Practical 01

Aim:

To build a logistic regression model to predict whether a person has diabetes using the Pima Indians Diabetes dataset and to compare the model performance before and after selecting the top 5 most important features using feature selection techniques.

Theory:

Feature Selection and Filtering

Feature selection is the process of identifying and selecting the most relevant features (variables) from a dataset that contribute the most to predicting the target variable. This reduces model complexity, speeds up training, and can improve model performance. Common methods include:

- **Filter Methods** (e.g., SelectKBest)
- **Wrapper Methods** (e.g., Recursive Feature Elimination)
- **Embedded Methods** (e.g., feature importance from tree models)

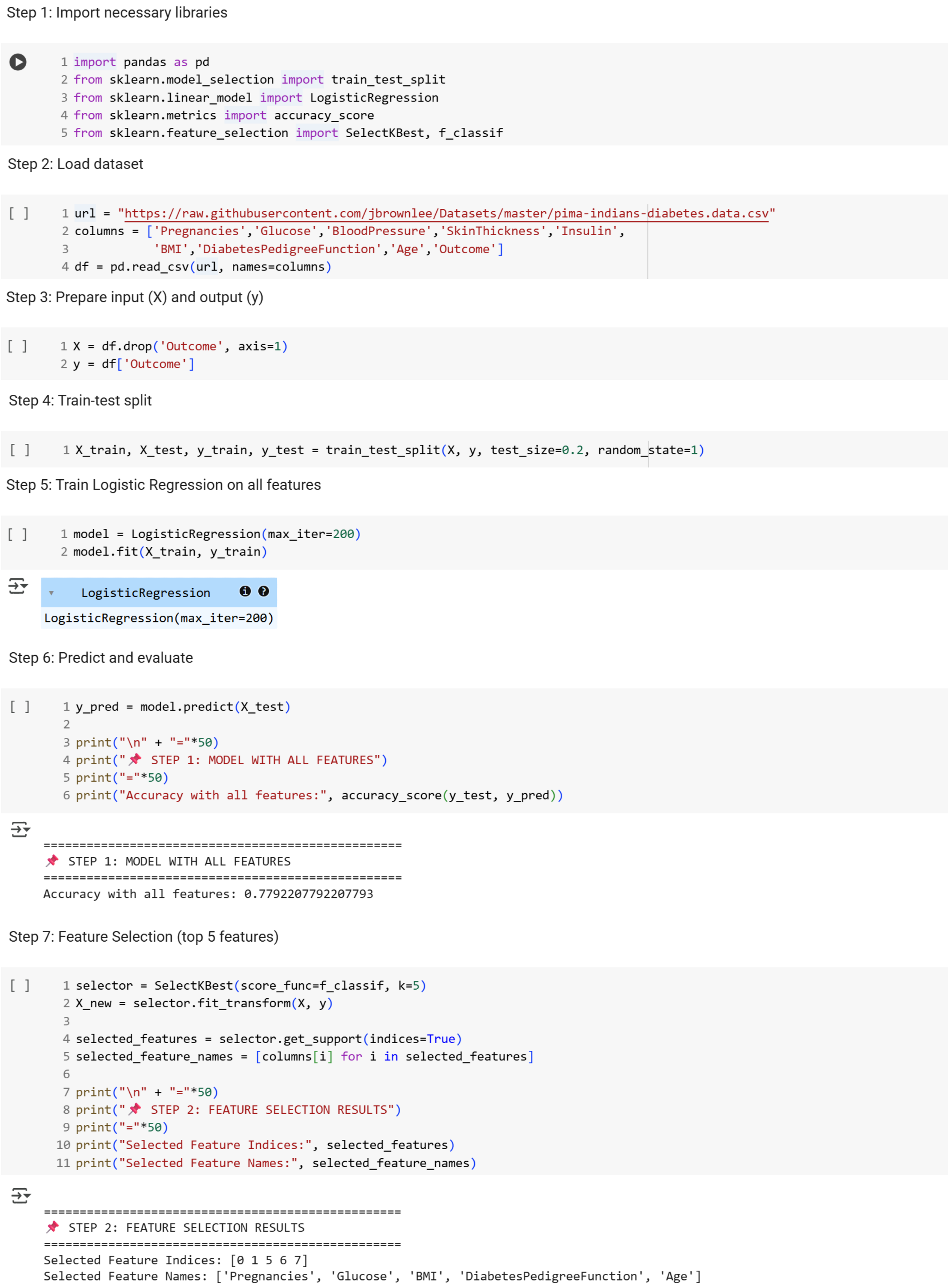
In this practical, we use **SelectKBest** with the **f_classif** scoring function, which uses ANOVA F-statistics to select the features most associated with the target outcome.

Logistic Regression Model

Logistic Regression is a supervised machine learning algorithm used for classification problems. It predicts the probability of a categorical dependent variable (binary outcome).
Key points:

- It uses the logistic (sigmoid) function to map predicted values between 0 and 1.
- Suitable for binary classification (e.g., diabetes: yes/no).
- The decision boundary is based on probability thresholds (typically 0.5).

Code:



Step 1: Import necessary libraries

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.feature_selection import SelectKBest, f_classif
```

Step 2: Load dataset

```python
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.data.csv"
columns = ['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin',
           'BMI','DiabetesPedigreeFunction','Age','Outcome']
df = pd.read_csv(url, names=columns)
```

Step 3: Prepare input (X) and output (y)

```python
X = df.drop('Outcome', axis=1)
y = df['Outcome']
```

Step 4: Train-test split

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1)
```

Step 5: Train Logistic Regression on all features

```python
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression(max_iter=200)
```

Step 6: Predict and evaluate

```python
y_pred = model.predict(X_test)

print("\n" + "="*50)
print("📌 STEP 1: MODEL WITH ALL FEATURES")
print("="*50)
print("Accuracy with all features:", accuracy_score(y_test, y_pred))
```

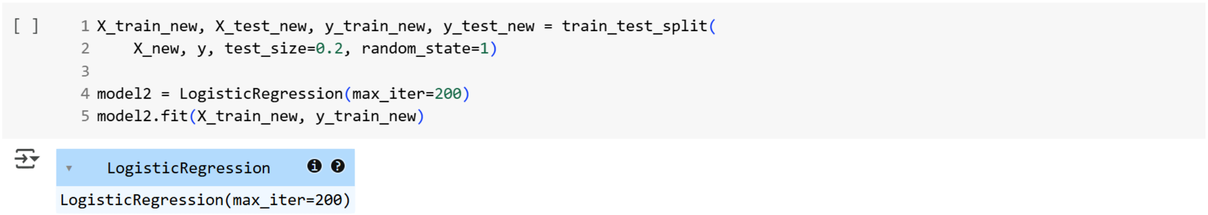
```

==================================================
📌 STEP 1: MODEL WITH ALL FEATURES
==================================================
Accuracy with all features: 0.7792207792207793
```

Step 7: Feature Selection (top 5 features)

```python
selector = SelectKBest(score_func=f_classif, k=5)
X_new = selector.fit_transform(X, y)

selected_features = selector.get_support(indices=True)
selected_feature_names = [columns[i] for i in selected_features]

print("\n" + "="*50)
print("📌 STEP 2: FEATURE SELECTION RESULTS")
print("="*50)
print("Selected Feature Indices:", selected_features)
print("Selected Feature Names:", selected_feature_names)
```

```

==================================================
📌 STEP 2: FEATURE SELECTION RESULTS
==================================================
Selected Feature Indices: [0 1 5 6 7]
Selected Feature Names: ['Pregnancies', 'Glucose', 'BMI', 'DiabetesPedigreeFunction', 'Age']
```

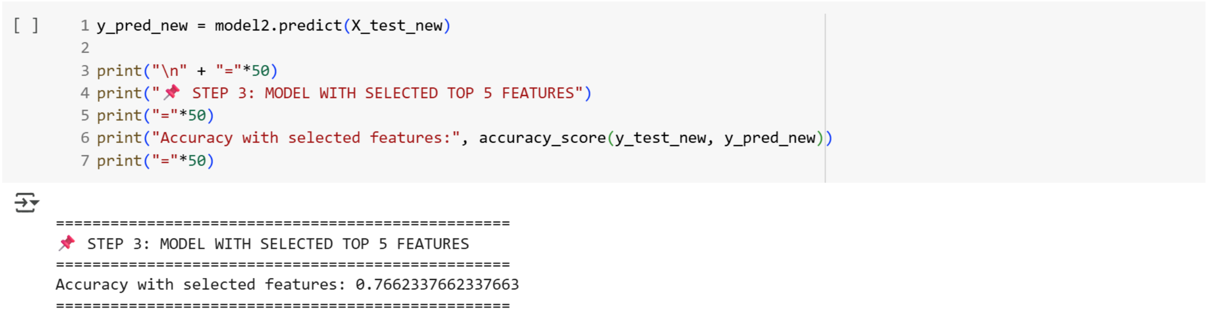
Step 8: Train model with selected features



```
X_train_new, X_test_new, y_train_new, y_test_new = train_test_split(
    X_new, y, test_size=0.2, random_state=1)

model2 = LogisticRegression(max_iter=200)
model2.fit(X_train_new, y_train_new)
```

```
LogisticRegression
LogisticRegression(max_iter=200)
```

Step 9: Evaluate new model

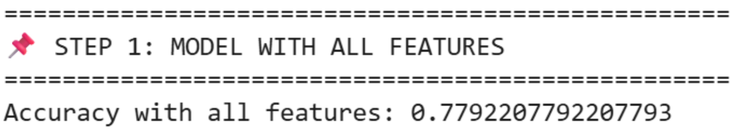


```
y_pred_new = model2.predict(X_test_new)

print("\n" + "="*50)
print("📌 STEP 3: MODEL WITH SELECTED TOP 5 FEATURES")
print("="*50)
print("Accuracy with selected features:", accuracy_score(y_test_new, y_pred_new))
print("="*50)
```

```
==================================================
📌 STEP 3: MODEL WITH SELECTED TOP 5 FEATURES
==================================================
Accuracy with selected features: 0.7662337662337663
==================================================
```

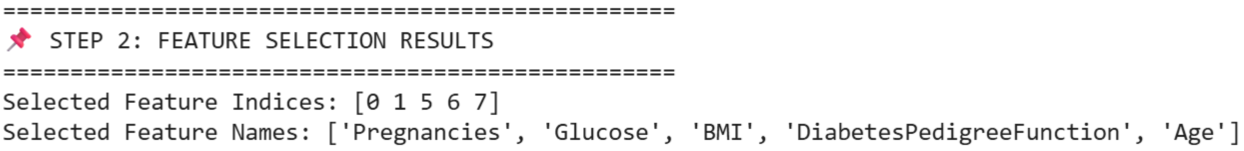
Output Screenshots:

- **First accuracy (before feature selection)**



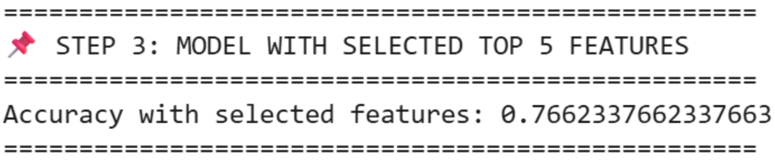
```
==================================================
📌 STEP 1: MODEL WITH ALL FEATURES
==================================================
Accuracy with all features: 0.7792207792207793
```

- **List of selected features**



```
==================================================
📌 STEP 2: FEATURE SELECTION RESULTS
==================================================
Selected Feature Indices: [0 1 5 6 7]
Selected Feature Names: ['Pregnancies', 'Glucose', 'BMI', 'DiabetesPedigreeFunction', 'Age']
```

- **Second accuracy (after feature selection)**



```
==================================================
📌 STEP 3: MODEL WITH SELECTED TOP 5 FEATURES
==================================================
Accuracy with selected features: 0.7662337662337663
==================================================
```

Conclusion:

From the above practical, we successfully built a logistic regression model to predict diabetes using all features and then after applying feature selection using SelectKBest.