**Project Synopsis Report: Detecting Fake News with Python and Machine Learning**

**1. Introduction**

The proliferation of fake news on social media and other platforms has become a significant concern. Fake news can mislead people and influence public opinion. This project aims to build a machine learning model to classify news articles as real or fake using Python.

**2. Objectives**

* Develop a machine learning model to detect fake news.
* Understand and apply text processing techniques.
* Utilize machine learning algorithms for text classification.
* Evaluate the performance of the model.

**3. System Requirements**

**Hardware Requirements:**

* Processor: Intel Core i5 or higher
* RAM: 8GB or more
* Storage: 50GB free space
* OS: Windows 10 / macOS / Linux

**Software Requirements:**

* Python 3.6 or higher
* Jupyter Lab
* Libraries: numpy, pandas, sklearn

**4. Key Concepts and Tools**

**Fake News:** Misleading or false information presented as news, often spread via social media to influence opinions or political agendas.

**TfidfVectorizer:** Converts text data into a matrix of TF-IDF features, reflecting the importance of words in documents relative to the entire corpus.

**PassiveAggressive Classifier:** An online learning algorithm that updates its model in response to misclassifications, designed for large-scale learning tasks.

**5. Dataset**

**Dataset Name:** news.csv

**Description:**

* 7796 news articles.
* Columns: Identifier, Title, Text, Label (REAL or FAKE).
* Size: 29.2MB

**6. Project Steps**

**Step 1: Setup Environment**

1. Install Python and Jupyter Lab.
2. Install required libraries:

bash

1. pip install numpy pandas sklearn

**Step 2: Load and Preprocess Data**

1. Load the dataset using pandas.
2. Preprocess the data by handling missing values and converting text to lowercase.

**Step 3: Feature Extraction**

1. Use TfidfVectorizer to transform the text data into numerical features.

**Step 4: Model Building**

1. Initialize the PassiveAggressiveClassifier.
2. Split the data into training and testing sets.
3. Train the model on the training data.
4. Evaluate the model on the testing data using accuracy score and confusion matrix.

**Step 5: Evaluation**

1. Calculate the accuracy score to measure the percentage of correct predictions.
2. Generate a confusion matrix to evaluate the performance in terms of true positives, false positives, true negatives, and false negatives.

**7. Code Implementation**

**Loading and Preprocessing Data:**

python

import pandas as pd

# Load dataset

df = pd.read\_csv('news.csv')

# Check for missing values

df.isnull().sum()

# Drop missing values

df.dropna(inplace=True)

# Convert text to lowercase

df['text'] = df['text'].str.lower()

**Feature Extraction:**

python

from sklearn.feature\_extraction.text import TfidfVectorizer

# Initialize TfidfVectorizer

tfidf\_vectorizer = TfidfVectorizer(stop\_words='english', max\_df=0.7)

# Fit and transform the text data

tfidf\_train = tfidf\_vectorizer.fit\_transform(df['text'])

**Model Building and Training:**

python

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import PassiveAggressiveClassifier

from sklearn.metrics import accuracy\_score, confusion\_matrix

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(tfidf\_train, df['label'], test\_size=0.2, random\_state=7)

# Initialize PassiveAggressiveClassifier

pac = PassiveAggressiveClassifier(max\_iter=50)

# Train the model

pac.fit(X\_train, y\_train)

# Predict on the test set

y\_pred = pac.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

print(f'Confusion Matrix:\n{conf\_matrix}')

**8. Conclusion**

This project provides a comprehensive approach to detecting fake news using Python and machine learning. By implementing this project, you will learn how to preprocess text data, extract features using TF-IDF, and build a classification model using the PassiveAggressive Classifier. This project not only enhances your skills in text classification but also gives insight into practical applications of machine learning.