SPAM Filtering Modelling for Youtube

2023 - 2024 Term 2
CDS 4001 Best Practices of Data Science

Table of Contents.

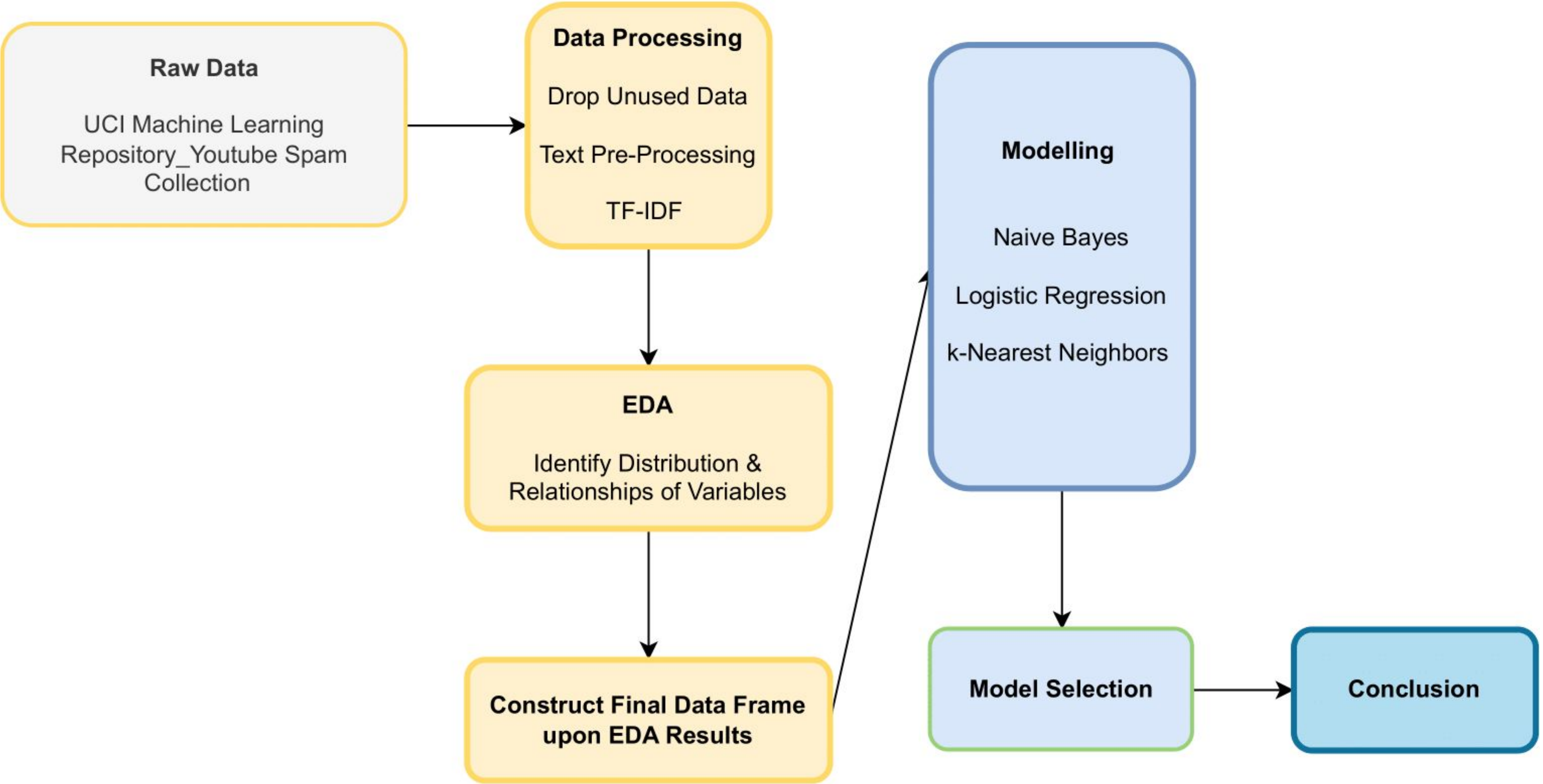
Part 1

Project Pipeline

Data Collection

Data Sampling

Project Pipeline



Data Collection

- **Data Sampling Barriers**
  - Instagram's API: unable to collect comments individually
  - Twitter: require Level 2 account to collect comments
  - Youtube and Facebook tools or API for scraping
    - **BUT** most of the "public" comments are filtered
    - Labeling is one big issue

Data Collection

- Data Source
  - **Time Frame:** The Most Viewed Videos on Youtube_Around 2013
  - **Source:**
    - UC Irvine Machine Learning Repository

Data Sampling

Predicators

Comment	Class
5 Videos 1,956 real Comments	SPAM / NOT-SPAM

Target: SPAM Detection_Optimal Model Selection

Part 2

Data Cleaning

Feature Engineering

EDA

Data Cleaning

- No Missing Values Found
- No Outliers Found
- No Inconsistent / Duplicates Data Found
- Dataset Merging
- Drop Necessary Data
    -

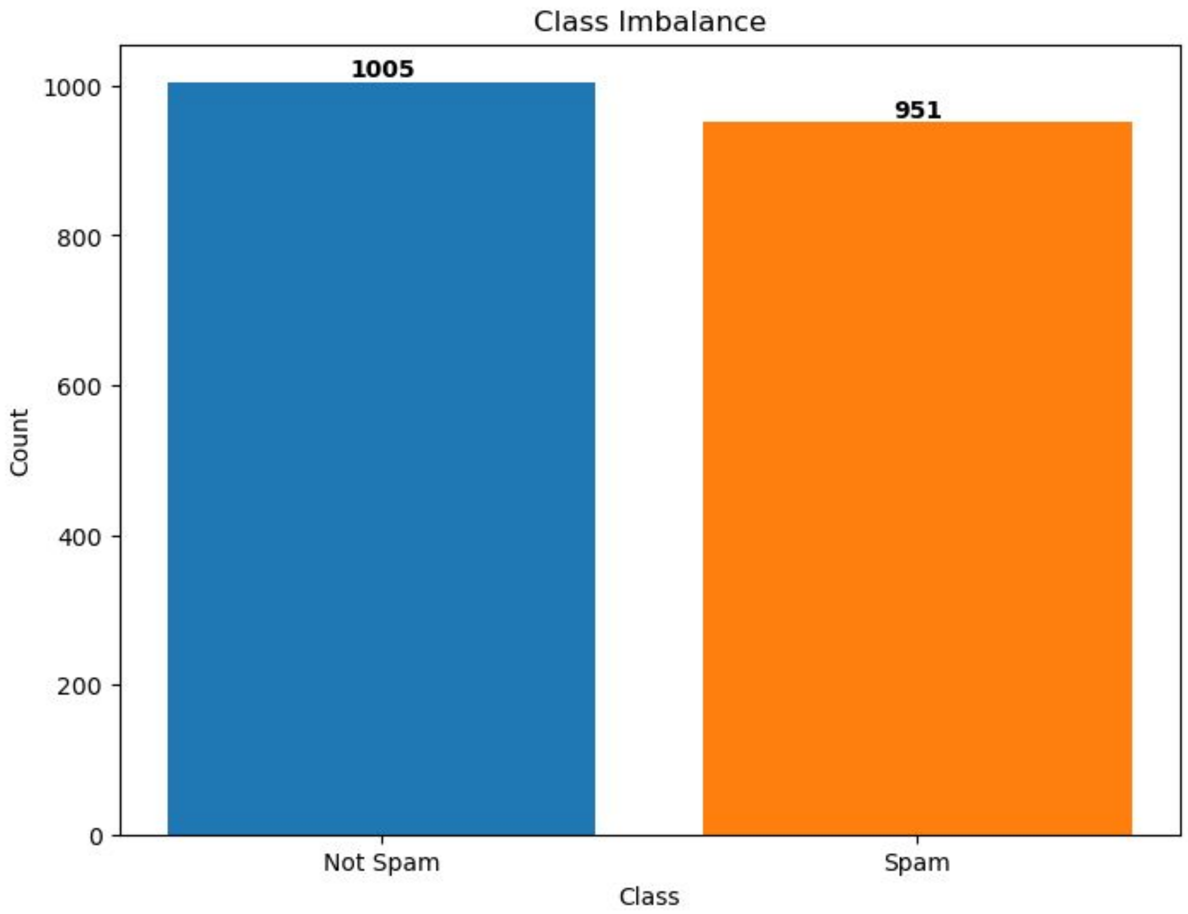
COMMENT_ID	AUTHOR	DATE
LZQPQhLyRh80UYxNuaDWhIGQYNQ96IuCg-AYWqNPjpU	Julius NM	2013-11-07T06:20:48
LZQPQhLyRh_C2cTtd9MvFRJedxydaVW-2sNg5Diuo4A adam	riyati	2013-11-07T12:37:15

Exploratory Data Analysis - EDA

- **Class Imbalance Test**
- **Text Analysis**
  - **Natural Language Processing Techniques**
  - Removal of Stopword, Punctuation & Special Characters
  - Replacement of Email, Numbers & URL
  - Lemmatization: Inflectional endings such as "s," "ed", "ing" are removed
  - Tokenization
- **Feature Extraction via NLTK Library**
  - CountVectorizer
  - Term Frequency-Inverse Document Frequency (TF-IDF)

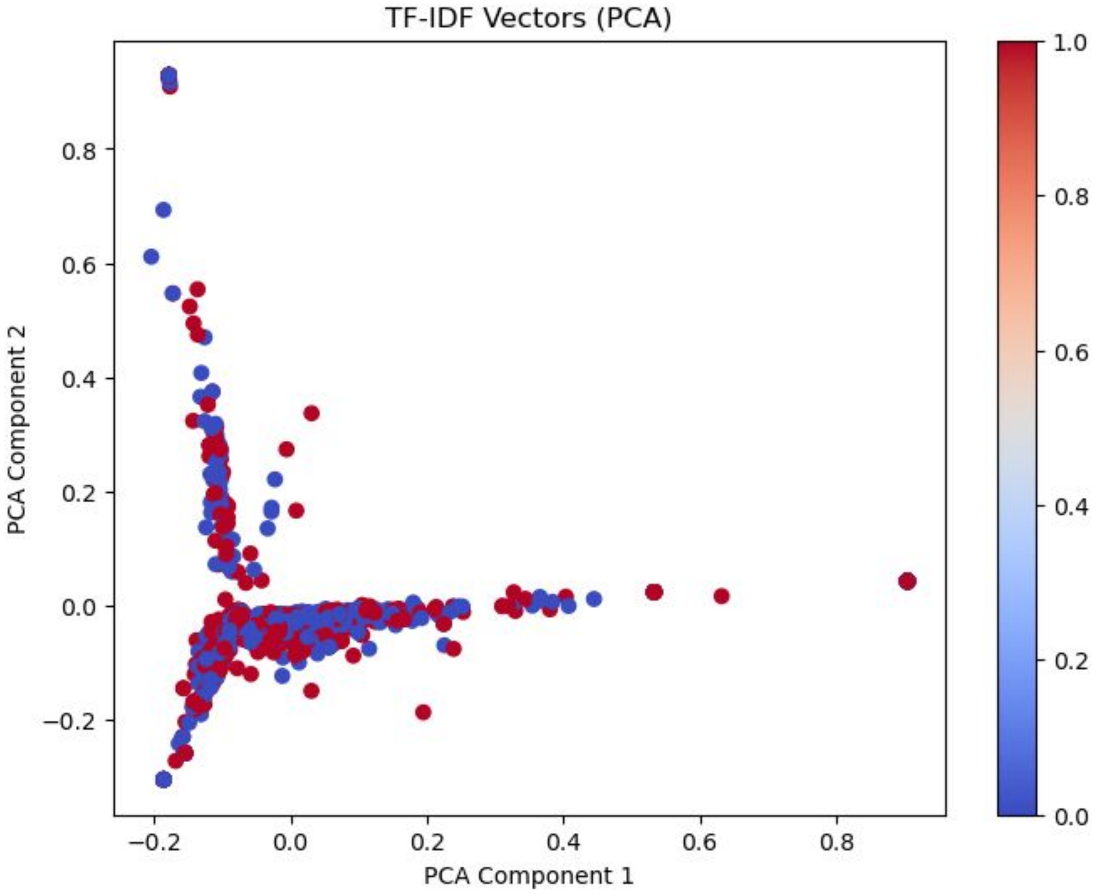
Exploratory Data Analysis - EDA

- **Class Imbalance Test**



Exploratory Data Analysis - EDA

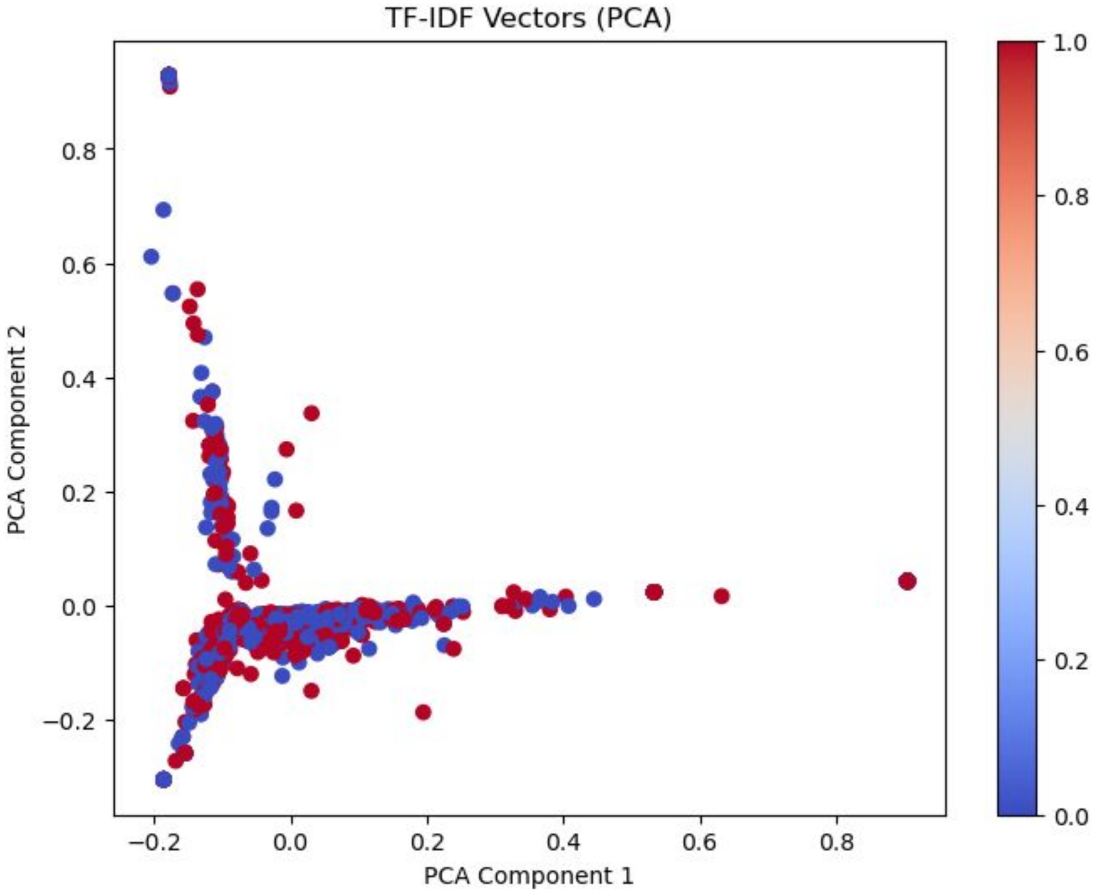
- **Text Analysis**
  - **Natural Language Processing Techniques**
  - Removal of Stopword, Punctuation & Special Characters
  - Replacement of Email, Numbers & URL
  - Lemmatization: Inflectional endings such as "s," "ed", "ing" are removed
  - Tokenization
  - PCA



Exploratory Data Analysis - EDA

- **Text Analysis** (PCA)

- The first principal component (PC1) has an eigenvalue of around 45% of the total variance in the data.
- **PC2, 25%** of the total variance. **PC3, 15%** of the total variance.
- **PC4 and PC5** account for approximately **10% and 8%** of the total variance
- **After the first 5** principal components, the eigenvalues **drop significantly**, indicating that the remaining principal components **contribute much less** to the overall variance in the data.
- Should chose to retain the first 5-10 principal components



Exploratory Data Analysis - EDA

- **Word Cloud**

Word Cloud of Most Frequent Words

Part 3

Data Modelling

1st Result

Final DataFrame

Datasets	Spam	Not Spam	Total
Psy	175	175	350
KatyPerry	175	175	350
LMFAO	236	202	438
Eminem	245	203	448
Shakira	174	196	370

Implemented Machine Learning Techniques

1. **Logistic Regression**
   - Easy to interpret & binary classification
2. **Support Vector Machine**
   - High-dimensional spaces(TF-IDF)
3. **Random Forest**
   - Usually high performance
4. **KNN**

5. **Neural Network**
   - Handle nonlinear and complex relationships

Default Model

	Logistic Regression	Support Vector Machine	Random Forest	KNN	Neural Network
“Default” Accuracy	0.931	0.931	0.941	0.633	0.85

Part 4

Fine-Tuning

Final Modelling Result

Fine-Tuning Methods

- Parameter tuning & cross-validation
  - **GridSearchCV**, **StratifiedKFold** from sklearn

```
stratified_kfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
```

  - KNN example:

```
knn_param_grid = ({'n_neighbors': [i for i in range(3,40) ],
                   'weights': ['uniform', 'distance']})
```

```
knn_grid_search = (GridSearchCV(KNeighborsClassifier(),
                                knn_param_grid,
                                cv=stratified_kfold,
                                scoring='accuracy'))
```

Fine-Tuning Methods

- Parameter tuning
  - **trial.suggest_int**, **trial.suggest_loguniform** from optuna

```
hidden_size = trial.suggest_int('hidden_size',16, 128)
learning_rate = trial.suggest_loguniform('learning_rate',1e-4, 1e-1)
```

- Optimizer
  - Adam

```
optimizer = optim.Adam(model_nn.parameters(), lr=learning_rate)
```

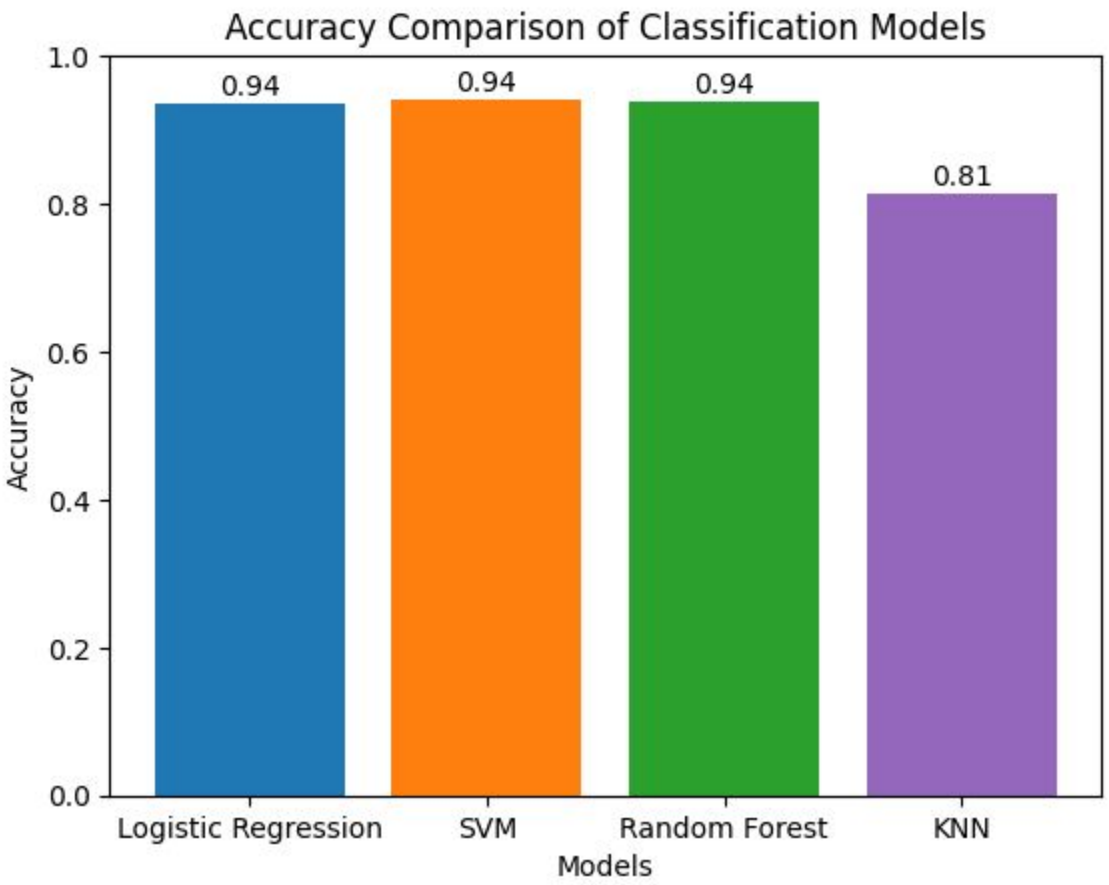
- loss function
  - BCELoss

```
loss_fn = nn.BCELoss()
```

Default Model vs Fine-tuned Model

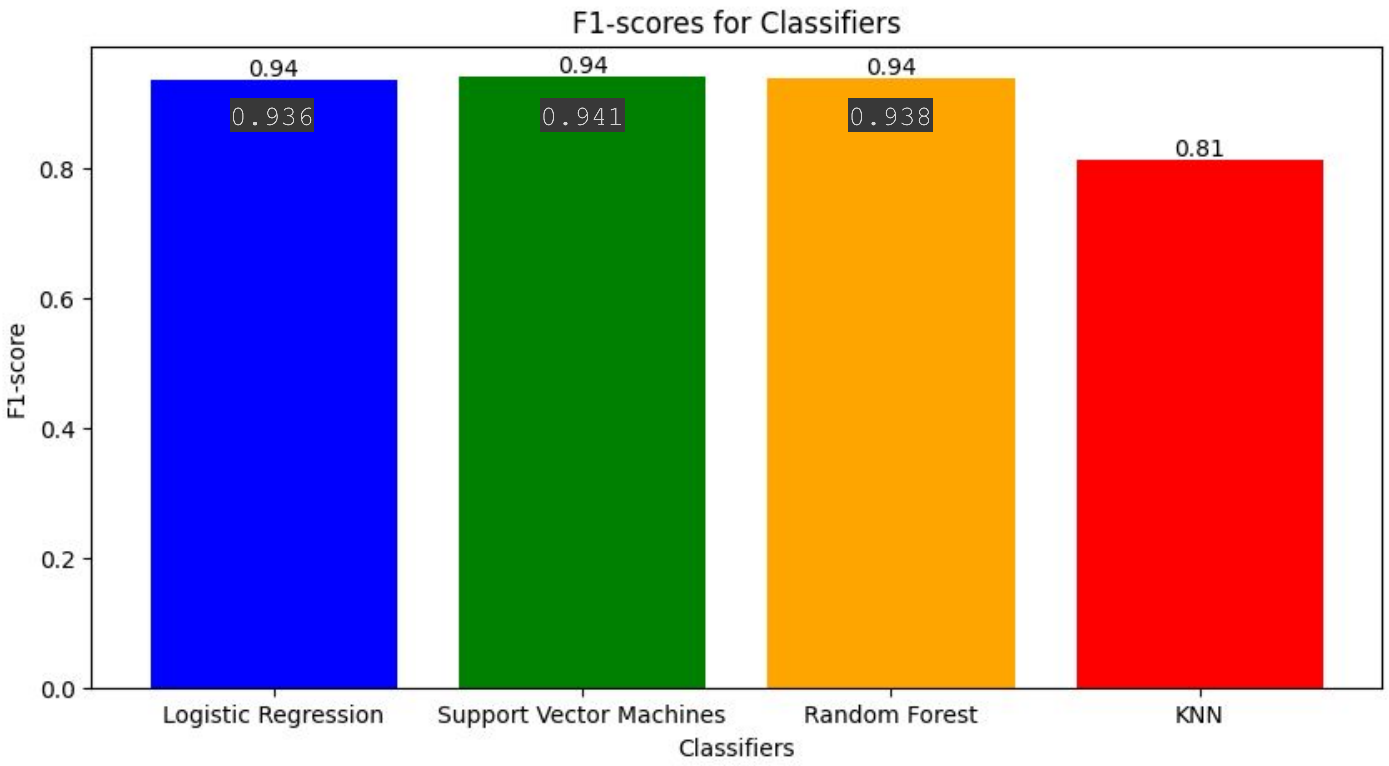
Method	"Default" Accuracy	"Fine-tuned" Accuracy	Parameters Setting
Logistic Regression	0.931	0.936	'C': 10, 'penalty': 'l2', 'solver': 'liblinear'
Support Vector Machine	0.931	0.941	'C': 1, 'kernel': 'linear'
Random Forest	0.941	0.941(same)	'max_depth': None 'n_estimators': 100
KNN	0.633	0.814	'n_neighbors': 26, 'weights': 'distance'
Neural Network	0.85	0.91	'hidden_size': 76, 'learning_rate': 0.0276

Modelling Result - Accuracy



- **Random Forest & SVM**
- KNN:
  - Correctly **classified 81%** of the instances in the testing dataset
  - Struggled to **accurately identify the characteristics** or patterns that differentiate spam comments from non-spam comments

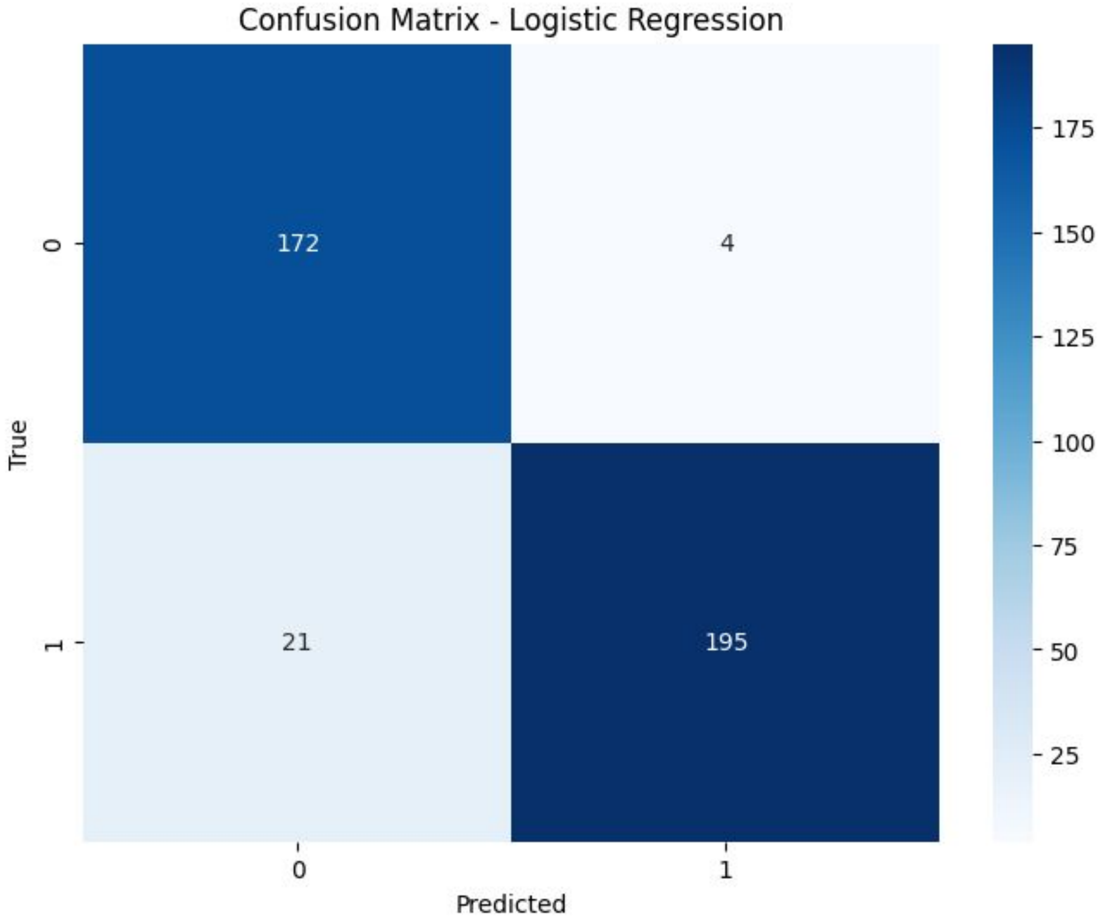
Modelling Result - F1 Scores



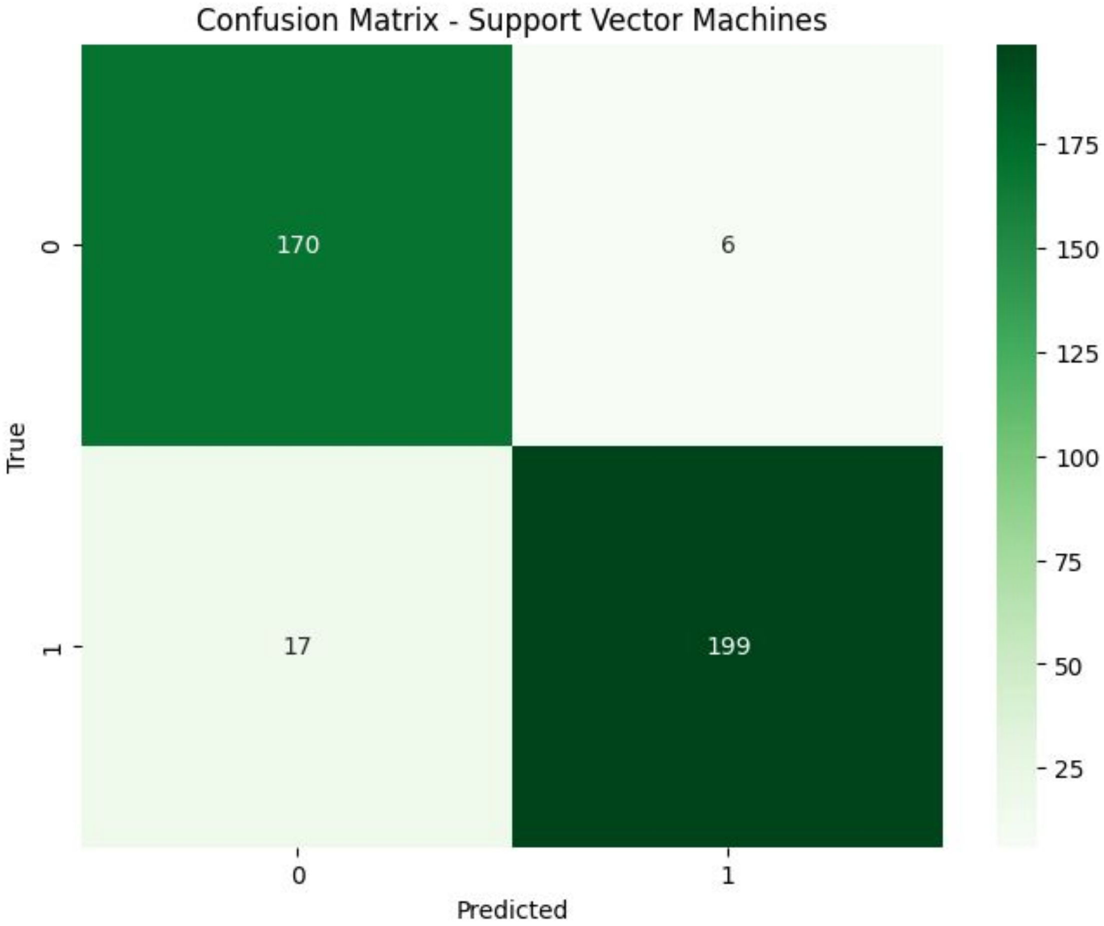
- **SVM:**
  - Excelled in balancing precision and recall and demonstrated a strong ability to accurately classify spam comments.
- **KNN:**
  - Higher number of misclassifications

Modelling Result - Confusion Matrix

Logistic Regression

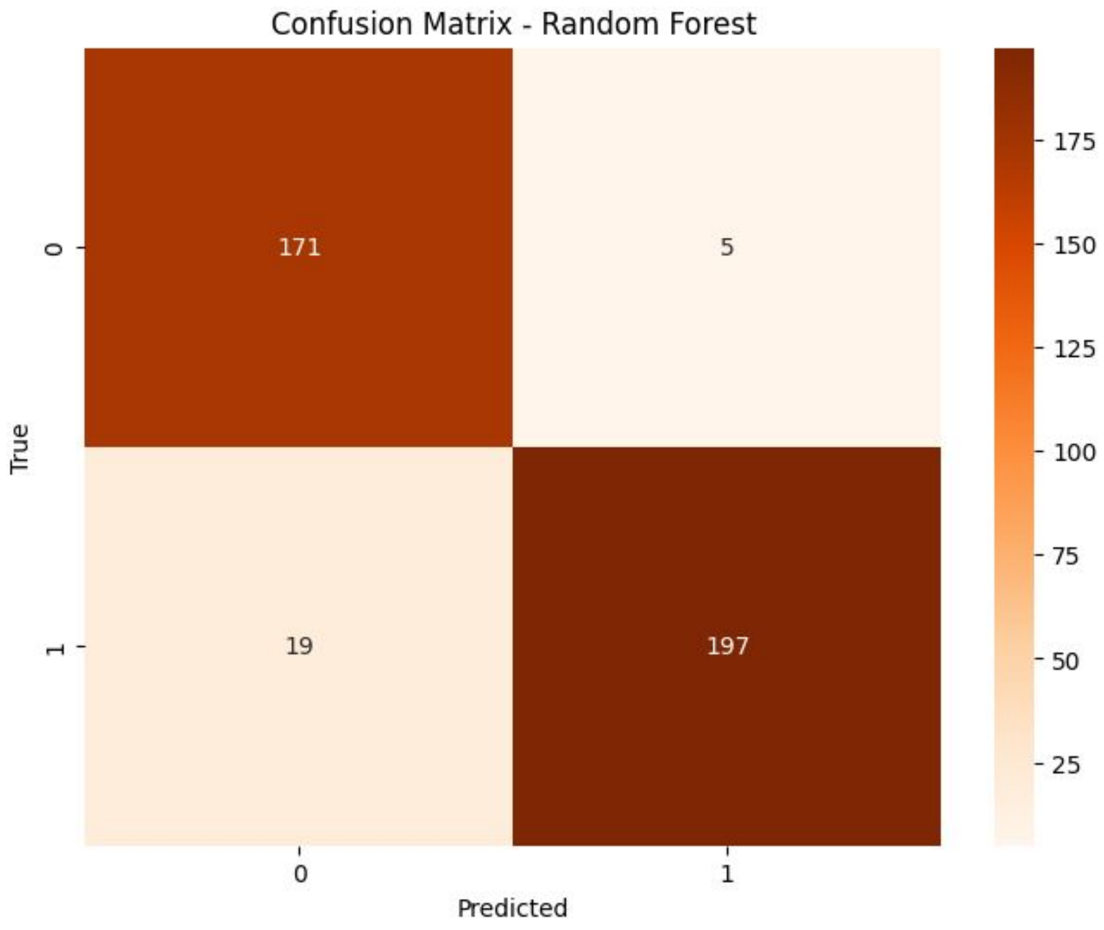


Support Vector Machines

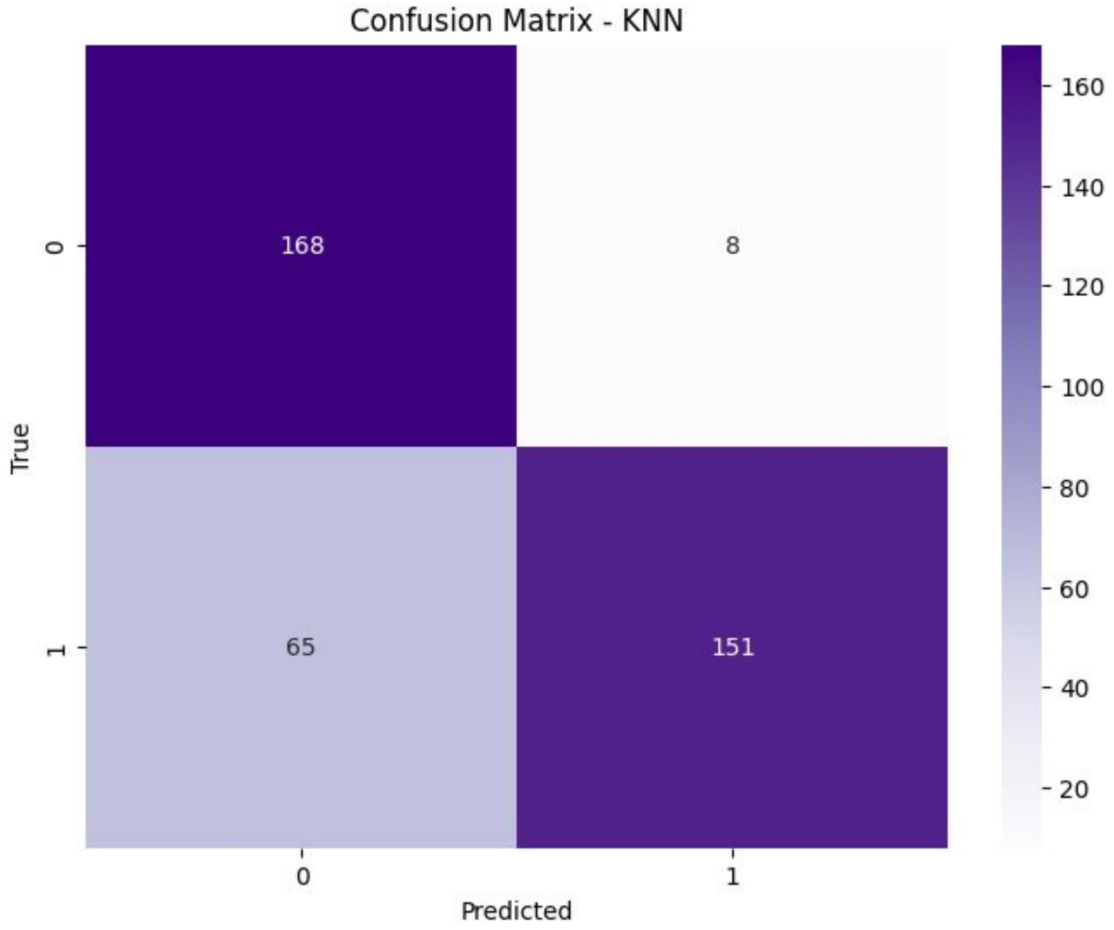


Modelling Result - Confusion Matrix

Random Forest

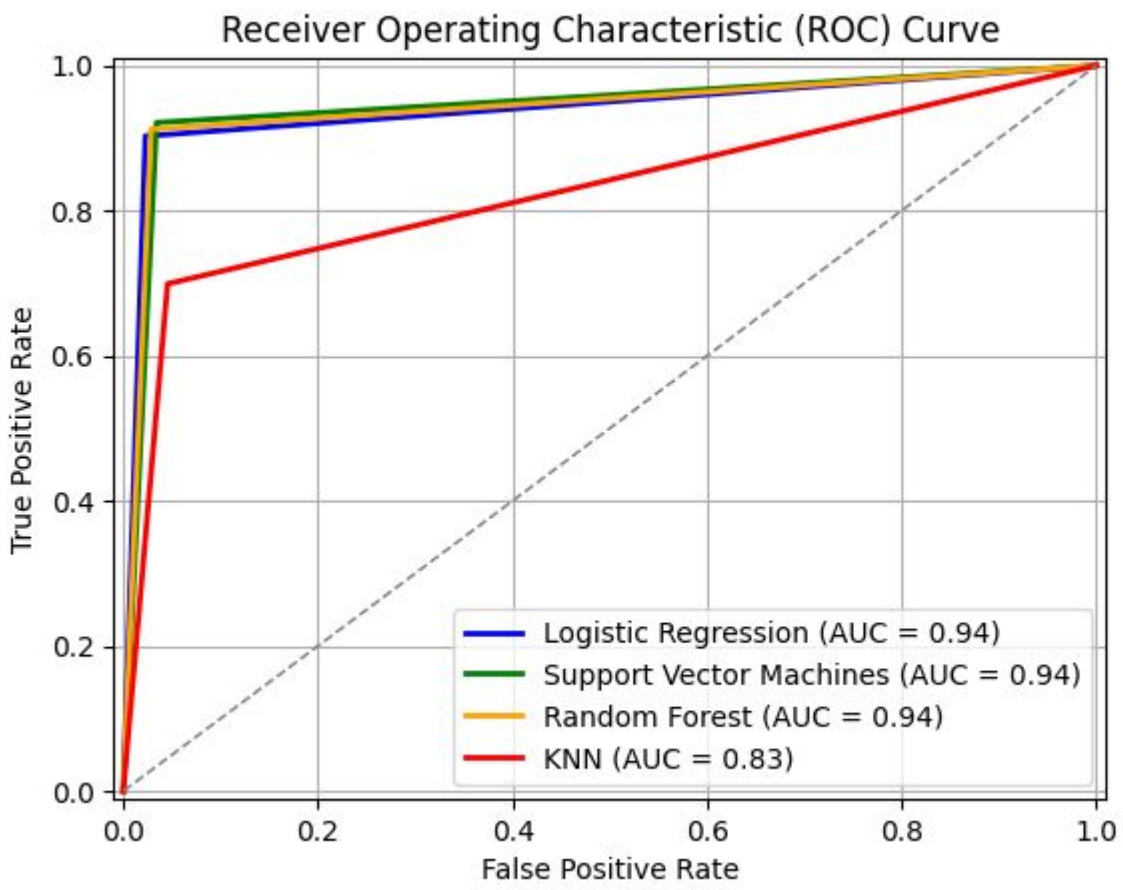


KNN



Modelling Result

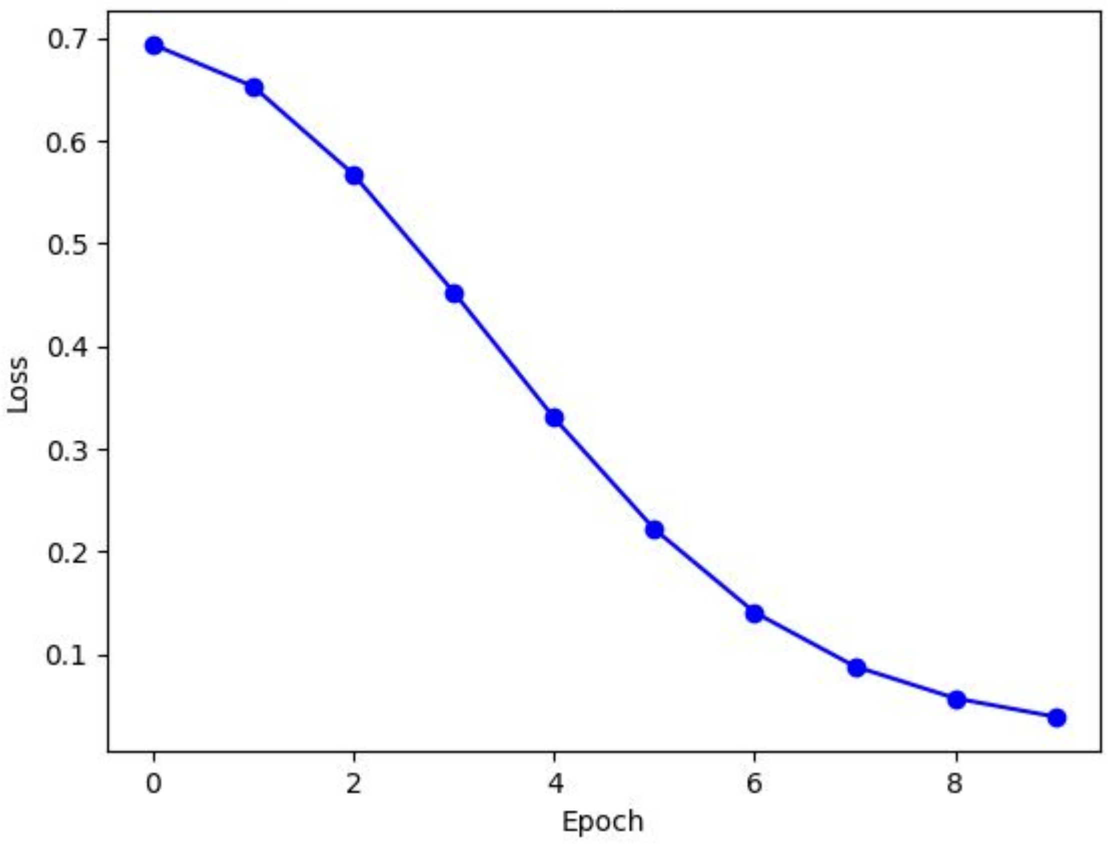
ROC Curve



- **Random Forest**
  - Favorable balance between true positive and false positive rate
  - High Predictive Power
- **KNN:**
  - AUC value: 0.83
  - Lower Discriminative Ability
  - Limited Predictive Power

Further Modelling Result - Neural Network

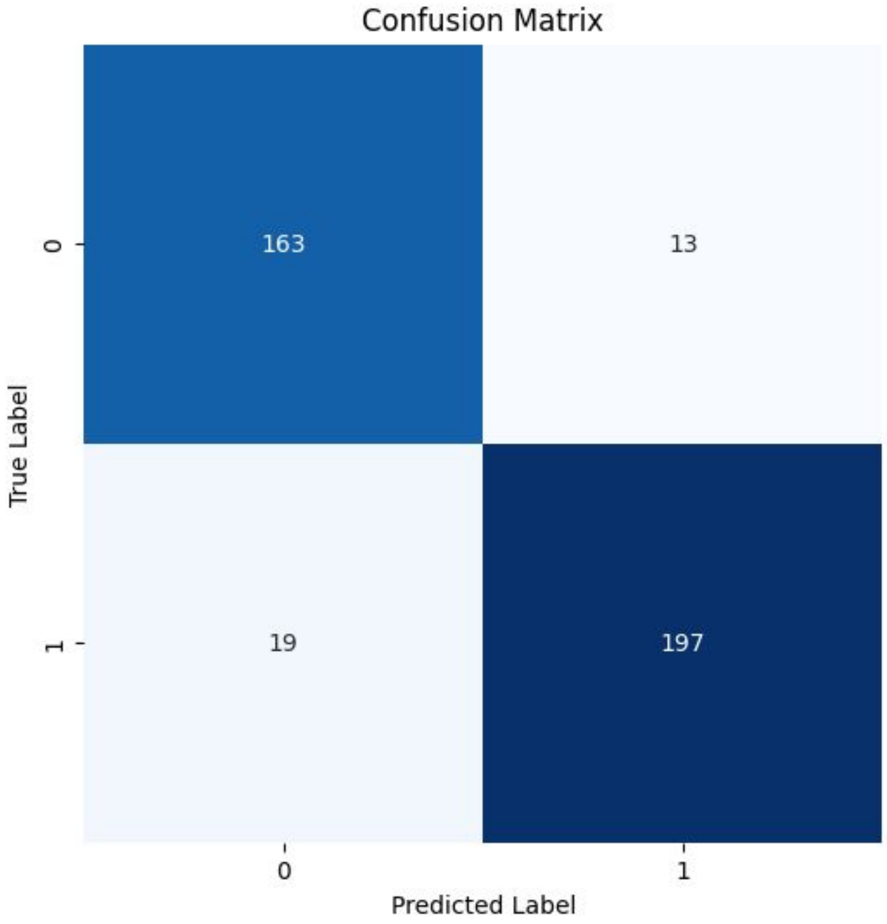
Neural Network



- **Accuracy - 91.8%**
- **10 Epochs**
  - Training loss gradually decreased with each epoch
  - Learning and adjusting its weights to minimize the loss function

Further Modelling Result - Neural Network

Neural Network



Final Model

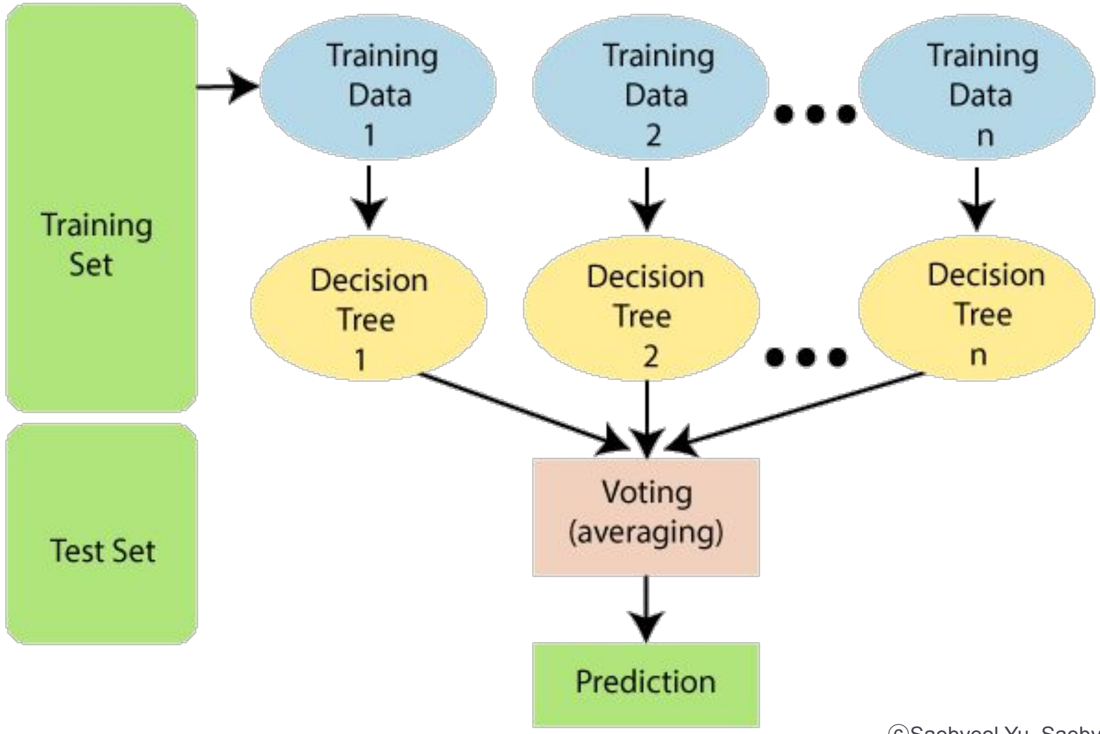
Method	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.93	0.94	0.93	0.93
Support Vector Machine	0.93	0.94	0.93	0.93
Random Forest	0.94	0.94	0.94	0.94
KNN	0.81	0.85	0.81	0.81
Neural Network	0.92	0.92	0.92	0.92

Final Model Selection

Method	Accuracy	Precision	Recall	F1-Score
Random Forest	0.94	0.94	0.94	0.94

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (fi - yi)^2$$

Where *N* is the number of data points, *fi* is the value returned by the model and *yi* is the actual value for data point *i*.



Final Model Selection

Method	Accuracy	Precision	Recall	F1-Score
Random Forest	0.94	0.94	0.94	0.94

- Compare to SVM
  - Easy to Interpret
  - Faster training time
  - Youtube Comment(small size) & TF-IDF
  - Recall & F1 higher