**Assessment Test – Associate AI Engineer**

**Candidate: Karthik BR**

**Submission Date: 06-08-2024**

## Project Overview

The task is to classify products into their respective categories (or subcategories) based on their descriptions and names. The dataset provided contains 5000 observations, with each row representing a product along with its category and subcategory.

## Steps Followed

### 1. Initial Pre-processing

* **Data Loading**: Loaded the dataset using pandas.
* **Data Cleaning**: Removed any null or duplicate entries.
* **Text Pre-processing**: Performed text pre-processing which includes:
  + Lowercasing all text
  + Removing punctuation and special characters
  + Removing stop words
  + Lemmatization

### 2. Exploratory Data Analysis

* **Category Distribution**: Analysed the distribution of products across different categories and subcategories.
* **Word Cloud**: Generated word clouds for different categories to visualize the most common words.

### 3. Preparing Data/Transformations

* **Vectorization**: Used TF-IDF vectorization to convert text data into numerical format suitable for model training.
* **Train-Test Split**: Split the dataset into training and testing sets with a 80-20 ratio.

### 4. Modelling

Two models were developed and evaluated:

#### Model 1: Logistic Regression

* **Algorithm**: Logistic Regression
* **Accuracy**: 76%

precision recall f1-score support

0.0 0.76 1.00 0.86 751

1.0 0.00 0.00 0.00 242

accuracy 0.76 993

macro avg 0.38 0.50 0.43 993

#### weighted avg 0.57 0.76 0.65 993

#### Model 2: Random Forest

* **Algorithm**: Random Forest
* **Accuracy**: 75%

precision recall f1-score support

0.0 0.75 0.99 0.85 739

1.0 0.50 0.02 0.03 254

accuracy 0.74 993

macro avg 0.62 0.51 0.44 993

weighted avg 0.68 0.74 0.64 993

### 5. Production-Level Code Quality

* **Modularization**: The code is structured in a modular fashion with separate functions for data loading, pre-processing, model training, and evaluation.
* **Documentation**: Each function and class are well-documented with clear descriptions of their purposes and parameters.
* **Code Standards**: Followed PEP 8 coding standards to ensure readability and maintainability.
* **Error Handling**: Implemented robust error handling to manage unexpected inputs and edge cases.

## Conclusion

Both models, Logistic Regression and Random Forest, performed reasonably well, achieving accuracies of 76% and 75%, respectively. The Random Forest model demonstrated slightly better performance. The project is structured in a way that allows for easy scalability and maintainability, following best practices in software development.

Thank you for considering my submission.