IBM Naan Mudhalvan

Phase 3 – Project Submission

Topic:

“Building a Smarter AI-Powered Spam Classifier”

To get started with building a smarter AI-powered spam classifier, the following steps are mentioned below:

1. Loading the Dataset:

- Obtain a spam dataset that includes labeled examples of spam and non-spam (ham) messages.

2. Preprocessing the Dataset:

- **Text Cleaning:**

Remove any special characters, punctuation, or irrelevant symbols from the messages.

Convert all text to lowercase to ensure consistency.

- **Tokenization:**

Split the messages into individual words or tokens.

- **Stop word Removal:**

Remove common words like 'and', 'the', 'is', etc., as they don't provide meaningful information for classification.

- **Vectorization:**

Convert text data into numerical vectors using techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings.

3. Exploratory Data Analysis (EDA):

- Analyze the distribution of spam and non-spam messages in the dataset.

- Visualize the distribution using plots or charts to gain insights.

4. Feature Selection:

- Identify relevant features or words that can help distinguish between spam and non-spam messages.
- Utilize techniques like feature importance scores to select the most informative features.

5. Model Selection and Training:

- Choose an appropriate machine learning algorithm like Naïve Bayes, Support Vector Machines (SVM), or deep learning models such as LSTM or CNN.
- Split the dataset into training and testing sets for model training and evaluation.
- Train the selected model using the training dataset.

6. Model Evaluation:

- Evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score.
- Fine-tune the model if necessary to improve its performance.

7. Document Creation:

- Document the entire process, including dataset details, preprocessing steps, chosen algorithms, performance metrics, and any challenges faced.
- Include visualizations and analysis results in the document.

Code:

```
# Import necessary libraries
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

# Load the dataset (Assuming you have a CSV file with 'text' as messages
and 'label' as spam or ham)
data = pd.read_csv('spam.csv')

# Preprocessing: Convert text to lowercase, remove special characters,
and tokenize
data['text'] = data['text'].str.lower().replace('[^a-zA-Z\s]', '', regex=True)

# Split data into features (X) and labels (y)
X = data['text']
y = data['label']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Vectorize the text data using TF-IDF Vectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)
```

```
# Choose a classifier (Multinomial Naive Bayes in this example)
classifier = MultinomialNB()

# Train the classifier
classifier.fit(X_train_tfidf, y_train)

# Predictions
predictions = classifier.predict(X_test_tfidf)

# Evaluate the model
accuracy = accuracy_score(y_test, predictions)
print('Accuracy:', accuracy)

# Generate a classification report
print('Classification Report:')
print(classification_report(y_test, predictions))
```