Republic of the Philippines
Laguna State Polytechnic University
Province of Laguna

Exercise No. 1

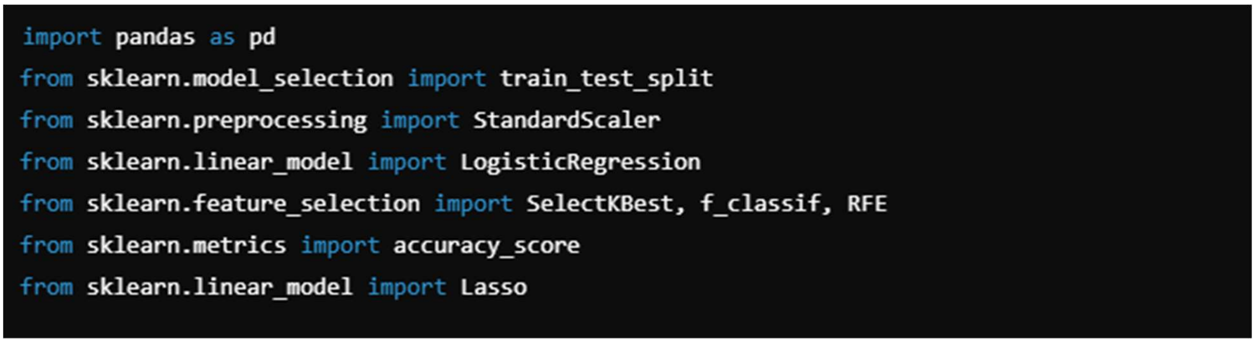
Topic:	**Topic 2: Feature Selection**	**Week No.**	3
Course Code:	**CSS104**	**Term:**	2nd Semester
Course Title:	**Advance Machine Learning**	**Academic Year:**	2024-2025
Student Name		**Section**	
Due date		**Points**	

Feature Selection

This notebook demonstrates how to apply feature selection techniques to improve a machine learning model using the Iris dataset. The goal is to evaluate the performance of the model with different feature selection methods and compare their accuracy.

Let's break down the key components of the code:

1. Import Libraries

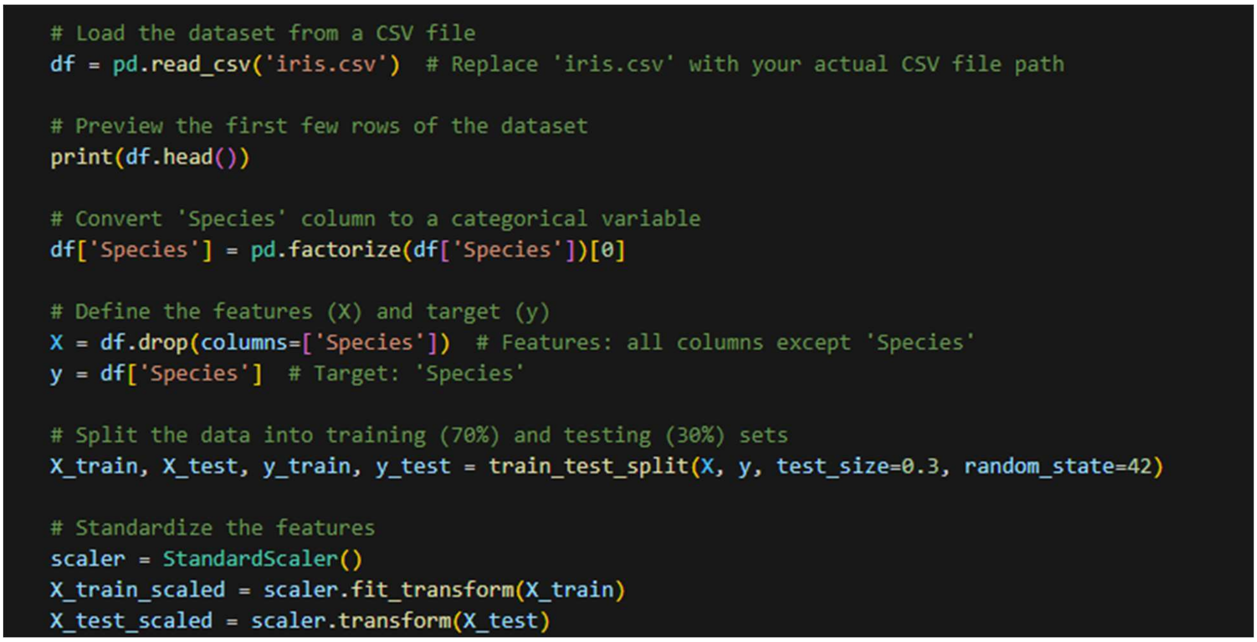


```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import SelectKBest, f_classif, RFE
from sklearn.metrics import accuracy_score
from sklearn.linear_model import Lasso
```

Here, we import necessary libraries:

- **pandas:** Used for handling and displaying data.
- **sklearn.datasets:** Contains pre-built datasets like the Iris dataset.
- **train_test_split:** Used to split the dataset into training and testing sets.
- **StandardScaler:** Standardizes the features (scaling them to have zero mean and unit variance).
- **LogisticRegression:** The machine learning model we'll use to classify the data.
- **SelectKBest:** A feature selection method that selects the best k features.
- **f_classif:** A statistical test used with SelectKBest (ANOVA F-value).
- **RFE:** Recursive Feature Elimination, a wrapper method for feature selection.
- **accuracy_score:** Used to evaluate the model's performance.
- **Lasso:** A linear regression technique that performs feature selection via L1 regularization.

2. **Load and Preprocess the Data**



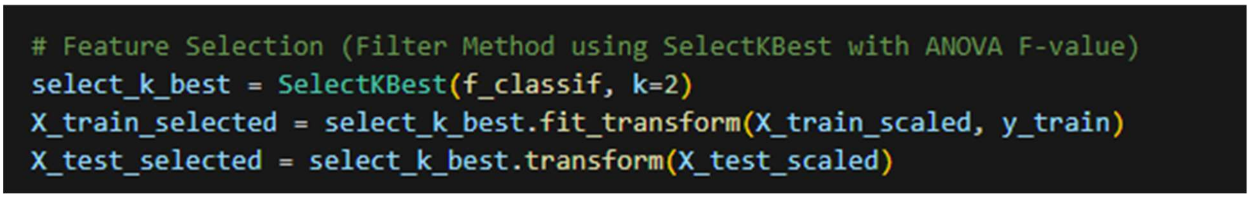
```
# Load the dataset from a CSV file
df = pd.read_csv('iris.csv')  # Replace 'iris.csv' with your actual CSV file path

# Preview the first few rows of the dataset
print(df.head())

# Convert 'Species' column to a categorical variable
df['Species'] = pd.factorize(df['Species'])[0]

# Define the features (X) and target (y)
X = df.drop(columns=['Species'])  # Features: all columns except 'Species'
y = df['Species']  # Target: 'Species'

# Split the data into training (70%) and testing (30%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

- **Loading the Iris dataset:** We load the Iris dataset, which contains features like the length and width of petals and sepals to classify iris flowers into three categories.
- **Splitting the data:** The dataset is split into training (70%) and testing (30%) sets.
- **Standardizing the features:** We scale the features (X-values) so that all features have a mean of 0 and a standard deviation of 1. This is crucial for certain feature selection methods and models like logistic regression.

3. Feature Selection

We apply **three different feature selection methods**:

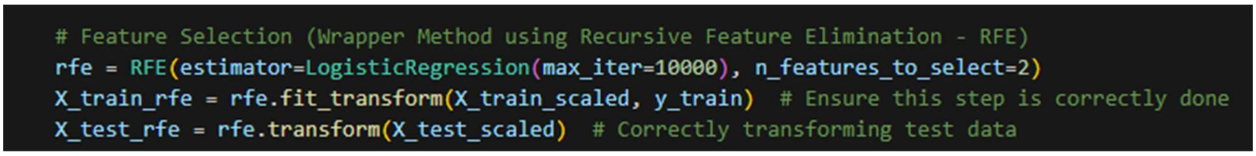
Filter Method (ANOVA F-value)



```
# Feature Selection (Filter Method using SelectKBest with ANOVA F-value)
select_k_best = SelectKBest(f_classif, k=2)
X_train_selected = select_k_best.fit_transform(X_train_scaled, y_train)
X_test_selected = select_k_best.transform(X_test_scaled)
```

- **SelectKBest** selects the **k** best features from the dataset based on a statistical test (in this case, ANOVA F-value with **f_classif**).
- We select **k=2** features (top 2) based on the highest F-value scores.

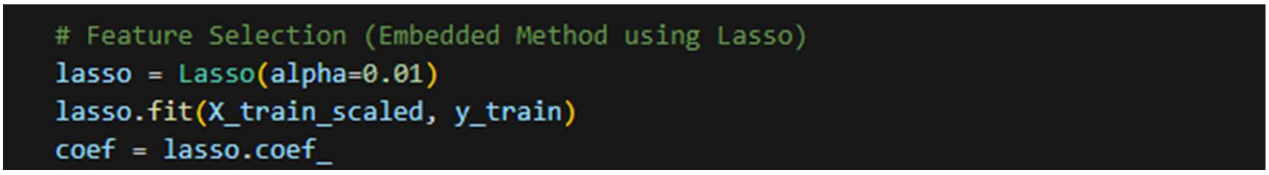
Wrapper Method (Recursive Feature Elimination - RFE)



```
# Feature Selection (Wrapper Method using Recursive Feature Elimination - RFE)
rfe = RFE(estimator=LogisticRegression(max_iter=10000), n_features_to_select=2)
X_train_rfe = rfe.fit_transform(X_train_scaled, y_train)  # Ensure this step is correctly done
X_test_rfe = rfe.transform(X_test_scaled)  # Correctly transforming test data
```

- **RFE** eliminates features recursively by training the model (in this case, logistic regression) and removing the least significant feature at each step.
- We choose to keep 2 features (**n_features_to_select=2**) after recursively eliminating the rest.

Embedded Method (Lasso Regression)

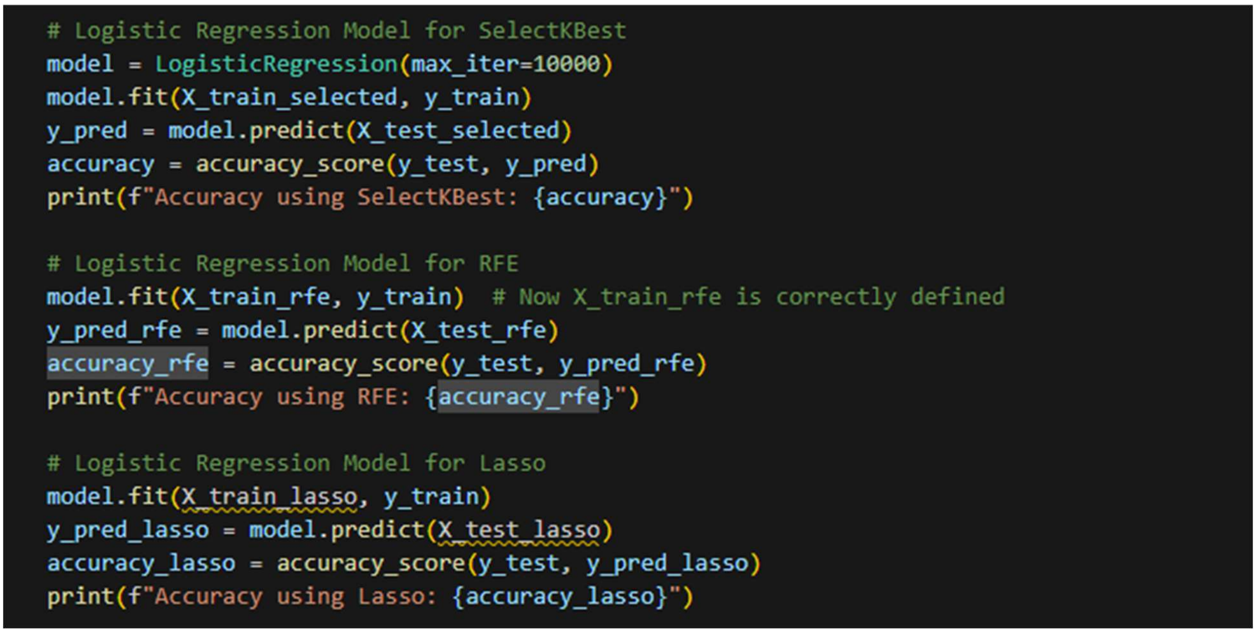


```
# Feature Selection (Embedded Method using Lasso)
lasso = Lasso(alpha=0.01)
lasso.fit(X_train_scaled, y_train)
coef = lasso.coef_
```

- **Lasso** is a linear regression model that includes L1 regularization, which forces some of the model coefficients to become exactly zero. These coefficients represent the least important features, which are effectively "dropped."
- The features with non-zero coefficients are selected.

4. Model Training and Evaluation

We train a **Logistic Regression** model for each feature selection method and evaluate its accuracy on the test set.



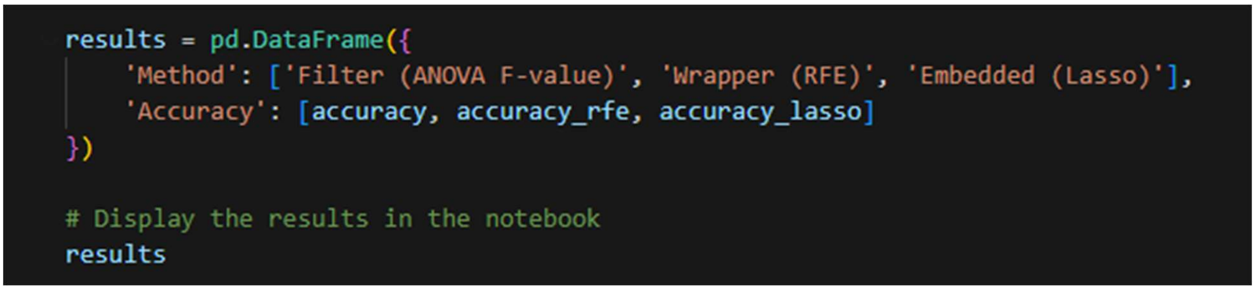
```
# Logistic Regression Model for SelectKBest
model = LogisticRegression(max_iter=10000)
model.fit(X_train_selected, y_train)
y_pred = model.predict(X_test_selected)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy using SelectKBest: {accuracy}")

# Logistic Regression Model for RFE
model.fit(X_train_rfe, y_train)  # Now X_train_rfe is correctly defined
y_pred_rfe = model.predict(X_test_rfe)
accuracy_rfe = accuracy_score(y_test, y_pred_rfe)
print(f"Accuracy using RFE: {accuracy_rfe}")

# Logistic Regression Model for Lasso
model.fit(X_train_lasso, y_train)
y_pred_lasso = model.predict(X_test_lasso)
accuracy_lasso = accuracy_score(y_test, y_pred_lasso)
print(f"Accuracy using Lasso: {accuracy_lasso}")
```

For each method:

- We **train** the model using the selected features.
- We **predict** the labels on the test data **(X_test)** and calculate the **accuracy** of the model

5. Comparison of Methods

After evaluating each method, we display the results:



```
results = pd.DataFrame({
    'Method': ['Filter (ANOVA F-value)', 'Wrapper (RFE)', 'Embedded (Lasso)'],
    'Accuracy': [accuracy, accuracy_rfe, accuracy_lasso]
})

# Display the results in the notebook
results
```

- **results** DataFrame: This contains the **method names** and their corresponding **accuracy** scores.
- The results will be displayed directly in the notebook.

Summary:

- **Filter Method (ANOVA F-value)**: Selects the best features based on statistical tests (works with scaled data).
- **Wrapper Method (RFE)**: Recursively eliminates the least important features by training a model (requires a model to perform the feature selection).
- **Embedded Method (Lasso Regression)**: Performs feature selection while fitting the model (built-in feature selection).

By comparing the accuracy scores of these methods, we can understand which feature selection technique performs best for the given dataset. The goal is to improve model accuracy by selecting only the most relevant features and discarding irrelevant or redundant ones.