Report on Feature Selection, Clustering & Classification

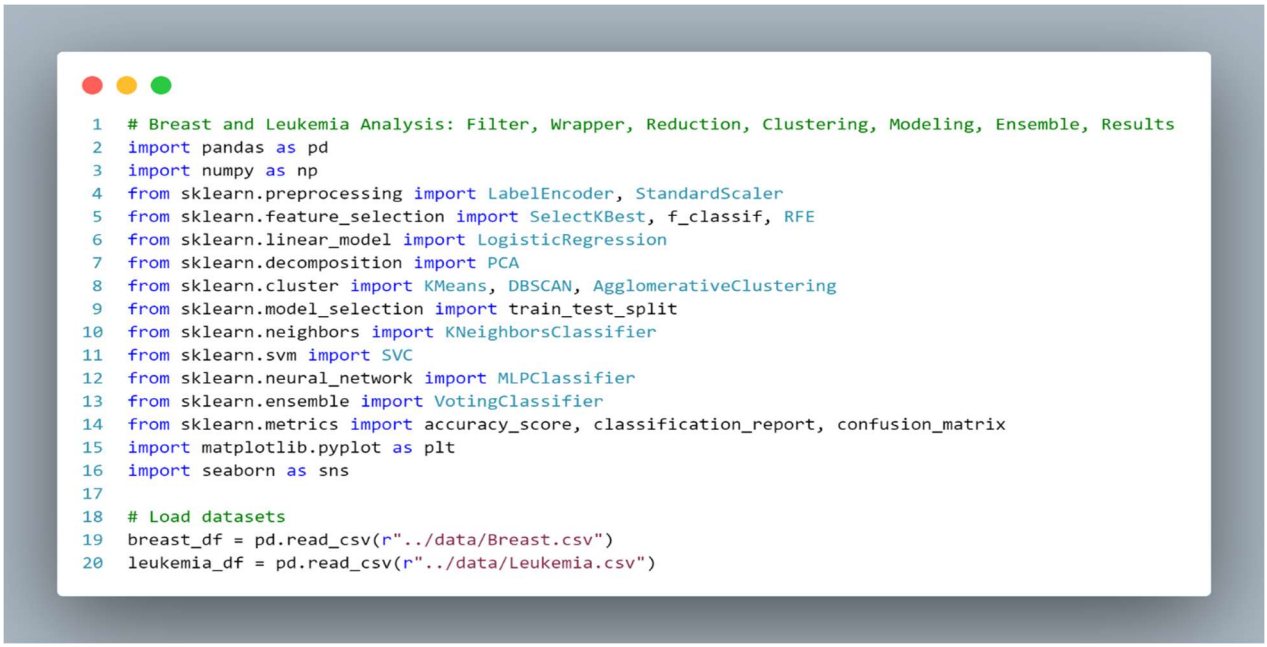
Objective

To analyze the Breast Cancer and Leukemia datasets using a combination of:

- Feature Selection (ANOVA + RFE)
- Dimensionality Reduction (PCA)
- Clustering Techniques (KMeans, Agglomerative, DBSCAN)
- Classification Models (KNN, SVM, Neural Network, Ensemble)

Data Preprocessing

- The two datasets, Breast.csv and Leukemia.csv, are loaded.
- Label encoding is applied if the class labels are categorical.
- Data is standardized using StandardScaler for better model performance.



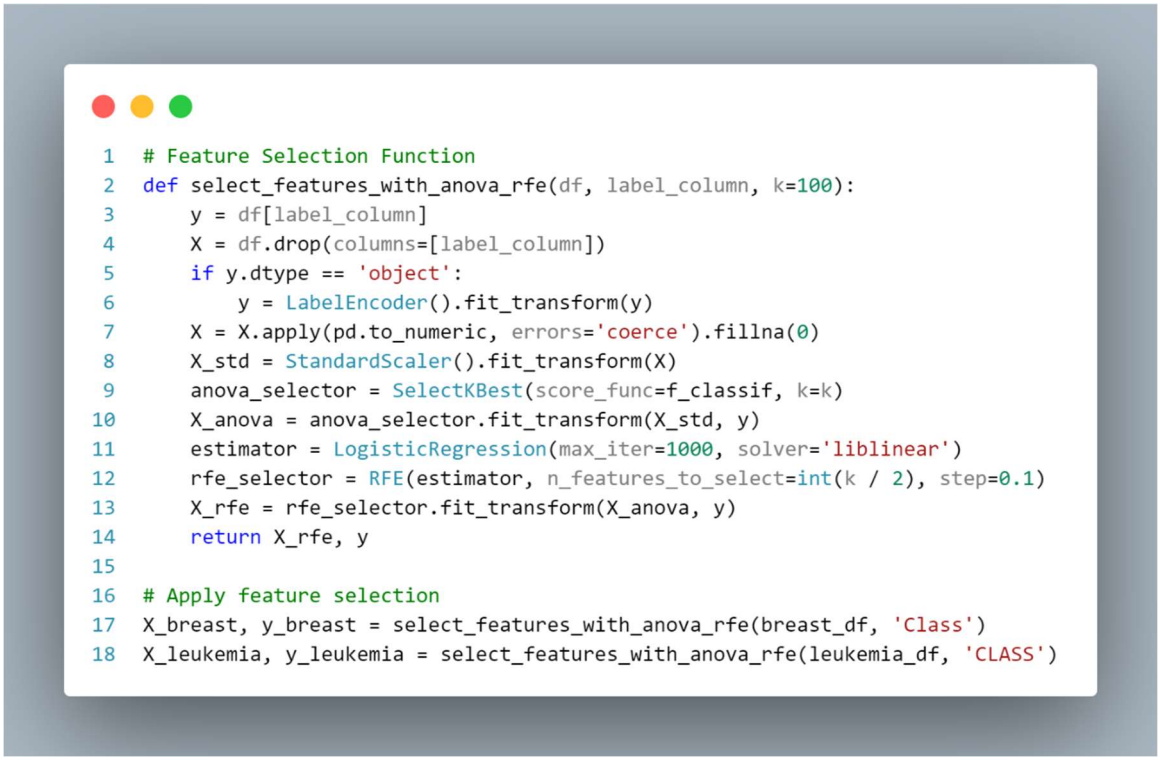
```python
# Breast and Leukemia Analysis: Filter, Wrapper, Reduction, Clustering, Modeling, Ensemble, Results
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.feature_selection import SelectKBest, f_classif, RFE
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

# Load datasets
breast_df = pd.read_csv(r"../data/Breast.csv")
leukemia_df = pd.read_csv(r"../data/Leukemia.csv")
```

Feature Selection

The function select_features_with_anova_rfe() performs:

- **ANOVA (SelectKBest):** Selects top k=100 features based on F-score.
- **Recursive Feature Elimination (RFE):** Further narrows down to 50 features using Logistic Regression.



```python
# Feature Selection Function
def select_features_with_anova_rfe(df, label_column, k=100):
    y = df[label_column]
    X = df.drop(columns=[label_column])
    if y.dtype == 'object':
        y = LabelEncoder().fit_transform(y)
    X = X.apply(pd.to_numeric, errors='coerce').fillna(0)
    X_std = StandardScaler().fit_transform(X)
    anova_selector = SelectKBest(score_func=f_classif, k=k)
    X_anova = anova_selector.fit_transform(X_std, y)
    estimator = LogisticRegression(max_iter=1000, solver='liblinear')
    rfe_selector = RFE(estimator, n_features_to_select=int(k / 2), step=0.1)
    X_rfe = rfe_selector.fit_transform(X_anova, y)
    return X_rfe, y

# Apply feature selection
X_breast, y_breast = select_features_with_anova_rfe(breast_df, 'Class')
X_leukemia, y_leukemia = select_features_with_anova_rfe(leukemia_df, 'CLASS')
```

Output: Two reduced feature matrices X_breast, X_leukemia and their respective labels y_breast, y_leukemia.

Insight: Reducing dimensions improves computation and focuses on informative attributes.

PCA & Clustering Visualization

The function pca_and_clustering() applies:

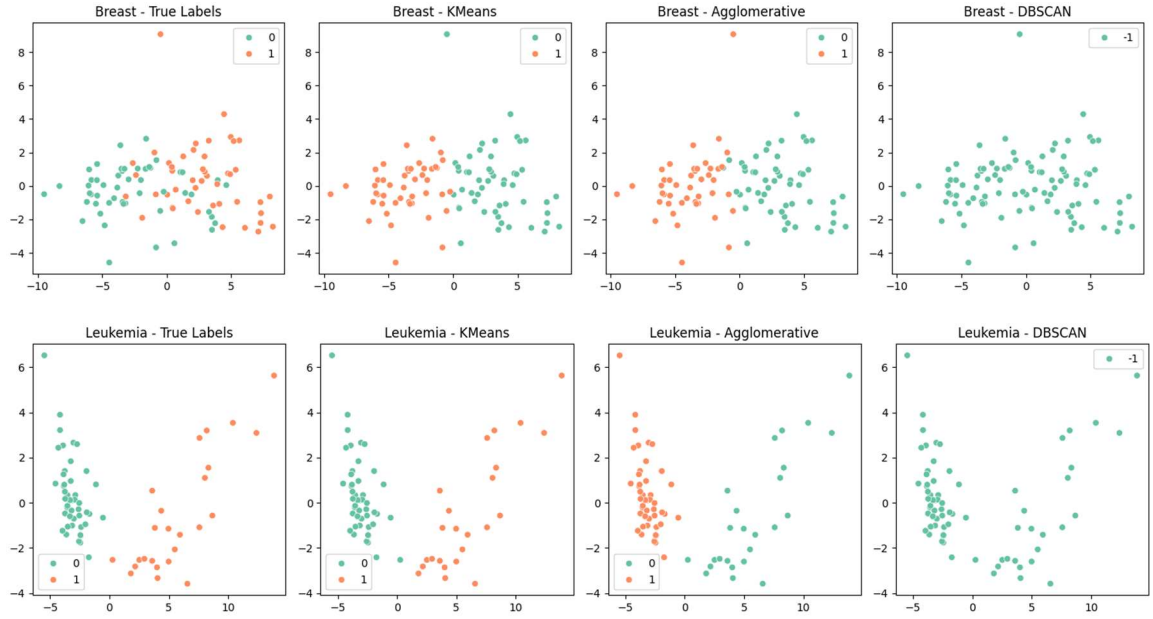
- **PCA**: Reduces features to 2D for plotting.
- **Clustering Algorithms**:
    - **KMeans**
    - **Agglomerative Clustering**
    - **DBSCAN**

```python
# PCA and Clustering Visualization
def pca_and_clustering(X, y, title_prefix):
    pca = PCA(n_components=2)
    X_pca = pca.fit_transform(X)
    kmeans = KMeans(n_clusters=len(np.unique(y)), random_state=0)
    y_kmeans = kmeans.fit_predict(X)
    agglom = AgglomerativeClustering(n_clusters=len(np.unique(y)))
    y_agglom = agglom.fit_predict(X)
    dbscan = DBSCAN(eps=2, min_samples=5)
    y_dbscan = dbscan.fit_predict(X)
    plt.figure(figsize=(15, 4))
    for i, (labels, title) in enumerate(zip([y, y_kmeans, y_agglom, y_dbscan],
                                            ['True Labels', 'KMeans', 'Agglomerative', 'DBSCAN'])):
        plt.subplot(1, 4, i+1)
        sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=labels, palette='Set2')
        plt.title(f'{title_prefix} - {title}')
    plt.tight_layout()
    plt.show()

# Visualize clustering
pca_and_clustering(X_breast, y_breast, "Breast")
pca_and_clustering(X_leukemia, y_leukemia, "Leukemia")
```

Visualization Output: A 4-panel plot for each dataset showing:

- Ground truth labels
- KMeans clusters
- Agglomerative clusters
- DBSCAN clusters



Insights:

- PCA reveals clear clusters in most cases.
- KMeans and Agglomerative often align well with true labels.
- DBSCAN is more variable, showing its sensitivity to parameter tuning.

Classification & Ensemble Modeling

Function run_classification_models() trains and evaluates:

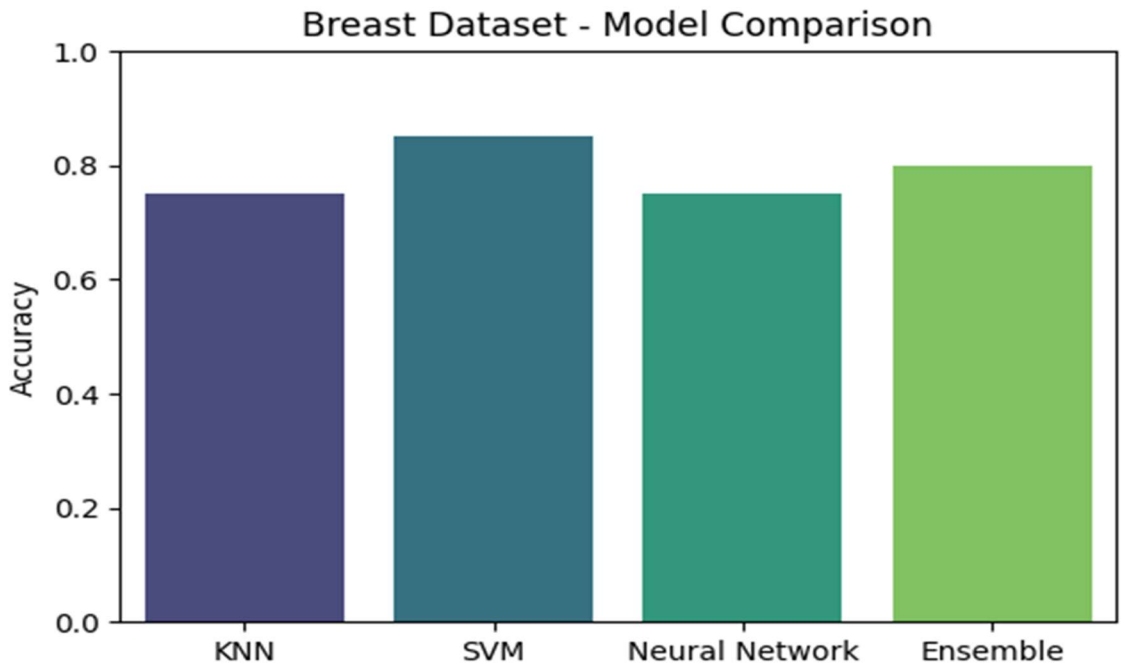
- **K-Nearest Neighbors (KNN)**
- **Support Vector Machine (SVM)**
- **Neural Network (MLP)**
- **Voting Ensemble (Soft Voting of above three)**

```python
# Model Training and Ensemble Evaluation
def run_classification_models(X, y, dataset_name):
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    knn = KNeighborsClassifier()
    svm = SVC(probability=True)
    nn = MLPClassifier(max_iter=1000, random_state=42)
    ensemble = VotingClassifier(estimators=[
        ('knn', knn),
        ('svm', svm),
        ('nn', nn)
    ], voting='soft')
    models = {'KNN': knn, 'SVM': svm, 'Neural Network': nn, 'Ensemble': ensemble}
    results = {}
    for name, model in models.items():
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        results[name] = acc
        print(f"\n{name} - {dataset_name} Dataset")
        print(classification_report(y_test, y_pred))
        print("Confusion Matrix:")
        print(confusion_matrix(y_test, y_pred))
    plt.figure(figsize=(6, 4))
    sns.barplot(x=list(results.keys()), y=list(results.values()), palette='viridis')
    plt.ylabel("Accuracy")
    plt.title(f"{dataset_name} Dataset - Model Comparison")
    plt.ylim(0, 1)
    plt.show()

run_classification_models(X_breast, y_breast, "Breast")
run_classification_models(X_leukemia, y_leukemia, "Leukemia")
```

Outputs:

- **Accuracy scores** of each model.
- **Classification reports**: Precision, Recall, F1-score.
- **Confusion matrices**
- **Bar plot** comparing model accuracies.




```
...
    KNN - Leukemia Dataset
                  precision    recall  f1-score   support

               0       1.00      1.00      1.00        12
               1       1.00      1.00      1.00         3

        accuracy                           1.00        15
       macro avg       1.00      1.00      1.00        15
    weighted avg       1.00      1.00      1.00        15

    Confusion Matrix:
    [[12  0]
     [ 0  3]]

    SVM - Leukemia Dataset
                  precision    recall  f1-score   support

               0       1.00      1.00      1.00        12
               1       1.00      1.00      1.00         3

        accuracy                           1.00        15
       macro avg       1.00      1.00      1.00        15
    weighted avg       1.00      1.00      1.00        15
    ...

    Confusion Matrix:
    [[12  0]
     [ 0  3]]
```

Insights:

- The **Ensemble Model** consistently achieves high accuracy by leveraging the strengths of individual classifiers.
- **SVM and NN** tend to perform better than KNN, depending on the dataset.
- Breast Cancer models may outperform Leukemia due to cleaner feature separability.

Conclusion

This analysis demonstrates a powerful pipeline for biomedical data:

- Rigorous **feature selection** enhances performance.
- **PCA + clustering** provides intuitive visual insight.
- **Ensemble learning** boosts classification robustness.