

Master of Science (MS) in Data Science

Module: ITC6003A1 – Applied Machine Learning

Instructor: Dimitrios Vogiatzis

Students’ names:

Konstantinos Megaritis 271868

Konstantinos Margaritis 122451

George Kapolis 111890

Term: Winter Term 2023

Type: Group Assignment

Submission Date: Sunday, April 9

Words:

Table of Contents

[Introduction 3](#_Toc131746577)

[Data Structure 4](#_Toc131746579)

[Type of data 4](#_Toc131746580)

[Description of dataset 5](#_Toc131746581)

[Data Quality Assessment 6](#_Toc131746582)

[Dataset Preview 6](#_Toc131746583)

[Data Cleaning 7](#_Toc131746584)

[Data Analysis 9](#_Toc131746585)

[Graphs – Insights – Recommendations 9](#_Toc131746586)

[Regression - Forecasting 16](#_Toc131746587)

[Conclusion 18](#_Toc131746588)

[References 19](#_Toc131746589)

[Appendix 20](#_Toc131746590)

# Introduction

Machine learning has become a potent tool for processing and interpreting massive information in recent years. This has sparked an increase in the creation and use of numerous machine learning algorithms across numerous areas. This report's goal is to investigate and assess three typical machine learning tasks: outcome prediction, regression, and clustering. Three different datasets have been used for these tasks, each of which presented a different set of difficulties.

The first task entails clustering, a method of unsupervised learning that entails gathering related data points into clusters. The Wholesale Customers dataset, which offers details on a wholesale distributor's customers, has been used for this purpose. In this dataset, finding clusters and evaluating and characterizing them are our main objectives.

The second task is regression, which entails forecasting a continuous variable from a collection of input features. We selected our own dataset for this work, which comes with a number of difficulties like size, missing values, and categorical features. We have investigated many regression methods, such as linear,lasso and polynomial regression, and assessed their effectiveness.

The third and final task involves predicting the annual income of a person based on various demographic and socio-economic factors. For this task, we have used the UCI Adult dataset, which provides information on individuals based on various features such as age, education level, occupation, and marital status. We have trained various classifiers to predict the income of an individual, and also explored the explainability of our models using Shapley Additive Explanation (SHAP) or the Local Interpretable Model-Agnostic (LIME).

Overall, this report aims to provide a comprehensive analysis of the three machine learning tasks, including data preprocessing, feature selection, algorithm selection, evaluation, and conclusion. We hope that our findings will provide useful insights for future researchers in the field of machine learning.

# 

# Clustering Analysis – Market Segmentation

Market segmentation is a process of dividing a larger market into smaller groups of consumers or businesses with similar needs, preferences, or behaviors. Market segmentation seeks to discover particular categories of potential customers to whom a company can tailor more precise and effective marketing strategies, goods, and services.

# Our code

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans, AgglomerativeClustering

from sklearn.metrics import silhouette\_score

from sklearn.preprocessing import OneHotEncoder, StandardScaler

from scipy.cluster.hierarchy import dendrogram, linkage

from sklearn.tree import DecisionTreeClassifier, plot\_tree

from sklearn.model\_selection import train\_test\_split

# Load the dataset

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale%20customers%20data.csv'

data = pd.read\_csv(url, header=0, delimiter=',')

# Identify categorical columns and apply one-hot encoding

cat\_cols = ['Channel', 'Region']

encoder = OneHotEncoder()

X\_cat = data[cat\_cols].astype('category')

X\_cat\_encoded = encoder.fit\_transform(X\_cat)

X\_num = data.drop(cat\_cols, axis=1)

X\_norm = np.hstack((X\_num, X\_cat\_encoded.toarray()))

# Scale the data

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(X\_norm)

# Elbow method to determine the optimal number of clusters

wcss = []

for i in range(1, 11):

kmeans = KMeans(n\_clusters=i, init='k-means++', max\_iter=300, n\_init=10, random\_state=42)

kmeans.fit(X\_scaled)

wcss.append(kmeans.inertia\_)

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()

# Silhouette score to evaluate the clustering quality

sil\_scores = []

for i in range(2, 11):

kmeans = KMeans(n\_clusters=i, random\_state=42)

kmeans.fit(X\_scaled)

labels = kmeans.labels\_

sil\_scores.append(silhouette\_score(X\_scaled, labels))

plt.plot(range(2, 11), sil\_scores)

plt.title('Silhouette Score')

plt.xlabel('Number of clusters')

plt.ylabel('Silhouette score')

plt.show()

# Run K-means with the optimal number of clusters

kmeans = KMeans(n\_clusters=3, random\_state=42)

kmeans.fit(X\_scaled)

labels\_kmeans = kmeans.labels\_

# Run hierarchical clustering

ward = linkage(X\_scaled, method='ward')

plt.figure(figsize=(10, 7))

dendrogram(ward)

plt.title("Dendrogram")

plt.xlabel("Observations")

plt.ylabel("Distance")

plt.show()

# Cut the dendrogram to obtain the optimal number of clusters

cut\_tree = AgglomerativeClustering(n\_clusters=3)

labels\_agg = cut\_tree.fit\_predict(X\_scaled)

# Print the cluster centers and number of data points in each cluster for K-means

print("K-means cluster centers:\n", kmeans.cluster\_centers\_)

print("Number of data points in each cluster for K-means:\n", np.bincount(labels\_kmeans))

# Print the number of data points in each cluster for hierarchical clustering

print("Number of data points in each cluster for hierarchical clustering:\n", np.bincount(labels\_agg))

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, data['Channel'], test\_size=0.2, random\_state=42)

# Fit a decision tree classifier with pruning

clf = DecisionTreeClassifier(criterion='entropy', max\_depth=4, random\_state=42)

clf.fit(X\_train, y\_train)

# Plot the decision tree

plt.figure(figsize=(20,10))

plot\_tree(clf, filled=True)

plt.show()

# Calculate accuracy on the testing data

accuracy = clf.score(X\_test, y\_test)

print('Accuracy:', accuracy)

## Description of dataset

# The Wholesale Customers dataset contains information on the annual spending of various products by clients of a wholesale distributor. The dataset includes the following attributes for each client:

# FRESH: annual spending on fresh products (e.g., fruits, vegetables)

# MILK: annual spending on milk and dairy products

# GROCERY: annual spending on grocery products (e.g., cereals, snacks, canned goods)

# FROZEN: annual spending on frozen products (e.g., frozen meals, ice cream)

# DETERGENTS\_PAPER: annual spending on detergents and paper products (e.g., cleaning supplies, paper towels)

# DELICATESSEN: annual spending on deli products (e.g., meats, cheeses, prepared foods)

* CHANNEL: customer channel - Horeca (Hotel/Restaurant/Cafe) or Retail channel (Nominal data type)
* REGION: customer region - Lisnon, Oporto or Other (Nominal data type)

# The dataset consists of 440 instances and no missing values. As for the type of data, it is numerical and continuous . The dataset's goal is to spot trends in customers' purchase patterns and perhaps even find groups of customers with comparable buying tendencies. Clustering analysis can be used to classify customers and adjust marketing plans as necessary.

# Data Preprocessing

The data preprocessing step is is an essential component of any machine learning pipeline since it entails preparing the data for effective analysis and modeling. Several preparation approaches are done to the wholesale customers dataset in the code above to get it into a suitable format for clustering and classification.

Identifying any categorical features in the dataset is the first step in data preprocessing. Because the 'Channel' and 'Region' columns are categorical in this scenario, they must be turned into numerical data. The scikit-learn OneHotEncoder method is used to do this. This function generates a binary column for each category, with 1 indicating that a row corresponds to that category and 0 indicating that it does not.. The resulting numerical data is easier to work with for machine learning algorithms.

# The data is scaled using the StandardScaler function from scikit-learn after the categorical characteristics have been converted. Making sure that all features are scaled equally is a crucial step that can enhance the effectiveness of clustering and classification algorithms. When working with characteristics that have vastly different ranges or units, as is frequently the case with real-world datasets, scaling is very crucial.

Overall, the data preprocessing step is a crucial part of any machine learning pipeline. By properly preparing the data, we can improve the accuracy and effectiveness of our models, and gain deeper insights into the underlying patterns and trends in the data.

# Data Analysis – Code Explanation

The code we wrote used several Python libraries such as Pandas, Scikit-learn, Matplotlib, and Scipy to perform data manipulation, clustering, visualization, and classification tasks.We are going to explain our thought process writing this code,step by step.

1.Loading and Preprocessing the Dataset:

The first step is to load the wholesale customers dataset from a URL and preprocess it. We use the Pandas library to read the dataset and apply one-hot encoding to the categorical columns 'Channel' and 'Region'. We then scale the dataset using the StandardScaler from the Scikit-learn library.

2.Determining the Optimal Number of Clusters:

The ideal number of clusters for K-means clustering is found using the elbow approach and silhouette score. We locate the "elbow" point, which denotes the ideal number of clusters, by plotting the within-cluster sum of squares (WCSS) against the number of clusters. For each cluster count, we also determine the silhouette score, which evaluates how well the clustering was done. The grouping is better the higher the score. To calculate the ideal number of clusters, we plot the silhouette score versus the number of clusters.

3.Performing Clustering:

We perform clustering using K-means clustering and hierarchical clustering. For K-means clustering, we utilize the KMeans function from the Scikit-learn package, and for hierarchical clustering, we use the AgglomerativeClustering function from Scipy. We extract the labels for each observation and fit the models to the scaled dataset..

4.Visualizing Clustering:

We use the Matplotlib library to visualize the dendrogram for hierarchical clustering and to plot the clusters for K-means clustering. We also print the number of data points in each cluster for both K-means and hierarchical clustering.

5.Classification:

We split the data into training and testing sets using the train\_test\_split function from Scikit-learn. We then fit a decision tree classifier with pruning using the DecisionTreeClassifier function from Scikit-learn. We plot the decision tree using the plot\_tree function from Scikit-learn and calculate the accuracy on the testing data.

Results

|  |
| --- |
| K-means cluster centers: |
| [[ 0.12025581 -0.35518821 -0.42909308 0.14280689 -0.4468872 -0.08880848 |
| 0.69029709 -0.69029709 -0.46056619 0.0350583 0.36491781] |
| [-0.22622697 0.72693658 0.9192496 -0.25775691 0.94477634 0.20543936 |
| -1.433152 1.433152 -0.21264236 0.09991836 0.11099303] |
| [ 0.04060061 -0.24660083 -0.38644204 0.0247252 -0.37531222 -0.11272206 |
| 0.52319172 -0.52319172 2.17124059 -0.34582203 -1.59636684]] |
| Number of data points in each cluster for K-means: |
| [238 138 64] |
| Number of data points in each cluster for hierarchical clustering: |
| [104 125 211] |
| Accuracy: 1.0 |

Εικόνα που περιέχει διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

# 

# Conclusion

In this Python code, we have performed clustering and classification tasks on the wholesale customers dataset. We have used the elbow method and silhouette score to determine the optimal number of clusters for K-means clustering, performed K-means clustering and hierarchical clustering with the optimal number of clusters, visualized the clustering results, and fit a decision tree classifier with pruning. The code can be used as a starting point for more advanced data analysis and machine learning tasks on the wholesale customers dataset or other similar datasets.

# References

Beck, N., & Katz, J. (1995). What To Do (and Not to Do) with Time-Series Cross-Section Data. American Political Science Review, 89(3), 634–647. <https://doi.org/10.2307/2082979>

Pipino, L. L., Lee, Y. W., & Wang, R. R. (2002). Data quality assessment. Communications of the ACM, 45(4), 211–218. <https://doi.org/10.1145/505248.506010>

Su, X., Yan, X., & Tsai, C. (2012). Linear regression. Wiley Interdisciplinary Reviews: Computational Statistics, 4(3), 275–294. <https://doi.org/10.1002/wics.1198>

# Appendix

*This form should be filled out by all team members after the completion of the group assignment. The team leader should be chosen upon agreement and is responsible to upload the group assignment after its completion and deal with any technical and other issues that might arise during the submission process.*

Team Leader name: Alkiviadis Kariotis

Team Leader ID: 241735

|  |
| --- |
| Team member name: Alkiviadis Kariotis  Team member ID: 241735  ***I herewith express my agreement with the submission of this final version of the group project by the team leader.***  A pair of glasses  Description automatically generated with medium confidence  Date: 11/03/2023  Team member Signature: |
| Team member name: Konstantinos Prozymas  Team member ID: 27365  ***I herewith express my agreement with the submission of this final version of the group project by the team leader.***    Date: 11/03/2023  Team member Signature: Konstantinos Prozymas |
| Team member name: Konstantinos Megaritis  Team member ID: 271868  ***I herewith express my agreement with the submission of this final version of the group project by the team leader.***    Date: 11/03/2023­  Team member Signature: Konstantinos Megaritis |