AI BASED DIABETES PREDICTION SYSTEM

Phase4: Development Part 2

Introduction:

The AI-based Diabetes Prediction System aims to provide accurate and timely predictions, which can assist healthcare professionals in making informed decisions regarding patient care and treatment plans. By identifying individuals at high risk of developing diabetes, preventive measures can be implemented, such as lifestyle modifications, dietary interventions, and regular screening.

Moreover, this system can also be utilized by individuals for self-assessment and awareness. By inputting their relevant information into the system, users can receive personalized risk assessments and recommendations for early intervention, empowering them to take proactive steps towards maintaining their health.

It is important to note that while the AI-based Diabetes Prediction System can provide valuable insights and predictions, it should not replace professional medical advice or diagnosis. It serves as a tool to support healthcare professionals and individuals in making informed decisions and taking preventive measures to mitigate the risk of diabetes.

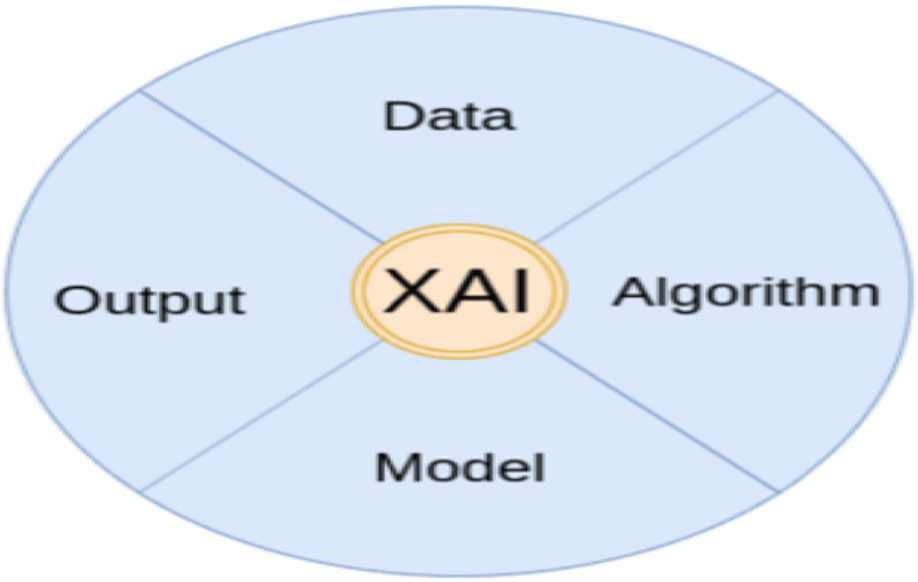
What type of Explanation do we need in Healthcare?

Data: It explains the data used for the prediction, their correlation, and EDA (Exploratory Data Analysis) to understand the hidden data patterns. It tells how the data is to be used for the AI system.

Algorithm: A complete transparency of the system's algorithm is given with the reason why the system chooses it and how it can be beneficial for the prediction.

Model: Akira AI gives a detailed explanation of model performance and working in a user-friendly manner.

Output: Akira AI gives a complete justification for the system's output with the reason. It also provides the factors that contribute to influence the result of the system.



Technologies:

1. Data Collection and Preprocessing:

Gather a diverse dataset that includes relevant features like age, gender, family history, lifestyle habits, and medical history. Preprocess the data by handling missing values, normalizing or scaling features, and encoding categorical variables.

2. Feature Engineering and Selection:

Analyze the dataset and extract meaningful features that are likely to be predictive for diabetes.Use techniques like correlation analysis, feature importance, or domain knowledge to select the most relevant features.

3. **Model Development:**

Choose a suitable machine learning algorithm(s) for diabetes prediction, such as logistic regression, decision trees, random forests, or neural networks. Train the model using the preprocessed dataset and the chosen algorithm(s).Optimize the model's hyperparameters using techniques like grid search or random search.

4. **Model Evaluation:**

Evaluate the trained model(s) using appropriate evaluation metrics like accuracy, precision, recall, F1-score, or area under the receiver operating characteristic curve (AUC-ROC). Use techniques like cross-validation to assess the model's generalization performance and identify potential issues like over fitting.

5. **Deployment and Integration:**

Develop a user interface or application where users can input their relevant information for prediction. Implement the trained model(s) within the system to generate predictions based on user inputs. Ensure the system provides accurate and reliable predictions in real-time.

6. **Continuous Improvement and Maintenance:**

Monitor the performance of the system over time and collect feedback for further improvements.Update the system periodically with new data and retrain the model(s) to incorporate the latest information. Stay updated with advancements in machine learning and diabetes research to enhance the system's effectiveness.

Coding:

```
import numpy as np

import pandas as pd

import seaborn as sns
```

```
import matplotlib.pyplot as plt

from collections import Counter

import os

from sklearn.metrics import confusion_matrix, accuracy_score,
precision_score

from sklearn.preprocessing import QuantileTransformer

from sklearn.linear_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier,
AdaBoostClassifier, GradientBoostingClassifier

from sklearn.model_selection import GridSearchCV, cross_val_score,
StratifiedKFold, learning_curve, train_test_split

from sklearn.svm import SVC

data =
pd.read_csv("../input/pima-indians-diabetes-database/diabetes.csv")

data['Glucose'] = data['Glucose'].replace(0, data['Glucose'].median())

data['BloodPressure'] = data['BloodPressure'].replace(0,
data['BloodPressure'].median())

data['BMI'] = data['BMI'].replace(0, data['BMI'].mean())

data['SkinThickness'] = data['SkinThickness'].replace(0,
data['SkinThickness'].mean())

data['Insulin'] = data['Insulin'].replace(0, data['Insulin'].mean())

plt.figure(figsize = (10, 8))

sns.heatmap(data.corr(), annot = True, fmt = ".3f", cmap = "YlGnBu")
```

```
plt.title("Correlation heatmap")

plt.figure(figsize = (10, 8))

kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 1], color =
"Red", shade = True)kde =
sns.kdeplot(data["Pregnancies"][data["Outcome"] == 0], ax = kde, color =
"Blue", shade= True)

kde.set_xlabel("Pregnancies")

kde.set_ylabel("Density")

kde.legend(["Positive Result", "Negative Result"])

def cv_model(models):

k_fold = StratifiedKFold(n_splits = 15)

r = []

for m in models :

r.append(cross_val_score(estimator = m, X = X_train, y = Y_train,
scoring = "accuracy", cv = k_fold, n_jobs = 4))

cross_val_means = []

cross_val_std = []

for result in r:

cross_val_means.append(result.mean())

cross_val_std.append(result.std())

df_result = pd.DataFrame({

"CrossValMean": cross_val_means,

"CrossValStd": cross_val_std,

"Model List":[
```

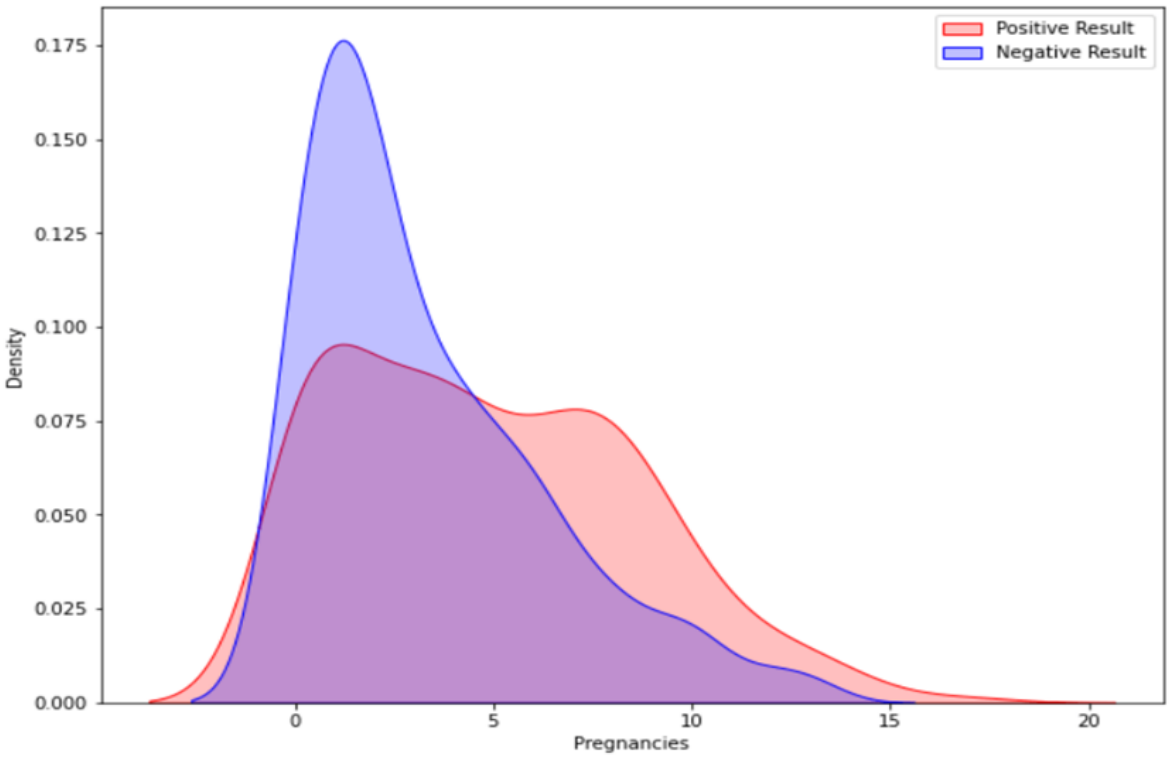
```
"DecisionTreeClassifier",

"LogisticRegression",

"SVC",

"AdaBoostClassifier",

"GradientBoostingClassifier",

"RandomForestClassifier",

"KNeighborsClassifier"

]

})

bar_plot = sns.barplot(x = cross_val_means, y = df_result["Model
List"].values, data = df_result)

bar_plot.set_xlabel("Mean of Cross Validation Accuracy Scores")

bar_plot.set_title("Cross Validation Scores of Models")

return df_result

tate = 20

models_list = [

DecisionTreeClassifier(random_state = state),

LogisticRegression(random_state = state, solver ='liblinear'),

SVC(random_state = random_state),

AdaBoostClassifier(DecisionTreeClassifier(random_state = state),
random_state = state, learning_rate = 0.3),

GradientBoostingClassifier(random_state = state),

RandomForestClassifier(random_state = state),

KNeighborsClassifier()
```

]

cv_model(models_list)

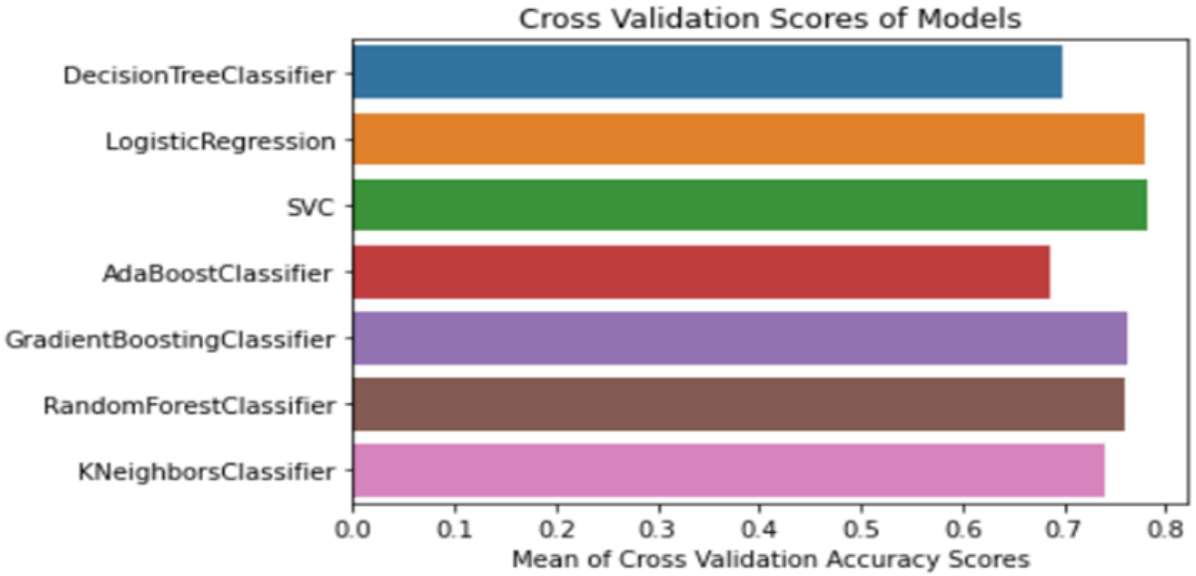
Output:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1



	CrossValMean	CrossValStd	Model List
0	0.697921	0.067773	DecisionTreeClassifier
1	0.780358	0.085376	LogisticRegression
2	0.782437	0.069578	SVC
3	0.686882	0.050551	AdaBoostClassifier
4	0.762796	0.072912	GradientBoostingClassifier
5	0.760717	0.079104	RandomForestClassifier
6	0.739283	0.043985	KNeighborsClassifier

Cross Validation Scores of Models



Conclusion:

Contribution of the Explainable AI in Diabetes Prediction system makes it easy for the end-user to understand the AI systems' complex working. It provides a human-centered interface to the user. Explainability is a key to producing a transparent, proficient, and accurate AI system that can help the healthcare practitioner, patients, and researcher understand and use the system.