**P a g e | 1**

****

**AI BASED DIABETES PREDICTION SYSTEM**

**PHASE4 DOCUMENT**

****

**DIABETES PREDICTION**

**Introduction:**

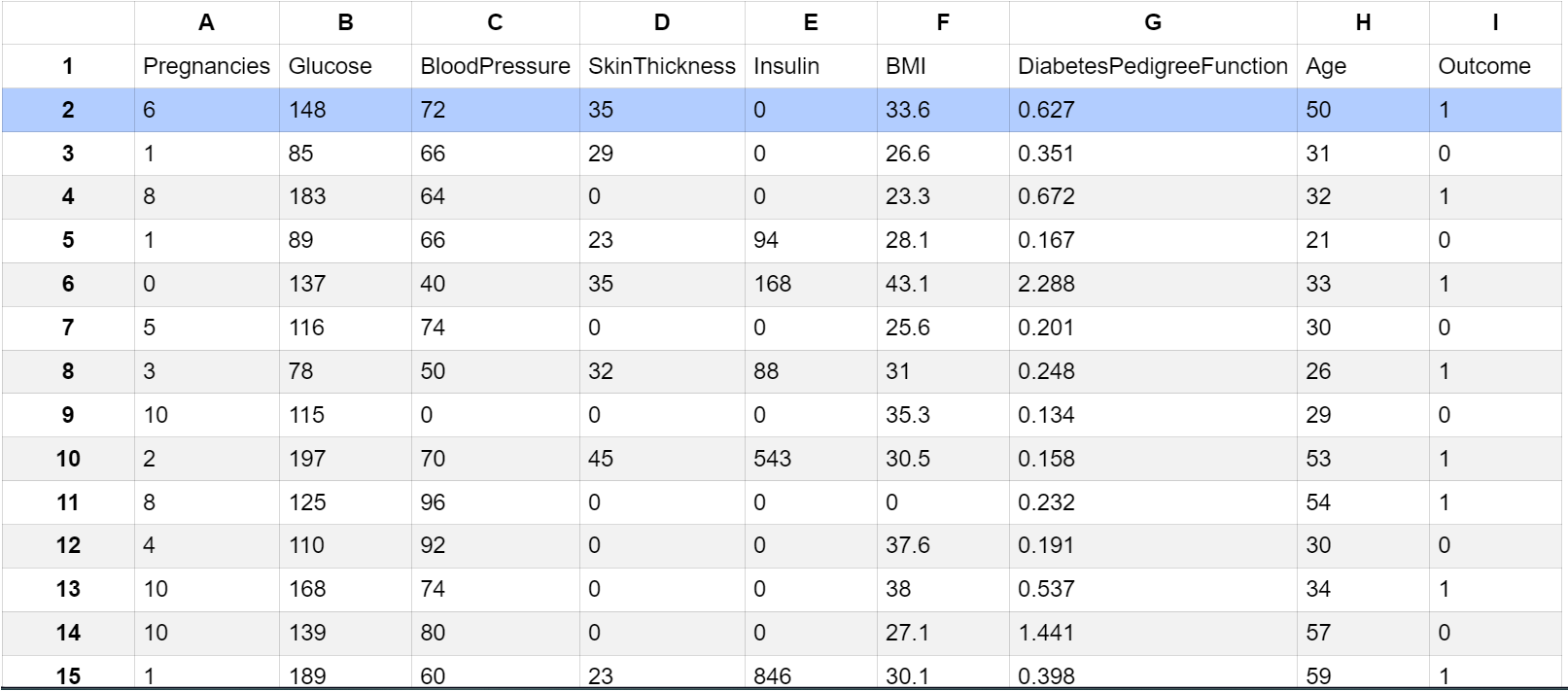
**Feature selection is the process of identifying and selecting the most relevant features from a dataset to improve the performance of a machine learning model. This is an important step in building a diabetes prediction model, as it can help to reduce overfitting and improve the generalization ability of the model.**

**Model training is the process of feeding the selected features to a machine learning algorithm and allowing it to learn the relationship between the features and the target variable . Once the model is trained, it can be used to predict the diabetes, given their features.**

**Model evaluation is the process of assessing the performance of a trained machine learning model on a held-out test set. This is important to ensure that the model is generalizing well and that it is not overfitting the training data.**

****

**Given data set:**



**Overview of the process:**

**The following is an overview of the process of building a diabetes prediction model by feature selection, model training, and evaluation:**

1. **Prepare the data: This includes cleaning the data, removing outliers, and handling missing values.**

**2.Perform feature selection: This can be done using a variety of methods, such as correlation analysis, information gain, and recursive feature elimination**

****

1. **Train the model: There are many different machine learning algorithms that can be used for diabetes prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.**
2. **Evaluate the model: This can be done by calculating the mean squared error (MSE) or the root mean squared error (RMSE) of the model's predictions on the held-out test set.**
3. **Deploy the model: Once the model has been evaluated and found to be performing well, it can be deployed to production so that it can be used to predict diabetes**

**PROCEDURE:**

**Feature selection:**

1. **Identify the target variable. This is the variable that you want to predict**
2. **Explore the data. This will help you to understand the relationships between the different features and the target variable. You can use data visualization and correlation analysis to identify features that are highly correlated with the target variable.**
3. **Remove redundant features. If two features are highly correlated with each other, then you can remove one of the features, as they are likely to contain redundant information.**
4. **Remove irrelevant features. If a feature is not correlated with the target variable, then you can remove it, as it is unlikely to be useful for prediction.**

******Model training:**

1. **Choose a machine learning algorithm. There are a number of different machine learning algorithms that can be used for diabetes prediction, such as linear regression, ridge regression, lasso regression, decision trees, and random forests are Covered above.**

**Machine Learning Models:**

**In [3]:**

**models =pd.DataFrame(columns=["Model","MAE","MSE","RMSE","R2 Score","RMSE (Cross-Validation)"])**

**Linear Regression:**

**In [4]:**

**lin\_reg =LinearRegression()**

**lin\_reg.fit(X\_train, y\_train)**

****

**predictions =lin\_reg.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

****

**print("MSE:", mse)**

****

**print("RMSE:", rmse)**

****

**print("R2 Score:", r\_squared)**

****

**print("-"\*30)rmse\_cross\_val =rmse\_cv(lin\_reg)**

****

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "LinearRegression","MAE": mae, "MSE": mse, "RMSE": rmse, "R2 Score": r\_squared, "RMSE (Cross-Validation)": rmse\_cross\_val}models =models.append(new\_row, ignore\_index=True)**

**Out[4]:**

**MAE: 23567.890565943395**

**MSE: 1414931404.6297863**

**RMSE: 37615.57396384889**

**R2 Score: 0.8155317822983865**

**------------------------------**

**RMSE Cross-Validation: 36326.451444669496**

**Ridge Regression:**

**In [5]:**

****

**ridge =Ridge()ridge.fit(X\_train, y\_train)predictions =ridge.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

****

**print("MSE:", mse)**

****

**print("RMSE:", rmse)**

****

**print("R2 Score:", r\_squared)**

****

**print("-"\*30)rmse\_cross\_val =rmse\_cv(ridge)**

****

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "Ridge","MAE": mae, "MSE": mse, "RMSE": rmse,**

**"R2 Score": r\_squared, "RMSE (Cross-Validation)": rmse\_cross\_val}model**

* **=models.append(new\_row, ignore\_index=True)Out[5]:**

**MAE: 23435.50371200822**

**MSE: 1404264216.8595588**

**RMSE: 37473.513537691644**

**R2 Score: 0.8169224907874508**

**------------------------------**

**RMSE Cross-Validation: 35887.852791598336**

**Lasso Regression:**

**In [6]:**

****

**lasso =Lasso()lasso.fit(X\_train, y\_train)predictions =lasso.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

****

**print("MSE:", mse)**

****

**print("RMSE:", rmse)**

****

**print("R2 Score:", r\_squared)**

****

**print("-"\*30)rmse\_cross\_val =rmse\_cv(lasso)**

****

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "Lasso","MAE": mae, "MSE": mse, "RMSE": rmse,**

**"R2 Score": r\_squared, "RMSE (Cross-Validation)": rmse\_cross\_val}model**

* **=models.append(new\_row, ignore\_index=True)Out[6]:**

**MAE: 23560.45808027236**

**MSE: 1414337628.502095**

**RMSE: 37607.680445649596**

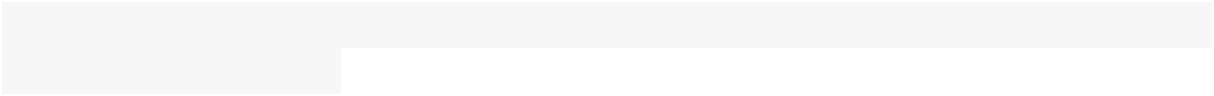
**R2 Score: 0.815609194407292**

**------------------------------**

**RMSE Cross-Validation: 35922.76936876075**

**Elastic Net:**

**In [7]:**

****

**elastic\_net =ElasticNet()elastic\_net.fit(X\_train, y\_train)predictions =elastic\_net.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

****

**print("MSE:", mse)**

****

**print("RMSE:", rmse)**

****

**print("R2 Score:", r\_squared)**

****

**print("-"\*30)rmse\_cross\_val =rmse\_cv(elastic\_net)**

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "ElasticNet","MAE": mae, "MSE": mse, "RMSE": r mse, "R2 Score": r\_squared, "RMSE (Cross-Validation)": rmse\_cross\_val} models =models.append(new\_row, ignore\_index=True)**

**Out[7]:**

**MAE: 23792.743784996732**

**MSE: 1718445790.1371393**

**RMSE: 41454.14080809225**

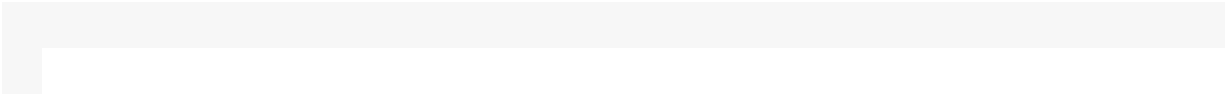
**R2 Score: 0.775961837382229**

**------------------------------**

**RMSE Cross-Validation: 38449.00864609558**

**Support Vector Machines:**

**In [8]:**

****

**svr = SVR(C=100000)svr.fit(X\_train, y\_train)predictions =svr.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

****

**print("MSE:", mse)**

****

**print("RMSE:", rmse)**

****

**print("R2 Score:", r\_squared)**

**print("-"\*30)rmse\_cross\_val =rmse\_cv(svr)**

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "SVR","MAE": mae, "MSE": mse, "RMSE": rmse, "**

**R2 Score": r\_squared, "RMSE (Cross-Validation)": rmse\_cross\_val}models**

* **models.append(new\_row, ignore\_index=True) Out[9]:**

**MAE: 17843.16228084976**

**MSE: 1132136370.3413317**

**RMSE: 33647.234215330864**

**R2 Score: 0.852400492526574**

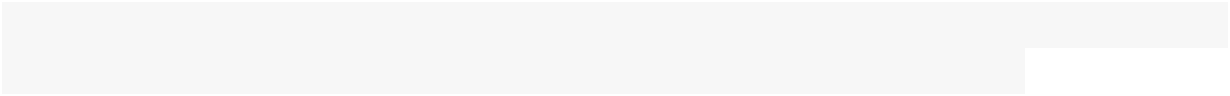
**------------------------------**

**RMSE Cross-Validation: 30745.475239075837**

****

**Random Forest Regressor:**

**In [9]:**

****

**random\_forest =RandomForestRegressor(n\_estimators=100)random\_forest.fit(X\_train, y\_train)predictions =random\_forest.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

**print("MSE:", mse)**

**print("RMSE:", rmse)**

**print("R2 Score:", r\_squared)**

**print("-"\*30)rmse\_cross\_val =rmse\_cv(random\_forest)**

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "RandomForestRegressor","MAE": mae, "MSE": ms e, "RMSE": rmse, "R2 Score": r\_squared, "RMSE (Cross-Validation)": rms e\_cross\_val}models =models.append(new\_row, ignore\_index=True)**

**Out[9]:**

**MAE: 18115.11067351598**

**MSE: 1004422414.0219476**

**RMSE: 31692.623968708358**

**R2 Score: 0.869050886899595**

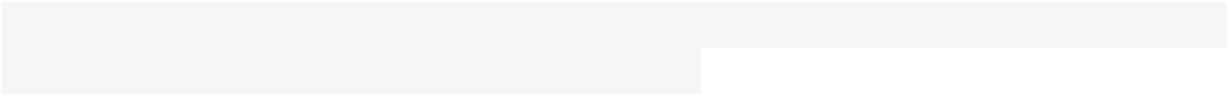
**------------------------------**

**RMSE Cross-Validation: 31138.863315259332**

****

**XGBoost Regressor:**

**In [10]:**

****

**xgb =XGBRegressor(n\_estimators=1000, learning\_rate=0.01)xgb.fit(X\_trai n, y\_train)predictions =xgb.predict(X\_test)**

****

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

****

**print("MAE:", mae)**

**print("MSE:", mse)**

**print("RMSE:", rmse)**

**print("R2 Score:", r\_squared)**

**print("-"\*30)rmse\_cross\_val =rmse\_cv(xgb)**

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "XGBRegressor","MAE": mae, "MSE": mse, "RMSE": rmse, "R2 Score": r\_squared, "RMSE (Cross-Validation)": rmse\_cross\_val}models =models.append(new\_row, ignore\_index=True)**

**Out[10]:**

**MAE: 17439.918396832192**

**MSE: 716579004.5214689**

**RMSE: 26768.993341578403**

**R2 Score: 0.9065777666861116**

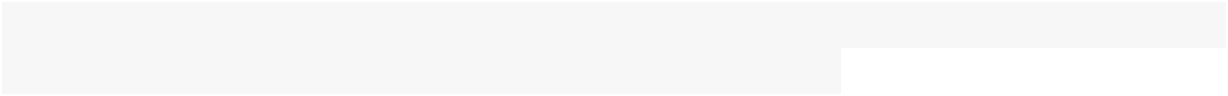
**------------------------------**

**RMSE Cross-Validation: 29698.84961808251**

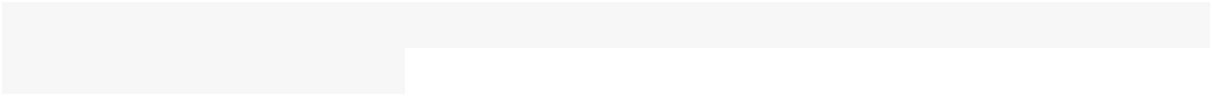
****

**Polynomial Regression (Degree=2)**

**In [11]:**

****

**poly\_reg =PolynomialFeatures(degree=2)X\_train\_2d =poly\_reg.fit\_transfo rm(X\_train)X\_test\_2d =poly\_reg.transform(X\_test)**

****

**lin\_reg =LinearRegression()lin\_reg.fit(X\_train\_2d, y\_train)predictions = li n\_reg.predict(X\_test\_2d)**

**mae, mse, rmse, r\_squared =evaluation(y\_test, predictions)**

**print("MAE:", mae)**

**print("MSE:", mse)**

**print("RMSE:", rmse)**

**print("R2 Score:", r\_squared)**

**print("-"\*30)rmse\_cross\_val =rmse\_cv(lin\_reg)**

**print("RMSE Cross-Validation:", rmse\_cross\_val)**

**new\_row = {"Model": "Polynomial Regression (degree=2)","MAE": mae, "**

**MSE": mse, "RMSE": rmse, "R2 Score": r\_squared, "RMSE (Cross-Validat ion)": rmse\_cross\_val}models =models.append(new\_row, ignore\_index=True)**

****

**Out[11]:**

**MAE: 2382228327828308.5**

**MSE: 1.5139911544182342e+32**

**RMSE: 1.230443478758059e+16**

****

**R2 Score: -1.9738289005226644e+22**

**------------------------------**

**RMSE Cross-Validation: 36326.451444669496**

**Model training:**

**Model training is the process of teaching a machine learning model to predict diabetes.**

**Once the model is trained, it can be used to predict diabetes for new data. For example, you could use the model to predict diabetes.**

1. **Prepare the data. This involves cleaning the data, removing any errors or inconsistencies, and transforming the data into a format that is compatible with the machine learning algorithm that you will be using.**
2. **Split the data into training and test sets. The training set will be used to train the model, and the test set will be used to evaluate**

**the performance of the model on unseen data.**

1. **Choose a machine learning algorithm. There are a number of different machine learning algorithms that can be used for diabetes prediction, such as linear regression, ridge regression, lasso regression, decision trees, and random forests.**
2. **Tune the hyperparameters of the algorithm. The hyperparameters of a machine learning algorithm are parameters that control the learning process. It is important to tune the hyperparameters of the algorithm to optimize its performance.**

****

1. **Train the model on the training set. This involves feeding the training data to the model .**
2. **Evaluate the model on the test set. This involves feeding the test data to the model and measuring how well it predicts the diabetes.**

**If the model performs well on the test set, then you can be confident that it will generalize well to new data.**

**Model evaluation:**

1. **Calculate the evaluation metrics. There are a number of different evaluation metrics that can be used to assess the performance of a machine learning model, such as R-squared, mean squared error (MSE), and root mean squared error (RMSE).**
2. **Interpret the evaluation metrics. The evaluation metrics will give you an idea of how well the model is performing on unseen data. If the model is performing well, then you can be confident that it will generalize well to new data. However, if the model is performing poorly, then you may need to try a different model or retune the hyperparameters of the current model.**

****

**Model evaluation:**

**Model evaluation is the process of assessing the performance of a machine learning model on unseen data. This is important to ensure that the model will generalize well to new data.**

**There are a number of different metrics that can be used to evaluate the performance of diabetes prediction model. Some of the most common metrics include:**

**Mean squared error (MSE): This metric measures the average squared difference between the predicted and normal .**

**Root mean squared error (RMSE): This metric is the square root of the MSE.**

**Mean absolute error (MAE): This metric measures the average absolute difference between the predicted and normal.**

**R-squared: This metric measures how well the model explains the variation in the normal.**

**In addition to these metrics, it is also important to consider the following factors when evaluating a diabetes prediction model:**

**Bias: Bias is the tendency of a model to consistently over- or underestimate diabetes prediction**

**Variance: Variance is the measure of how much the predictions of a model.**

**Interpretability: Interpretability is the ability to understand how the model makes its predictions. This is important for diabetes prediction models, as it allows users to understand the factors that influence the predicted diabetes.**

**************Various feature to perform model training:**

**Use a variety of feature engineering techniques.**

**Feature engineering is the process of transforming raw data into features that are more informative and predictive for machine learning models. By using a variety of feature engineering techniques, you can create a set of features that will help your model to predict house prices more accurately.**

**Use cross-validation.**

**Cross-validation is a technique for evaluating the performance of a machine learning model on unseen data. It is important to use cross-validation to evaluate the performance of your model during the training process. This will help you to avoid overfitting and to ensure that your model will generalize well to new data.**

**Use ensemble methods.**

**Ensemble methods are machine learning methods that combine the predictions of multiple models to produce a more accurate prediction. Ensemble methods can often achieve better performance than individual machine learning models.**

****

**Use cross-validation.**

**Cross-validation is a technique for evaluating the performance of a machine learning model on unseen data. It is important to use cross-validation to evaluate the performance of your model during the evaluation process. This will help you to avoid overfitting and to ensure that the model will generalize well to new data.**

**It is a set of data that is not used to train or evaluate the model during the training process. This data is used to evaluate the performance of the model on unseen data after the training process is complete.**

**Compare the model to a baseline.**

**A baseline is a simple model that is used to compare the performance of your model to.**

**Analyze the model's predictions.**

**Once you have evaluated the performance of the model, you can analyze the model's predictions to identify any patterns or biases. This will help you to understand the strengths and weaknesses of the model and to improve it.**

****