MACHINE LEARNING

FEATURE SELECTION

NIKKI EVANA BLESSY.N

2341459

Implement KNN regressor to predict the

Plasma glucose concentration. Select the top 5 features from the original dataset. Implement at least two techniques for feature selection.

```
[14] # Import necessary libraries
     import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.feature_selection import SelectKBest, f_regression
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.neighbors import KNeighborsRegressor
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import mean_squared_error
```

```
     # Import necessary libraries
     import pandas as pd
     from google.colab import files

     uploaded = files.upload()

     data = pd.read_csv("pima-indians-diabetes.csv")
```

```
data.columns = [
    "Pregnancies",
    "Glucose",
    "BloodPressure",
    "SkinThickness",
    "Insulin",
    "BMI",
    "DiabetesPedigreeFunction",
    "Age",
    "Outcome",
]
```

```
[16] # Feature Selection
     X = data.drop(columns=["Glucose"])
     y = data["Glucose"]
```

```
[17] #Correlation-based feature selection (1)
     correlation = data.corr()["Glucose"].sort_values(ascending=False)
     top_corr_features = correlation.index[1:6]
```

```
# SelectKBest based on F-test (2)
select_k_best = SelectKBest(score_func=f_regression, k=5)
select_k_best.fit(X, y)
selected_features = X.columns[select_k_best.get_support()]
```

```
[19] # Union of features from both methods
     important_features = list(set(top_corr_features).union(set(selected_features)))
```

```
[20] # Train-Test Split using important features
     X_important = data[important_features]
     X_train, X_test, y_train, y_test = train_test_split(X_important, y, test_size=0.2, random_state=42)
```

```
# Implement KNN Regressor
knn_regressor = KNeighborsRegressor(n_neighbors=5)  # Default is 5 neighbors
knn_regressor.fit(X_train, y_train)
```

```
KNeighborsRegressor
KNeighborsRegressor()
```

```
[22] # Predictions and Evaluation
     y_pred = knn_regressor.predict(X_test)
     mse = mean_squared_error(y_test, y_pred)
     rmse = np.sqrt(mse)
```

```
[23] # Display Results
     print("Selected Features:", important_features)
     print("Root Mean Squared Error (RMSE):", rmse)
```

```
Selected Features: ['Outcome', 'Insulin', 'BMI', 'Age', 'BloodPressure']
Root Mean Squared Error (RMSE): 30.81270609899197
```

```
# Visualization (Feature Importance from F-test and Correlation)
plt.figure(figsize=(12, 6))
plt.bar(important_features, select_k_best.scores_[select_k_best.get_support()], color="skyblue")
plt.title("Feature Importance (F-test)", fontsize=14)
plt.ylabel("F-Score")
plt.xticks(rotation=45)
plt.show()
```