TERM PROJECT

SUBMISSION DATE (June 24, 2024)

BY

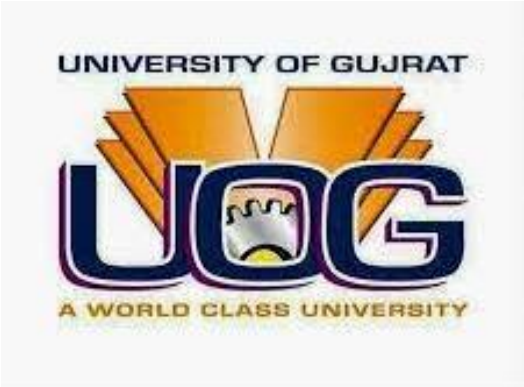
ABDULLAH KHAN

ROLL # 20014119-037

Course Code (CS-447)

Bachelor's in Computer Science (A)

Dr. Zahid Iqbal



UNIVERSITY OF GUJRAT
(HAFIZ HAYAT CAMPUS)

Chapter No. 01

Project Proposal

Problem Statement:

The objective is to ***develop a machine learning model that can accurately classify SMS messages as either spam or not spam.*** The model will be evaluated based on its ability to correctly identify spam messages while minimizing false positives for legitimate messages.

Dataset:

SMS Spam Collection Dataset:

This dataset contains 5,572 SMS messages in English, labeled as either "ham" or "spam". I have sourced this dataset from Kaggle - SMS Spam Collection Dataset. The format of the dataset is in CSV file with two columns: "label" (spam or ham) and "message" (the text of the SMS).

The content dataset consists of real SMS messages, providing a realistic set of examples for training and evaluating a spam detection model. Each message is pre-labeled, simplifying the supervised learning process. ***ham*** *indicates a legitimate message*, and ***spam*** *indicates an unsolicited or unwanted message*. The dataset will be split into training and testing sets. The messages will undergo preprocessing to transform them into a format suitable for machine learning models.

Classifiers to Apply:

We will use the following classifiers:

1. **Logistic Regression:**
    - **Advantages:** Simple, fast, and effective for text classification problems.
2. **Support Vector Machine (SVM):**
    - **Advantages:** High accuracy, particularly effective for text data, and good at handling high-dimensional data.
3. **Random Forest Classifier:**
    - **Advantages:** High accuracy, robustness to overfitting, and good performance with imbalanced datasets.

Implementation Plan:

Data Preprocessing:

- Load and clean the datasets.
- Remove duplicates and irrelevant information.
- Normalize the text by converting it to lowercase, removing stop words and stemming.

Feature Extraction:

- Convert text data into numerical features using techniques like TF-IDF (Term Frequency-Inverse Document Frequency).

Model Training:

- Split the datasets into training and testing sets.
- Train the Logistic Regression, SVM, and Random Forest classifiers on the training data. Tune hyperparameters using cross-validation.

Model Evaluation:

- Evaluate the models on the test data using accuracy metrices.
- Compare the performance of the different classifiers.

Model Deployment:

- Develop a simple web interface or API to demonstrate the spam detection capability of the best performing model

Chapter No. 02

Implementation & Analysis

Importing Required Libraries:

This section imports the necessary libraries for data manipulation, model training and evaluation including NumPy, Pandas and Scikit Learn.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

Reading & Printing the Dataset:

The dataset is loaded from a CSV file and printed to verify its contents. Any missing values are replaced with empty strings.

```
df = pd.read_csv('mail_data.csv')
```

```
print(df)
```

```
     Category                                            Message
0         ham  Go until jurong point, crazy.. Available only ...
1         ham                      Ok lar... Joking wif u oni...
2        spam  Free entry in 2 a wkly comp to win FA Cup fina...
3         ham  U dun say so early hor... U c already then say...
4         ham  Nah I don't think he goes to usf, he lives aro...
...       ...                                                ...
5567     spam  This is the 2nd time we have tried 2 contact u...
5568      ham              Will ü b going to esplanade fr home?
5569      ham  Pity, * was in mood for that. So...any other s...
5570      ham  The guy did some bitching but I acted like i'd...
5571      ham                        Rofl. Its true to its name

[5572 rows x 2 columns]
```

```
data = df.where((pd.notnull(df)), '')
```

```
data.head(10)
```

	Category	Message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...
5	spam	FreeMsg Hey there darling it's been 3 week's n...
6	ham	Even my brother is not like to speak with me. ...
7	ham	As per your request 'Melle Melle (Oru Minnamin...
8	spam	WINNER!! As a valued network customer you have...
9	spam	Had your mobile 11 months or more? U R entitle...

Preprocessing the Data:

The ‘Category’ column is converted from ‘spam’ to ‘ham’ to numerical values 0 and 1 respectively. The data is then split into input (X) and output (Y) variables.

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Category  5572 non-null   object
 1   Message   5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

```
data.loc[data['Category'] == 'spam', 'Category',] = 0
data.loc[data['Category'] == 'ham', 'Category',] = 1
```

```
X = data['Message']
Y = data['Category']
```

```
print(X)
```

```
0       Go until jurong point, crazy.. Available only ...
1                           Ok lar... Joking wif u oni...
2       Free entry in 2 a wkly comp to win FA Cup fina...
3       U dun say so early hor... U c already then say...
4       Nah I don't think he goes to usf, he lives aro...
                              ...
5567    This is the 2nd time we have tried 2 contact u...
5568                 Will ü b going to esplanade fr home?
5569    Pity, * was in mood for that. So...any other s...
5570    The guy did some bitching but I acted like i'd...
5571                           Rofl. Its true to its name
Name: Message, Length: 5572, dtype: object
```

```
print(Y)
```

```
0       1
1       1
2       0
3       1
4       1
       ..
5567    0
5568    1
5569    1
5570    1
5571    1
Name: Category, Length: 5572, dtype: object
```

Splitting the Dataset:

The dataset is divided into training and test sets to evaluate model performance. The shapes of these sets are printed to ensure correct spelling.

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=3)
```

```
print(X.shape)
print(X_train.shape)
print(X_test.shape)
```

```
(5572,)
(4457,)
(1115,)
```

```
print(Y.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(5572,)
(4457,)
(1115,)
```

Transforming the Data:

The text data is transformed into TF-IDF feature vectors to convert the text into numerical values suitable for machine learning models.

```
feature_extraction = TfidfVectorizer(min_df=1, stop_words='english', lowercase=True)

X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)

Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')
```

```
print(X_train)
```

```
3075                 Don know. I did't msg him recently.
1787    Do you know why god created gap between your f...
1614                          Thnx dude. u guys out 2nite?
4304                                       Yup i'm free...
3266    44 7732584351, Do you want a New Nokia 3510i c...
                              ...
789     5 Free Top Polyphonic Tones call 087018728737,...
968     What do u want when i come back?.a beautiful n...
1667    Guess who spent all last night phasing in and ...
3321    Eh sorry leh... I din c ur msg. Not sad alread...
1688    Free Top ringtone -sub to weekly ringtone-get ...
Name: Message, Length: 4457, dtype: object
```

```
print(X_train_features)
```

```
  (0, 5413)     0.6198254967574347
  (0, 4456)     0.4168658090846482
  (0, 2224)     0.413103377943378
  (0, 3811)     0.34780165336891333
  (0, 2329)     0.38783870336935383
  (1, 4080)     0.18880584110891163
  (1, 3185)     0.29694482957694585
  (1, 3325)     0.31610586766078863
  (1, 2957)     0.3398297002864083
  (1, 2746)     0.3398297002864083
  (1, 918)      0.22871581159877646
  (1, 1839)     0.2784903590561455
  (1, 2758)     0.3226407885943799
  (1, 2956)     0.33036995955537024
  (1, 1991)     0.33036995955537024
  (1, 3046)     0.2503712792613518
  (1, 3811)     0.17419952275504033
  (2, 407)      0.509272536051008
  (2, 3156)     0.4107239318312698
  (2, 2404)     0.45287711070606745
  (2, 6601)     0.6056811524587518
  (3, 2870)     0.5864269879324768
  (3, 7414)     0.8100020912469564
  (4, 50)       0.23633754072626942
  (4, 5497)     0.15743785051118356
  :     :
```

Training and Evaluating Logistic Regression Model:

A logistic regression model is trained on the training data and its accuracy on both the training and test sets is calculated and printed.

```
model = LogisticRegression()

model.fit(X_train_features, Y_train)

prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train, prediction_on_training_data)

print('Acc on training data: ', accuracy_on_training_data)

prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test, prediction_on_test_data)

print('Acc on test data: ', accuracy_on_test_data)
```

```
Acc on training data:  0.9670181736594121
Acc on test data:  0.9659192825112107
```

Training and Evaluating SVM Model:

An SVM model is trained and evaluated similarly, with accuracy on the training and test sets being calculated and printed.

```
svm_model = SVC()

svm_model.fit(X_train_features, Y_train)

svm_train_predictions = svm_model.predict(X_train_features)

svm_train_accuracy = accuracy_score(Y_train, svm_train_predictions)
print('SVM accuracy on training data: ', svm_train_accuracy)

svm_test_predictions = svm_model.predict(X_test_features)

svm_test_accuracy = accuracy_score(Y_test, svm_test_predictions)
print('SVM accuracy on test data: ', svm_test_accuracy)
```

```
SVM accuracy on training data:  0.99798070450976
SVM accuracy on test data:  0.979372197309417
```

Training and Evaluating Random Forest Model:

A random forest model is trained and evaluated in the same manner, with the accuracy on both sets printed for comparison.

```
random_forest_model = RandomForestClassifier()

random_forest_model.fit(X_train_features, Y_train)

rf_train_predictions = random_forest_model.predict(X_train_features)

rf_train_accuracy = accuracy_score(Y_train, rf_train_predictions)
print('Random Forest accuracy on training data: ', rf_train_accuracy)

rf_test_predictions = random_forest_model.predict(X_test_features)

rf_test_accuracy = accuracy_score(Y_test, rf_test_predictions)
print('Random Forest accuracy on test data: ', rf_test_accuracy)
```

```
Random Forest accuracy on training data:  1.0
Random Forest accuracy on test data:  0.9748878923766816
```

Testing with Example Email:

An example email is transformed into a feature vector and classified using all three models. The predictions are printed to indicate whether the email is spam or ham according to each model.

```
input_your_mail = ["This is the 2nd time we have tried to contact u. U have won the Â£1450 prize to cl

input_data_features = feature_extraction.transform(input_your_mail)
```

```
logistic_prediction = logistic_model.predict(input_data_features)
print('Logistic Regression prediction: ', logistic_prediction)
print('Logistic Regression - Ham' if logistic_prediction[0] == 1 else 'Logistic Regression - Spam')

svm_prediction = svm_model.predict(input_data_features)
print('SVM prediction: ', svm_prediction)
print('SVM - Ham' if svm_prediction[0] == 1 else 'SVM - Spam')

rf_prediction = random_forest_model.predict(input_data_features)
print('Random Forest prediction: ', rf_prediction)
print('Random Forest - Ham' if rf_prediction[0] == 1 else 'Random Forest - Spam')
```

```
Logistic Regression prediction:  [0]
Logistic Regression - Spam
SVM prediction:  [0]
SVM - Spam
Random Forest prediction:  [0]
Random Forest - Spam
```

Performance Comparison

Model	Training Accuracy	Testing Accuracy
Logistic Regression	96.7%	96.59%
SVM	99.8%	97.94%
Random Forest	100%	97.49%

Analysis

Logistic Regression	SVM	Random Forest
Accuracy Comparison		
It has the lowest test accuracy rate at 96.59%, but it is still quite high and close to others.	It has the highest test accuracy rate at 97.94%, indicating it generalizes better on unseen data compared to LR and RF.	It also performs well with a test accuracy of 97.49%, slightly lower than SVM.
Overfitting		
It has a balanced training and test accuracy, suggesting it does not overfit much.	It also has a high training accuracy and high test accuracy, but the gap between training and test accuracy is smaller than RF, suggesting it generalizes well without much overfitting.	It shows signs of overfitting with a perfect training accuracy but a slightly lower test accuracy. This indicates that while it performs perfectly on training data, it may struggle a bit with generalizing to new data.

Computational Complexity		
It is the least computationally expensive model to train and predict, making it suitable for large datasets or scenarios where quick predictions are required.	It can be computationally expensive, especially with large datasets and high dimensional data due to complexity of finding the optimal hyperplane.	It can also be computationally expensive due to the need to build and evaluate multiple decision trees, but it benefits from parallelism.
Interpretability		
It is highly interpretable, as it provides coefficients for each feature, showing the impact of each word on the prediction.	Such a model with a linear kernel is somewhat interpretable but becomes a black ox model with nonlinear models.	It is less interpretable because it involves many decision trees and averaging their results.