ML-1 COMPREHENSIVE PROJECT

Problem Statement:

Diabetes prediction poses significant challenges in the healthcare sector, resulting in substantial health impacts and eroding patient trust.

Imagine you work for HealthSecure, a leading healthcare analytics firm specialising in predictive algorithms. With the rapid increase in digital health data, HealthSecure aims to develop a state-of-the-art system that can predict the onset of diabetes efficiently. Your task is to leverage machine learning to enhance the accuracy and reliability of HealthSecure's diabetes prediction capabilities

Dataset: Pima Indians Diabetes Database | Kaggle

Note: The dataset is sourced from the Kaggle platform, and the problem statement differs from the one on Kaggle but the idea is same. Please read the kaggle projects and its submission requirements to make submission on Kaggle platform also. Kaggle is great platform to show case on your resume and will increase your chances of hiring.

Upload it to the Newton School and Kaggle platforms, as HR will review it.

Objectives:

1. Load the Pima Indians Diabetes dataset and display the first few rows. How many features are there, and what is the distribution of target classes?
2. Are there any missing values in the dataset?
3. Provide visualizations to show the distribution of transactions over time. Draw:
   (a) Count plot for "Outcome"
   (b) Boxplots for each numerical feature
   (c) Pairplot
   (d) Histogram for all the continuous features
   (e) Feature Correlation Heatmap
4. Scale the data.
5. Implement a logistic regression model and compare the accuracy, precision, recall and F1 scores for it.
6. Use decision trees to classify and then tune the hyperparameters using RandomizedSearchCV. Use the parameters given below for RandomizedSeachCV:

```
'max_depth': [3, 5, 10, None],
'max_features': range(1, 11),
'min_samples_leaf': range(1, 5),
'criterion': ['gini', 'entropy']
```

7. Implement the k-nearest neighbors algorithm for the diabetes dataset and then print the accuracy, recall, and F1 score.

8. Create a Random Forest model and then, tune the hyperparameters using RandomizedSearchCV. After training the model, print the classification report. Use the parameters given below for RandomizedSeachCV:

```
'n_estimators': [50, 100, 200],
 'max_features': ['auto', 'sqrt', 'log2'],
 'max_depth': [4, 6, 8, 10, 12],
'criterion': ['gini', 'entropy'],
 'min_samples_split': [2, 5, 10],
'min_samples_leaf': [1, 2, 4]
```

9. Create a StackingClassifier using the KNN and Random Forest models.(StackingClassifier from the mlxtend library)