	X	9.2	6.7	8.1	35	186
1	2000	ACURA	1.6EL	COMPACT	1.6	4	M5	X	8.5	6.5	7.6	37	175
2	2000	ACURA	3.2TL	MID-SIZE	3.2	6	AS5	Z	12.2	7.4	10.0	28	230
3	2000	ACURA	3.5RL	MID-SIZE	3.5	6	A4	Z	13.4	9.2	11.5	25	264
4	2000	ACURA	INTEGRA	SUBCOMPACT	1.8	4	A4	X	10.0	7.0	8.6	33	198

Data description and insights



Data description and insights



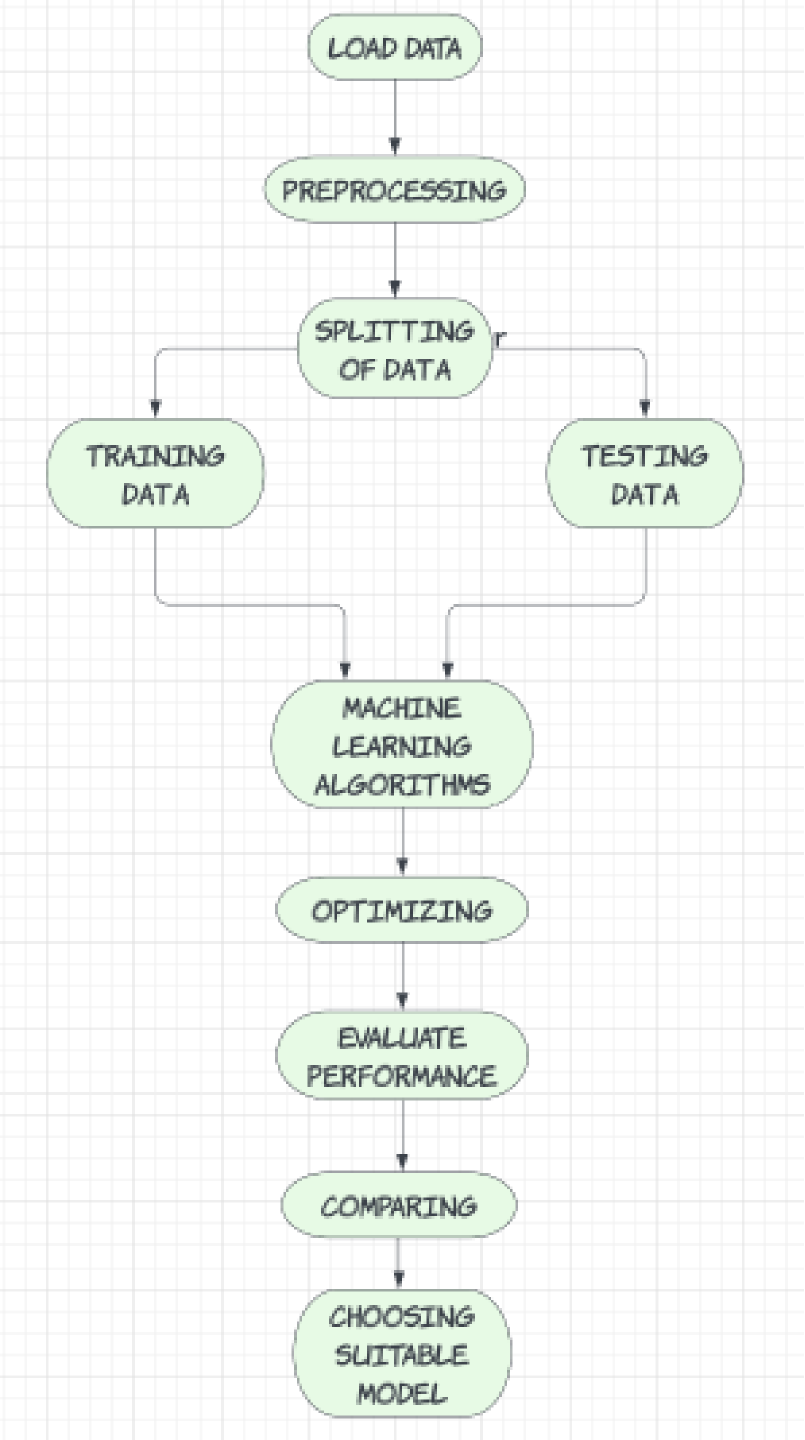
Data pre-processing

- Drop examples which the types of fuel is not Gasoline.
- Transform the emission columns into binary values (1 and 0) depending on whether they exceed the threshold.

$$CO_2 \text{ emissions per km} = \frac{CO_2 \text{ per gallon}}{MPG} = \frac{8,887}{35.4} = 251 \text{ grams}$$

- Eliminate outliers by considering the interquartile range (IQR) of the fuel consumption variable.

METHOD



LOAD DATA
PREPROCESSING
SPLITTING OF DATA
TRAINING DATA
TESTING DATA
MACHINE LEARNING ALGORITHMS
OPTIMIZING
EVALUATE PERFORMANCE
COMPARING
CHOOSING SUITABLE MODEL

Machine Learning Algorithms

- Logistic Regression
- Linear SVC
- Random Forest
- Neural Network
- Naïve Bayes

Logistic regression model

Set default for the initial parameters.

```
The Acuracy is : 97.44 %
Classification report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1217
           1       0.97      0.97      0.97       856

    accuracy                           0.97      2073
   macro avg       0.97      0.97      0.97      2073
weighted avg       0.97      0.97      0.97      2073
```

Logistic regression model

Optimizing the model with GridSearchCV

```
lr_model = LogisticRegression()

# Define hyperparameters to search over
hyperparameters = {
    'penalty': ['l1', 'l2',None],
    'C': [0.01, 0.1, 1, 10, 100,1000]
}

# Use GridSearchCV to find best hyperparameters
grid_search = GridSearchCV(lr_model, hyperparameters, cv=5)
grid_search.fit(train_x, train_y)
```

```
Best hyperparameters:  {'C': 0.01, 'penalty': None}
Best accuracy score:  0.9868131868131869
```

Logistic regression model

Optimizing the model with SelectKBest and GridSearchCV

```
pipeline = Pipeline([
    ('select', SelectKBest(score_func=f_classif)), # Select top features using ANOVA F-value
    ('scale', StandardScaler()), # Standardize the data
    ('classify', LogisticRegression()) # Classifier
])
para = {
    'select__k':[5,8,10],
    'classify__penalty': ['l1', 'l2',None],
    'classify__C': [0.01, 0.1, 1, 10, 100,1000]
}

grid_search = GridSearchCV(pipeline, para, cv=5)
grid_search.fit(train_x, train_y)
```

```
Best hyperparameters:  {'classify__C': 10, 'classify__penalty': 'l2', 'select__k': 8}
Best accuracy score:  0.9897078531224872
```

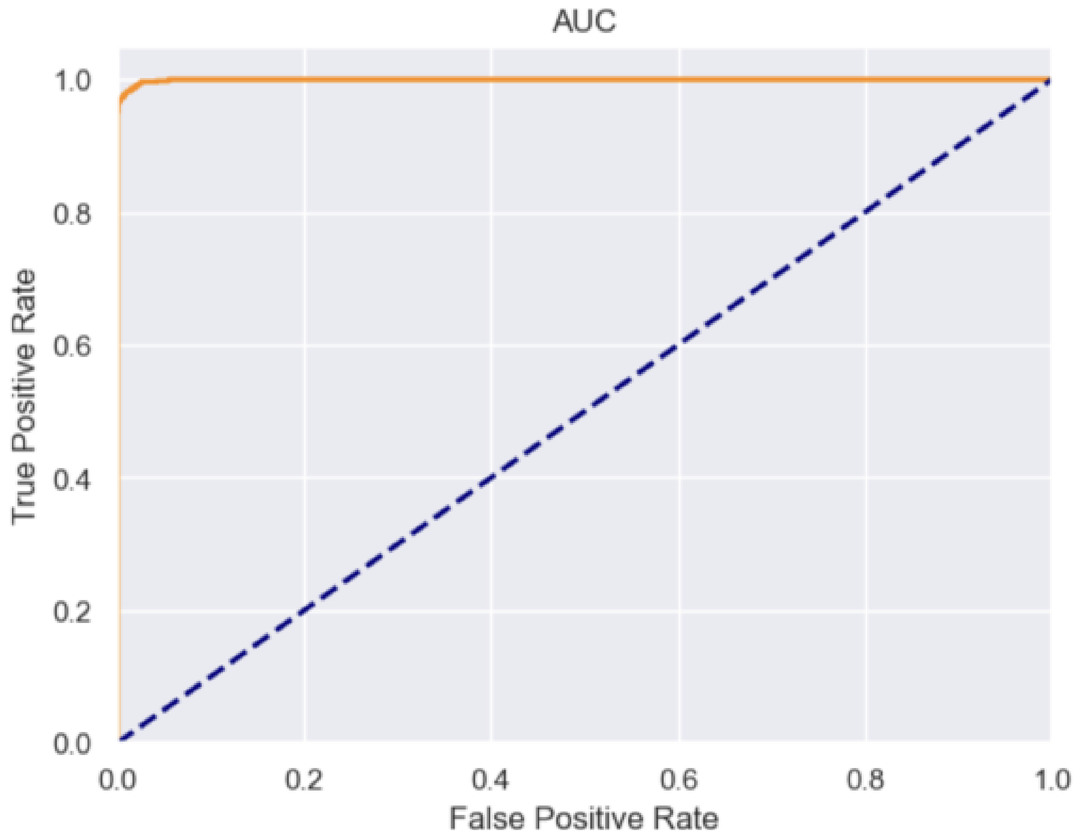
Logistic regression model

The accuracy for the test set

```
The Acuracy is : 98.65 %
Classification report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1217
           1       0.99      0.98      0.98       856

    accuracy                           0.99      2073
   macro avg       0.99      0.99      0.99      2073
weighted avg       0.99      0.99      0.99      2073
```



Linear SVC

The accuracy of the linear SVC model was 97.35% initially.

However, the cross-validation showed a lower accuracy of 92.97%, revealing the model's bias.

Linear SVC

Optimizing the model with GridSearchCV

```
Best hyperparameters: {'C': 1, 'dual': False, 'loss': 'squared_hinge', 'max_iter': 1000, 'penalty': 'l1'}
Best accuracy score:  0.9867594582520532
The Acuracy is : 97.88 %
Classification report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      1217
           1       0.98      0.97      0.97       856

    accuracy                           0.98      2073
   macro avg       0.98      0.98      0.98      2073
weighted avg       0.98      0.98      0.98      2073
```

Random Forest

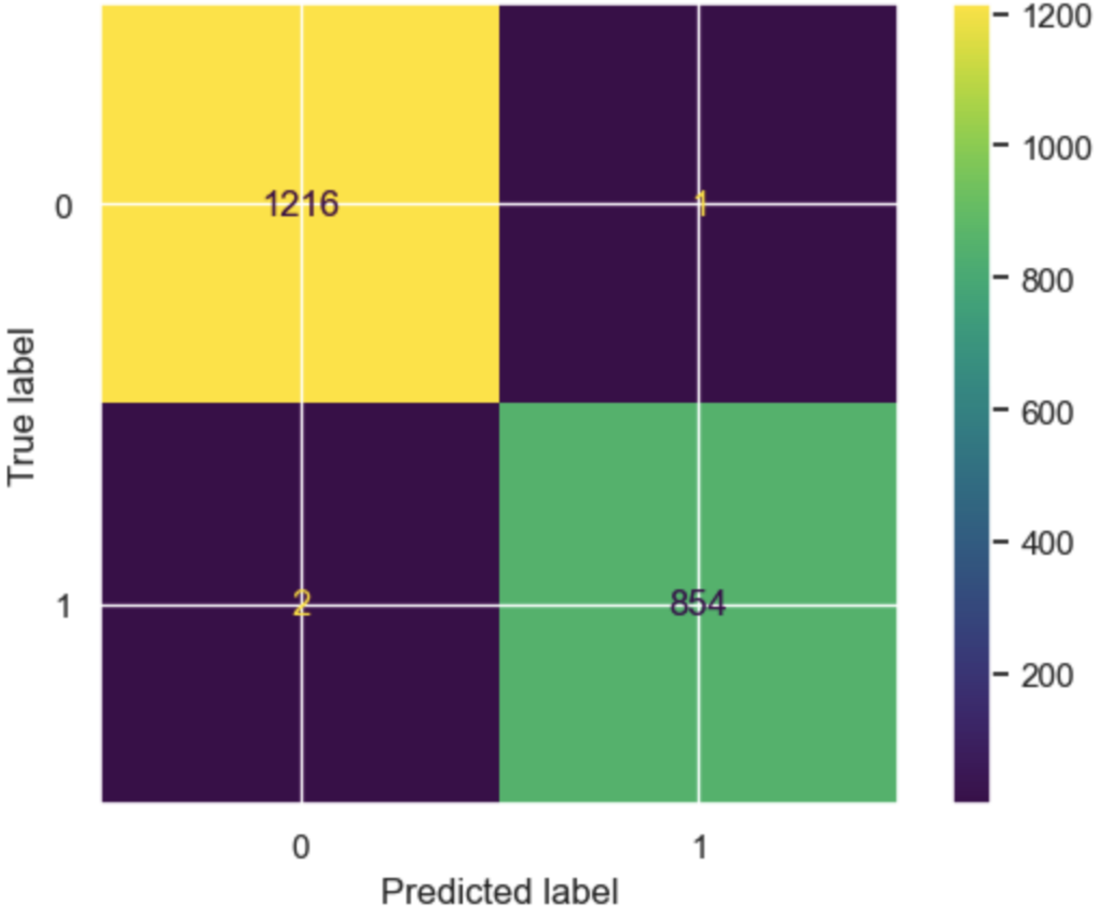
- Set the initial parameter:

RandomForestClassifier(n_estimators = 5,max_leaf_nodes = 5,random_state=42)

- The Accuracy is : 99.13 %

Random Forest

▸ Ran
▸ estimator:
▸ Rando




```
Best hyperpar… 2', 'n_estimators': 142}
Best accuracy
```

Neural Network

- Best Parameters: {'solver': 'adam', 'max_iter': 1000, 'learning_rate_init': 0.001, 'hidden_layer_sizes': (10, 5), 'alpha': 0.01, 'activation': 'logistic'}

```
The Acuracy is : 92.23 %
Classification report:
              precision    recall  f1-score   support

           0       0.88      1.00      0.94      1217
           1       1.00      0.81      0.90       856

    accuracy                           0.92      2073
   macro avg       0.94      0.91      0.92      2073
weighted avg       0.93      0.92      0.92      2073
```

```
The Acuracy is : 97.06 %
Classification report:
              precision    recall  f1-score   support

           0       0.95      1.00      0.98      1217
           1       1.00      0.93      0.96       856

    accuracy                           0.97      2073
   macro avg       0.98      0.96      0.97      2073
weighted avg       0.97      0.97      0.97      2073
```

Naïve Bayes

- The Naïve Bayes model initially achieved a high accuracy score of 95.8%.
- The model's performance worsened after using GridSearchCV to optimize the var_smoothing parameter.
- Combining the SelectKBest and GridSearchCV significantly improved the accuracy score of 96.33%.

The time train the optimized model

- Logistic regression: 0.05 seconds
- SVM: 0.83 seconds
- Random forest: 0.73 seconds
- Neural network: 1.43 seconds
- Naive Bayes: 0.01 seconds

Conclusion

- The Random Forest model demonstrated the highest accuracy score and f1-scores for both classes, making it the top performer.
- Meanwhile, the naive Bayes is the fastest model to train and predict.
- However, the Logistic Regression model is a suitable choice for predicting and evaluating our dataset in this project.

Thank you