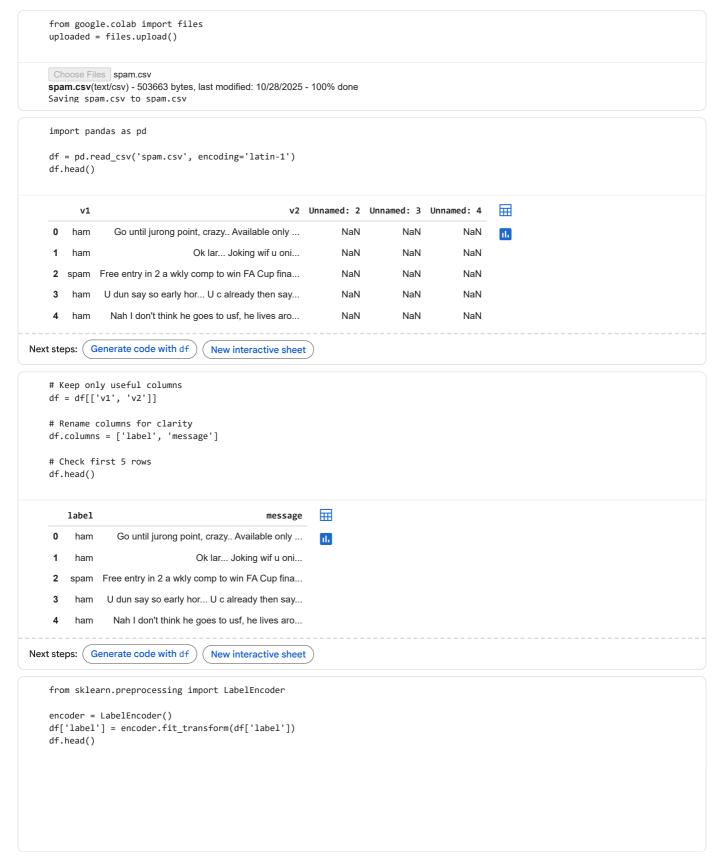
AICTE Internship – Shamgar Software Solutions

Task 2: AI & Machine Learning Internship (Batch 2 / Level 2)

Intern Name: Kuldeep Pandey
Problem Title: SMS Spam Detection

Objective: To build a machine learning model that can classify SMS messages as spam or not spam.

Tools & Technologies: Python, Pandas, Scikit-learn, TF-IDF Vectorizer, Logistic Regression Expected Outcome: The model accurately detects spam messages based on textual features. Result: Model trained successfully with high accuracy and correct manual test prediction.



```
label
                                                message
                                                           扁
           0
                 Go until jurong point, crazy.. Available only ...
    1
           0
                                 Ok lar... Joking wif u oni...
            1 Free entry in 2 a wkly comp to win FA Cup fina...
           0 U dun say so early hor... U c already then say...
    4
           0
                 Nah I don't think he goes to usf, he lives aro...
Next steps: ( Generate code with df )
                                    New interactive sheet
   from sklearn.model_selection import train_test_split
   X = df['message']
   y = df['label']
   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
   from sklearn.feature_extraction.text import TfidfVectorizer
   vectorizer = TfidfVectorizer(max features=3000)
    X_train_vectorized = vectorizer.fit_transform(X_train)
   X_test_vectorized = vectorizer.transform(X_test)
   from sklearn.linear_model import LogisticRegression
   model = LogisticRegression()
   model.fit(X_train_vectorized, y_train)
     ▼ LogisticRegression ① ?
    LogisticRegression()
    from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
   y_pred = model.predict(X_test_vectorized)
   print("☑ Accuracy:", accuracy_score(y_test, y_pred))
    \verb|print("\nConfusion Matrix:\n", confusion_matrix(y\_test, y\_pred))| \\
    print("\nClassification Report:\n", classification_report(y_test, y_pred))
    Accuracy: 0.968609865470852
   Confusion Matrix:
    [[964 1]
    [ 34 116]]
   Classification Report:
                   precision
                                recall f1-score
                                                    support
                       0.97
               0
                                 1.00
                                            0.98
                                                       965
                       0.99
                                 0.77
                                            0.87
                                                       150
               1
       accuracy
                                            0.97
                                                      1115
                       0.98
                                  0.89
      macro avg
                                            0.93
                                                       1115
   weighted avg
                       0.97
                                 0.97
                                            0.97
                                                      1115
    sample = ["Congratulations! You have won a free trip to Goa!"]
    sample_vectorized = vectorizer.transform(sample)
    print("Prediction:", "Spam" if model.predict(sample_vectorized)[0] == 1 else "Ham")
   Prediction: Spam
   import matplotlib.pyplot as plt
   # Count spam and ham messages
    counts = df['label'].value_counts()
    # Plot the bar chart
    plt.figure(figsize=(5,4))
```

plt.bar(['Ham (0)', 'Spam (1)'], counts, color=['green', 'red'])

