**Machine Learning Pipeline: Diabetes and Marketing Data Analysis**

***Author:*****Nishita Vikas Shewale, Trivikram Bhavesh Budhabhatti**

## **1. Overview**

This repository contains a Python script implementing a comprehensive machine learning pipeline for analyzing two datasets: **Diabetes** and **Marketing**. The pipeline includes data preprocessing, unsupervised labeling, feature extraction via PCA, and a Super Learner ensemble for classification.  
It leverages popular libraries like **NumPy**, **Pandas**, **Matplotlib**, **Seaborn**, and **Scikit-learn**.

## **2. Pipeline Components**

### 2.1. DataPreprocessor Class

* **Purpose**: Cleans and prepares data for modeling.
* **Functions**:
  + show\_outliers: Visualizes outliers using boxplots (before/after cleaning).
  + clean\_outliers: Removes outliers using z-score (threshold = 3).
  + fill\_missing: Imputes missing values using column means.
  + scale\_data: Standardizes features using StandardScaler.
  + preprocess: Executes all steps, tailored to “Medical Data” or “Marketing Data”.
* **Key Features**: Handles missing values, scales data, removes outliers with visualizations.

### 2.2. LabelMaker Class

* **Purpose**: Generates binary labels for the unsupervised Diabetes dataset.
* **Method**:
  + Applies KMeans (2 clusters) on Glucose, BMI, Age.
  + Assigns “Outcome” based on higher glucose mean.

### 2.3. FeatureExtractor Class

* **Purpose**: Prepares data using PCA and train-test splits.
* **Functions**:
  + split: Splits dataset (80% train, 20% test).
  + pca\_analysis: Applies PCA to reduce dimensions to 4 components.

### 2.4. SuperLearner Class

* **Purpose**: Ensemble model using stacking for classification.
* **Models**:
  + Base: GaussianNB, MLPClassifier, KNeighborsClassifier
  + Final: DecisionTreeClassifier
* **Functions**:
  + tune\_models: Hyperparameter tuning with GridSearchCV.
  + get\_preds: Generates meta-features via cross-validation.
  + train\_final: Trains final model on stacked outputs.
  + predict, check\_accuracy: Makes predictions and evaluates performance.

### 2.5. Encoder Class

* **Purpose**: Encodes categorical features in the Marketing dataset using LabelEncoder.

## **3. Workflow**

### 3.1. Diabetes Data

1. Load diabetes\_project.csv
2. Preprocess (clean, impute, scale)
3. Label using KMeans
4. PCA dimensionality reduction
5. Train & evaluate Super Learner

### 3.2. Marketing Data

1. Load Marketing.csv
2. Encode categorical columns
3. Preprocess numeric columns
4. Apply PCA and train Super Learner

## **4. Results**

* Cleaned data previews
* PCA-transformed shapes
* Tuned model parameters
* Accuracy scores for each datasets

## **5. Requirements**

* Python 3.x
* Libraries:
  + numpy
  + pandas
  + matplotlib
  + seaborn
  + scikit-learn
* Datasets: Place diabetes\_project.csv and Marketing.csv inside ./data\_file/

## **6. Installation**

bash

CopyEdit

git clone <repository-url>

cd <repo-folder>

pip install -r requirements.txt

## **7. Usage**

bash

CopyEdit

python main.py

## **8. Conclusion**

* Datasets must follow expected formats with columns like Glucose, BMI, successful\_marketing, etc.
* Modular design allows easy adaptation to other datasets with minimal changes.
* Accuracy scores where we found that from the given two datasets Accuracy for Marketing data is more than the Diabetes data

## **9. Future Use Case**

**Healthcare:**

* The diabetes prediction model could be used in clinical decision support systems to flag high-risk patients for early intervention.

**Marketing:**

* Predicting successful marketing campaign outcomes using client history.
* Identifying high-value clients for retention strategies.

**Smart Wearables & IoT:**

* Real-time classification of patient health status using live sensor data.
* Detecting anomalies in physical activity patterns (e.g., fall detection in elderly patients).

**Model Deployment and API Integration**

* Convert the trained model into a deployable REST API using Flask or FastAPI for real-time predictions in clinical or marketing environments.

## **10. GitHub Repository Link**

<https://github.com/trivikram-bhavesh-budhabhatti/Data-Science-Project>