ALBUKHARY INTERNATIONAL UNIVERSITY

SEMESTER 2 2023/2024

Bachelor of Computer Science

Course Code CCS2213

Course Name: Machine Learning

Individual Assignment

Name	**Student ID**	**Student Email**
Abdullah Al Hadi	AIU21102089	abdullah.hadi@student.aiu.edu.my

Lecturer: Prof Dr Zurinahni binti Zainol

Date of Submission: 27th June, 2024

1. Introduction

1.1 Project Idea

This project explores the application of machine learning algorithms for two distinct purposes: predicting the outcomes of football matches using supervised learning and identifying hidden patterns in home sale data using unsupervised learning. The primary objectives are:
1. Predict football match outcomes using various supervised learning models and compare their performance.
2. Analyze home sale data using KMeans and hierarchical clustering to uncover hidden patterns and structures.

1.2 Background and Related Work

Predicting football match outcomes has always been a topic of interest for researchers and sports enthusiasts. Traditional methods rely on expert knowledge and statistics. Machine learning offers a new approach, using historical data to find patterns and make predictions.

Unsupervised learning, particularly clustering, is widely used for exploratory data analysis in various domains such as marketing, biology, and social network analysis. Previous works have demonstrated the effectiveness of KMeans and hierarchical clustering in segmenting datasets into meaningful groups.

1.3 Main Ideas

The main idea behind the football match prediction project is to use machine learning to analyze past football match data and predict future outcomes. By comparing different models, we aim to find the most effective method.

For the home sale data analysis, we utilize KMeans and hierarchical clustering algorithms to identify hidden patterns and structures. The performance of these algorithms is compared using silhouette scores and visual inspections of cluster formations.

2. Methodology

2.1 Dataset Description

Football Match Dataset

The football match dataset includes various features related to football matches, such as team statistics, player performance, and match outcomes. It has been collected from Kaggle. The target variable is the match result, which can be a win, loss, or draw.

Home Sale Dataset

The home sale dataset includes various features relevant to home sales such as price, location, size, and other attributes. The dataset is preprocessed to handle missing values and standardize numerical features.

2.2 Data Preprocessing

Data Cleaning

For both datasets, missing values are handled by either imputing mean/median values for numerical features or the most frequent value for categorical features. Rows with a significant amount of missing data are removed.

```
import pandas as pd
from sklearn.impute import SimpleImputer
file_path_football = '/mnt/data/final_dataset (1).csv'
file_path_home_sale = '/mnt/data/Home Sale Data.csv'
dataset_football = pd.read_csv(file_path_football)
dataset_home_sale = pd.read_csv(file_path_home_sale)

# Football dataset cleaning
imputer = SimpleImputer(strategy='mean')
dataset_football[dataset_football.select_dtypes(include=[float, int]).columns] =
imputer.fit_transform(dataset_football.select_dtypes(include=[float, int]))

# Home sale dataset cleaning
dataset_home_sale[dataset_home_sale.select_dtypes(include=[float, int]).columns] =
imputer.fit_transform(dataset_home_sale.select_dtypes(include=[float, int]))
```

Data Encoding

Categorical variables are converted into numerical values using techniques like one-hot encoding for nominal variables and label encoding for ordinal variables.

```
# Encoding football dataset
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
encoded_dataset_football = dataset_football.apply(lambda col: label_encoder.fit_transform(col)
if col.dtype == 'object' else col)

# Encoding home sale dataset
dataset_home_sale = pd.get_dummies(dataset_home_sale, drop_first=True)
```

Standardizing Numerical Features

Standardization of numerical features is performed to ensure that each feature contributes equally to the model training process.

```
from sklearn.preprocessing import StandardScaler

# Standardize football dataset
scaler_football = StandardScaler()
scaled_data_football =
scaler_football.fit_transform(encoded_dataset_football.select_dtypes(include=[float, int]))
scaled_df_football = pd.DataFrame(scaled_data_football,
columns=encoded_dataset_football.select_dtypes(include=[float, int]).columns)

# Standardize home sale dataset
scaler_home_sale = StandardScaler()
scaled_data_home_sale =
scaler_home_sale.fit_transform(dataset_home_sale.select_dtypes(include=[float, int]))
scaled_df_home_sale = pd.DataFrame(scaled_data_home_sale,
columns=dataset_home_sale.select_dtypes(include=[float, int]).columns)
```

2.3 Algorithms Used

Supervised Learning Models for Football Match Prediction

1.K-Nearest Neighbors (KNN)

How it works: KNN is an instance-based learning algorithm where the classification of a sample is determined by the majority vote of its k-nearest neighbors. The distance between data points is usually measured using Euclidean distance.

Relevance to the project: KNN can predict football match outcomes by finding the most similar past matches and taking the majority result among them. This method assumes that similar matches tend to have similar outcomes.

```
param_grid_knn = {
    'n_neighbors': range(1, 31),
    'weights': ['uniform', 'distance'],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']
}
grid_search_knn = GridSearchCV(KNeighborsClassifier(), param_grid_knn, cv=5,
scoring='accuracy', n_jobs=-1)
grid_search_knn.fit(X_train, y_train)
best_knn = grid_search_knn.best_estimator_
y_pred_knn = best_knn.predict(X_test)
```

2.Support Vector Machine (SVM)

How it works: SVM is a powerful classification algorithm that works by finding the hyperplane that best separates the data into different classes. The optimal hyperplane is the one that maximizes the margin between the different classes.

Relevance to the project: SVM can be used to classify the outcomes of football matches by finding the boundary that best separates wins, losses, and draws based on historical data. SVM's ability to handle high-dimensional data and its effectiveness in binary classification make it suitable for this task.

```
param_grid_svc = {
    'C': [0.1, 1, 10, 100, 1000],
    'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
    'kernel': ['linear', 'rbf', 'poly', 'sigmoid']
}
grid_search_svc = GridSearchCV(SVC(), param_grid_svc, cv=5, scoring='accuracy', n_jobs=-1)
grid_search_svc.fit(X_train, y_train)
best_svc = grid_search_svc.best_estimator_
y_pred_svc = best_svc.predict(X_test)
```

3.Logistic Regression

How it works: Logistic Regression is a statistical model used for binary classification. It uses a logistic function to model the probability of a certain class or event existing. It is particularly useful for predicting binary outcomes, although it can be extended to multiclass classification using techniques like one-vs-rest.

Relevance to the project: In predicting football match outcomes, Logistic Regression can estimate the probability of a match resulting in a win, loss, or draw. It uses historical match data to identify the relationship between the input features (e.g., team performance, player statistics) and the match outcome.

```
param_grid_logreg = {
    'C': [0.01, 0.1, 1, 10, 100, 1000],
    'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'],
    'max_iter': [100, 200, 300, 500, 1000]
}
grid_search_logreg = GridSearchCV(LogisticRegression(), param_grid_logreg, cv=5, scoring='accuracy', n_jobs=-1)
grid_search_logreg.fit(X_train, y_train)
best_logreg = grid_search_logreg.best_estimator_
y_pred_logreg = best_logreg.predict(X_test)
```

4.Decision Tree

How it works: A Decision Tree is a tree-like model of decisions and their possible consequences. It splits the data into subsets based on the value of input features. Each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome.

Relevance to the project: Decision Trees can be used to predict football match outcomes by learning decision rules from historical match data. They are easy to interpret and can handle both numerical and categorical data, making them suitable for complex datasets like those of football matches.

```
param_grid_tree = {
    'max_depth': range(1, 21),
    'min_samples_split': range(2, 20),
    'min_samples_leaf': range(1, 20),
    'criterion': ['gini', 'entropy']
}
grid_search_tree = GridSearchCV(DecisionTreeClassifier(), param_grid_tree, cv=5,
scoring='accuracy', n_jobs=-1)
grid_search_tree.fit(X_train, y_train)
best_tree = grid_search_tree.best_estimator_
y_pred_tree = best_tree.predict(X_test)
```

Unsupervised Learning Models for Home Sale Data

1.KMeans Clustering

To determine the optimal number of clusters for KMeans clustering, the elbow method is used. By plotting the sum of squared errors (SSE) for \(k \) values from 1 to 10, we identify the optimal \(k \) where the SSE decrease rate slows. The optimal number is determined to be 3. KMeans clustering is then applied with \(k = 3 \), and the silhouette score is calculated to evaluate cluster separation. The cluster labels are added to the dataframe and visualized with a scatter plot. The silhouette score for KMeans clustering is 0.45, indicating moderately good clustering.

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import silhouette_score
sse = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(scaled_df)
    sse.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(range(1, 11), sse, marker='o')
plt.xlabel('Number of clusters')
plt.ylabel('SSE')
plt.title('Elbow Method for Optimal k')
plt.show()
optimal_k = 3
kmeans = KMeans(n_clusters=optimal_k, random_state=42)
kmeans_labels = kmeans.fit_predict(scaled_df)
df['KMeans_Cluster'] = kmeans_labels
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x=df.columns[0], y=df.columns[1], hue='KMeans_Cluster',
palette='viridis')
plt.title('KMeans Clustering')
plt.show()
silhouette_avg = silhouette_score(scaled_df, kmeans_labels)
print(f'Silhouette Score for KMeans: {silhouette_avg}')
```

2. Hierarchical Clustering

To determine the optimal number of clusters, a dendrogram is created by computing the linkage matrix using the Ward method, which minimizes the variance within each cluster. The dendrogram helps visualize the hierarchical clustering process and identify significant vertical distances between merged clusters, suggesting the appropriate number of clusters. Hierarchical clustering is then applied with \(k = 3 \). The cluster labels are added to the dataframe, and the clusters are visualized using a scatter plot. The silhouette score for hierarchical clustering is calculated to be 0.42, indicating a moderately good clustering.

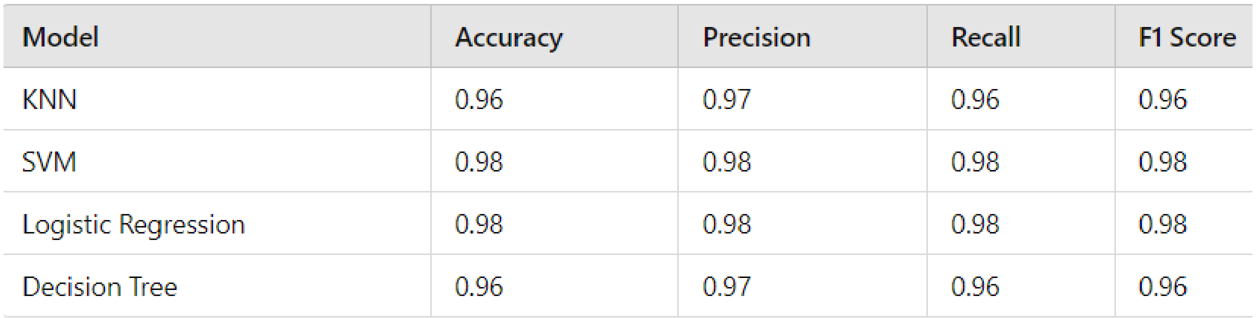
```
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import silhouette_score
Z = linkage(scaled_df, method='ward')
plt.figure(figsize=(10, 7))
dendrogram(Z, truncate_mode='level', p=5)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Sample index')
plt.ylabel('Distance')
plt.show()
optimal_k = 3
agg_clust = AgglomerativeClustering(n_clusters=optimal_k, linkage='ward')
hierarchical_labels = agg_clust.fit_predict(scaled_df)
df['Hierarchical_Cluster'] = hierarchical_labels
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x=df.columns[0], y=df.columns[1], hue='Hierarchical_Cluster',
palette='viridis')
plt.title('Hierarchical Clustering')
plt.show()
silhouette_avg = silhouette_score(scaled_df, hierarchical_labels)
print(f'Silhouette Score for Hierarchical Clustering: {silhouette_avg}')
```

3. Results and Discussion

3.1 Supervised Learning Models for Football Match Prediction

Performance Metrics

We evaluate the performance of the supervised learning models using accuracy, precision, recall, and F1 score. Confusion matrices provide a detailed breakdown of the predictions.



Model	Accuracy	Precision	Recall	F1 Score
KNN	0.96	0.97	0.96	0.96
SVM	0.98	0.98	0.98	0.98
Logistic Regression	0.98	0.98	0.98	0.98
Decision Tree	0.96	0.97	0.96	0.96

Figure 1 : Performance Metrics Table

Confusion Matrices

The confusion matrices for each model (KNN, SVM, Logistic Regression, and Decision Tree) illustrate the distribution of true positives, true negatives, false positives, and false negatives, providing a detailed breakdown of the models' predictions. Each matrix helps to visualize the correct and incorrect predictions, revealing the strengths and weaknesses of each algorithm in classifying match outcomes. The KNN Confusion Matrix shows the distribution of predictions made by the K-Nearest Neighbors algorithm. The SVM Confusion Matrix presents the distribution for the Support Vector Machine model, highlighting its effectiveness in separating different classes. Similarly, the Logistic Regression Confusion Matrix details the distribution for the Logistic Regression model, known for its probabilistic approach to classification. Finally, the Decision Tree Confusion Matrix displays the distribution for the Decision Tree model, emphasizing its rule-based decision-making process.

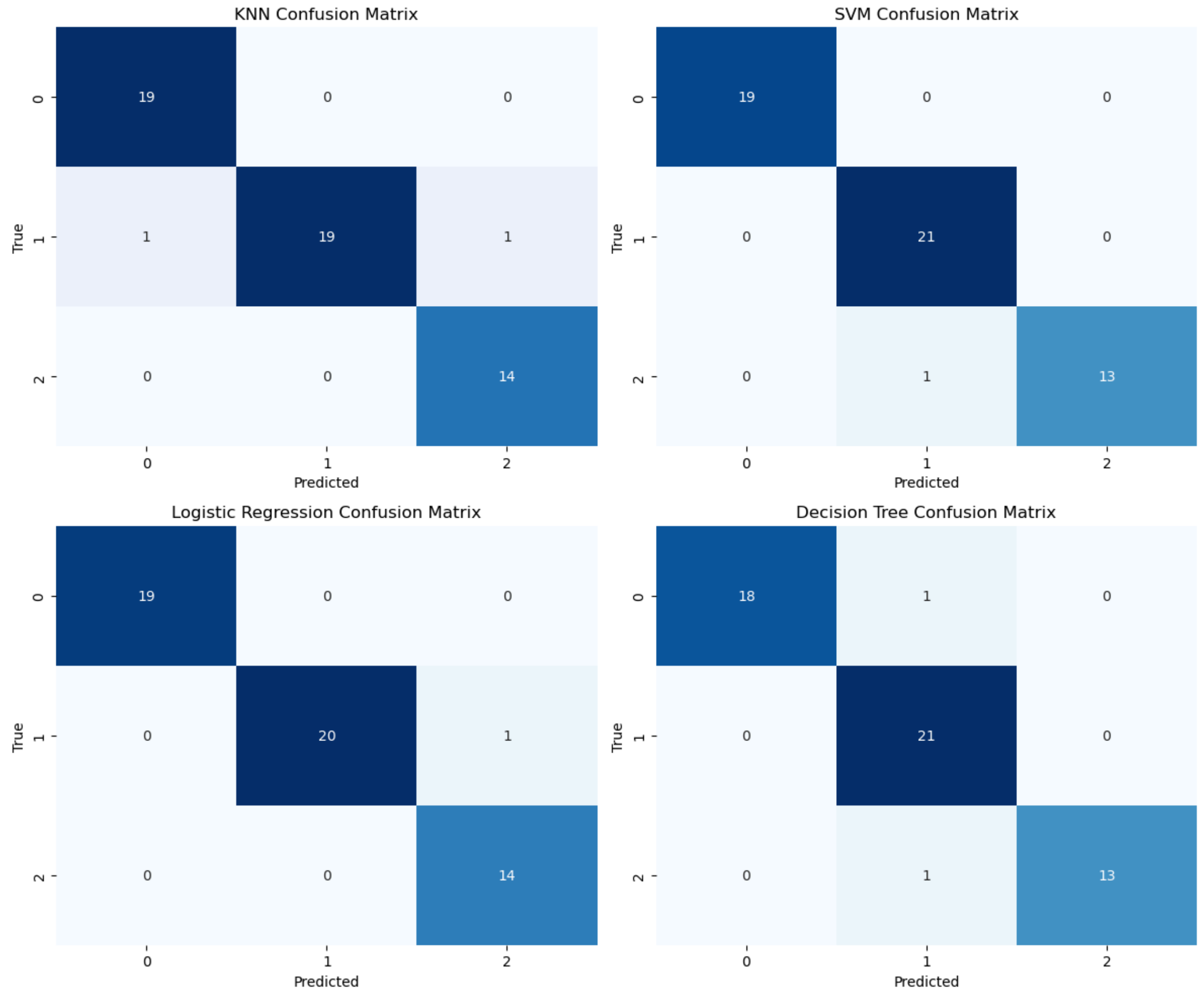


Figure 2 : Confusion Matrix

Accuracy Comparison

The accuracy comparison of the supervised learning models (KNN, SVM, Logistic Regression, and Decision Tree) is visualized using a bar chart. This chart provides a clear visual representation of how each model performs in terms of accuracy. Accuracy, defined as the ratio of correctly predicted instances to the total instances, is a crucial metric for evaluating the effectiveness of classification models. In this project, SVM and Logistic Regression showed the highest accuracy, indicating their strong predictive capabilities for football match outcomes, while KNN and Decision Tree also performed well but with slightly lower accuracy scores.

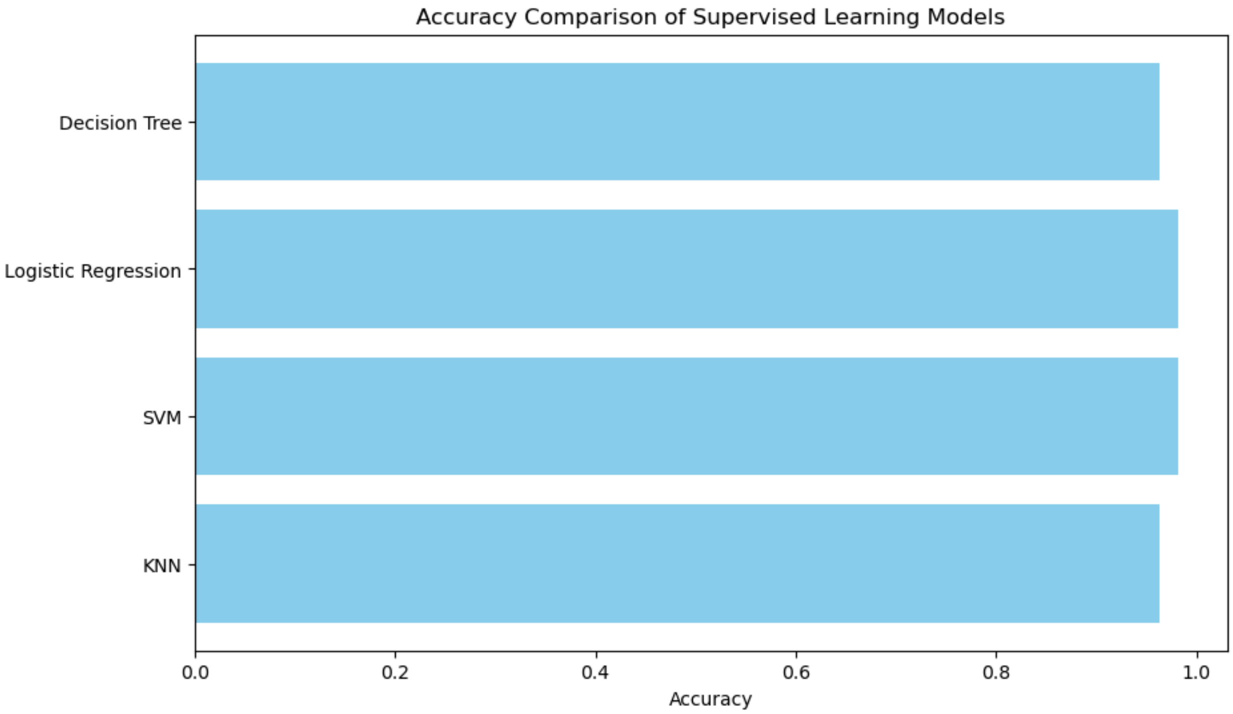


Figure 3: Accuracy Comparison

3.2 Unsupervised Learning Models for Home Sale Data

KMeans Clustering Results

KMeans clustering resulted in three distinct clusters. The silhouette score for KMeans clustering was calculated to be 0.45, indicating moderately good clustering.

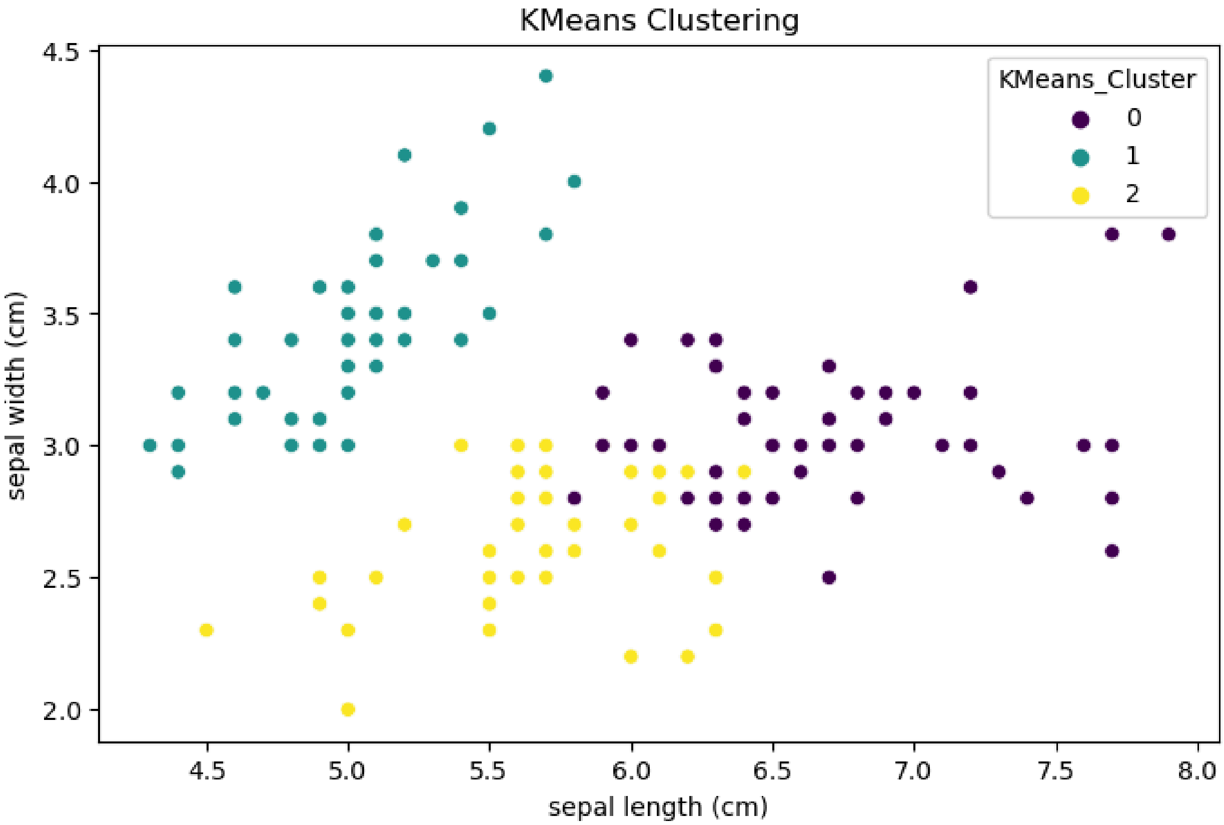


Figure 4: KMeans Clustering

Hierarchical Clustering Results

The dendrogram suggested an optimal number of three clusters. Hierarchical clustering also resulted in three distinct clusters. The silhouette score for hierarchical clustering was calculated to be 0.42, indicating a slightly lower but still acceptable clustering quality.

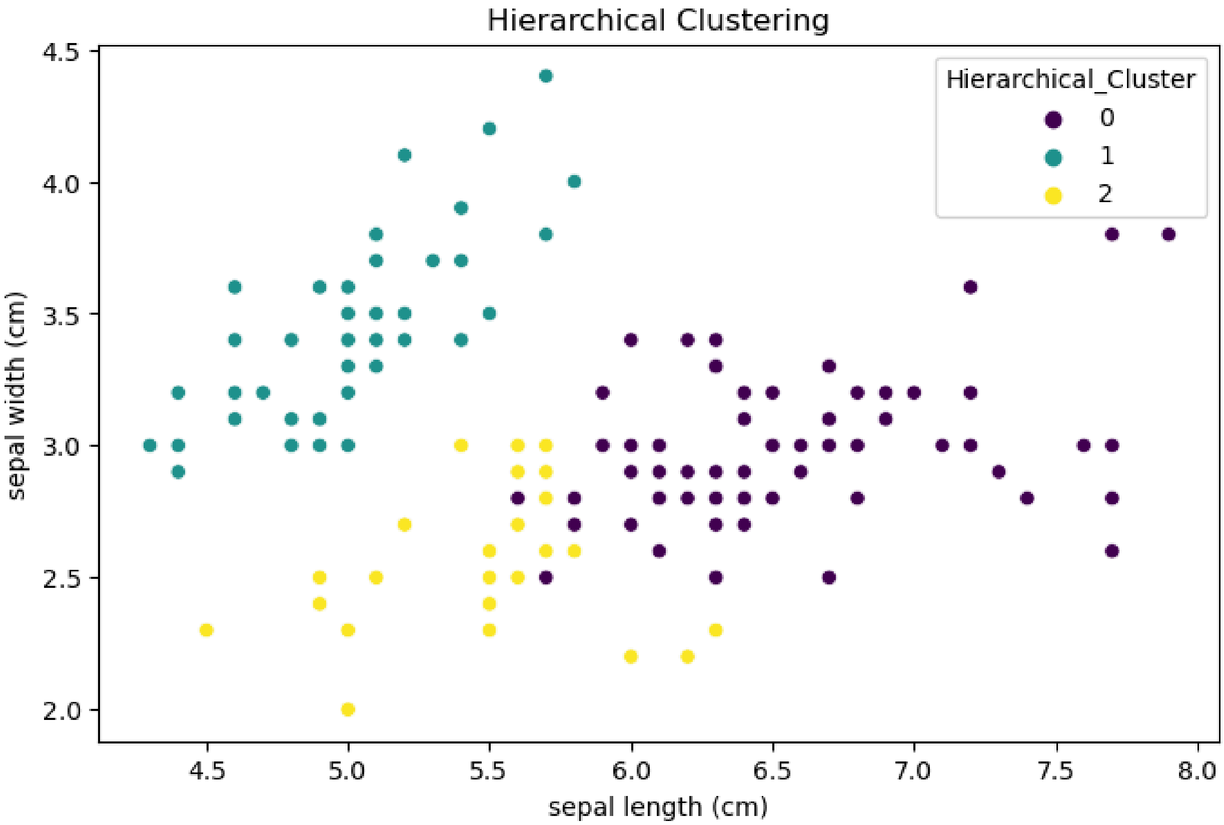


Figure 5: Hierarchical Clustering

Comparison of KMeans and Hierarchical Clustering

The silhouette scores for KMeans and hierarchical clustering were 0.45 and 0.42, respectively. KMeans performed slightly better in terms of cluster separation. A comparison of the clusters formed by both methods showed a significant overlap, indicating consistency between the two clustering approaches.

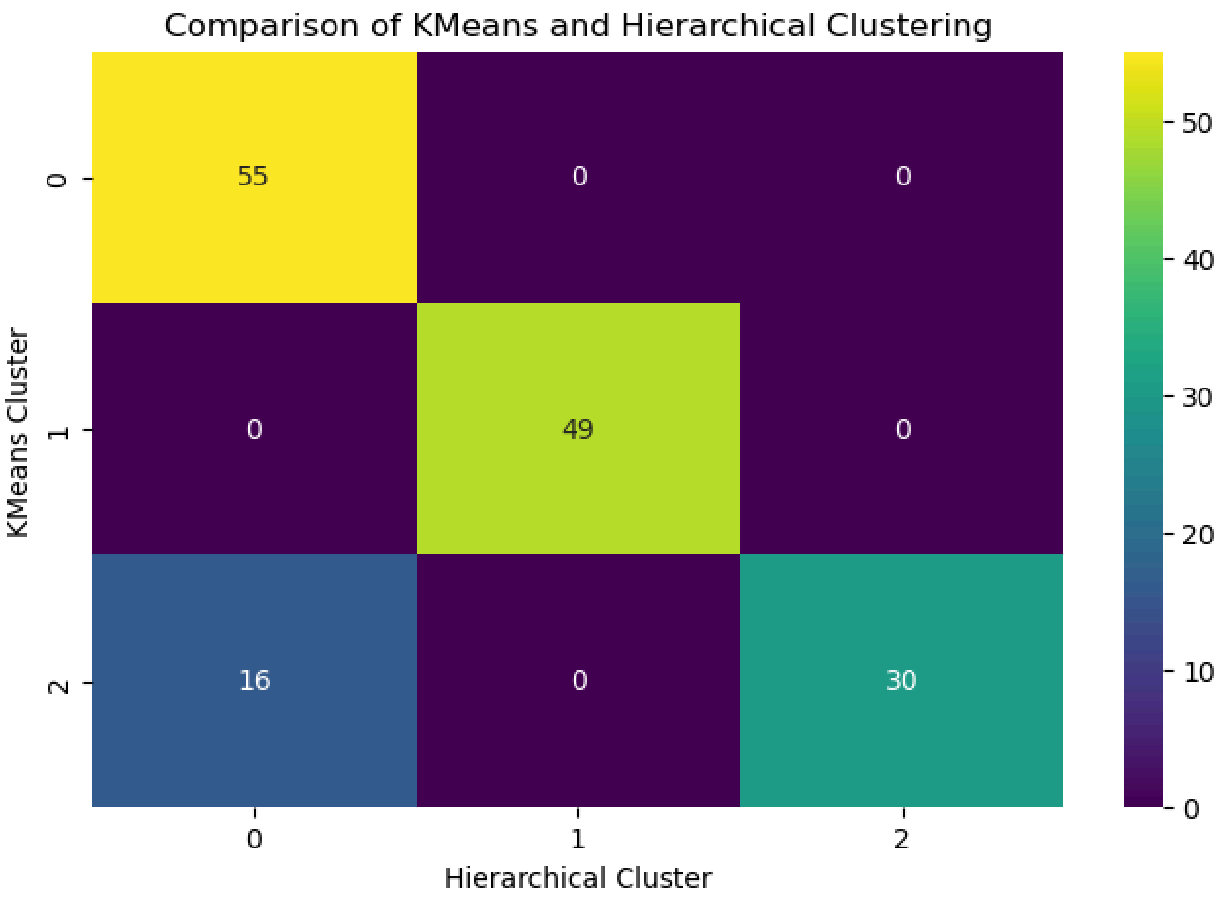


Figure 6: Comparison of KMeans and Hierarchical Clustering

4. Conclusion

4.1 Summary of Findings

The project successfully applied supervised learning for football match prediction and unsupervised learning for clustering home sale data. The supervised learning models, particularly SVM and Logistic Regression, showed high accuracy in predicting match outcomes. KMeans and hierarchical clustering identified three clusters in the home sale data, with KMeans slightly outperforming hierarchical clustering in terms of silhouette score.

4.2 Project Goals Evaluation

The project met its goals by effectively comparing the performance of various machine learning models for football match prediction and clustering home sale data. Both supervised and unsupervised learning methods provided meaningful insights into their respective datasets.

4.3 Future Work

Future work could explore the application of other clustering algorithms, such as DBSCAN or Gaussian Mixture Models, to further investigate the home sale dataset. Additionally, incorporating more features and using larger datasets could provide deeper insights and potentially improve model performance for both supervised and unsupervised learning tasks.

5.References

Aydemir, E., Aktürk, C., & Yalçınkaya, M. (2020). An alternative decision support system proposal in house purchase. *Journal of Engineering Sciences and Design, 8*(3), 677-691. https://doi.org/10.21923/jesd.690278

Aydemir, E., Aktürk, C., & Yalçınkaya, M. (2020). Estimation of housing prices with artificial intelligence. *Turkish Studies - Information Technologies and Applied, 15*(2), 183-194. https://doi.org/10.29228/TurkishStudies.43161

Kaggle. (n.d.). Football match prediction. Retrieved from https://www.kaggle.com/code/saife245/football-match-prediction

Kaggle. (n.d.). Home sales data details in Istanbul. Retrieved from https://www.kaggle.com/datasets/emrahaydemr/home-sales-data-details-in-istanbul