MEF UNIVERSITY

SEGMENTATION WITH UNSUPERVISED LEARNING: AN APPLICATION USING THE WALKER'S DATA

Capstone Project

Taylan Polat

İSTANBUL, 2021

MEF UNIVERSITY

SEGMENTATION WITH UNSUPERVISED LEARNING: AN APPLICATION USING THE WALKER'S DATA

Capstone Project

Taylan Polat

Advisor: Prof. Dr. Özgür Özlük

İSTANBUL, 2021

MEF UNIVERSITY

Name of the project: Segmentation with Unsupervised Learning: An Application Using the Walker's Data

Name/Last Name of the Student: Taylan Polat

Date of Thesis Defense: 05/09/2019

I hereby state that the graduation project prepared by Your Name (Title Format) has been completed under my supervision. I accept this work as a "Graduation Project".

05/09/2021
Prof. Dr. Özgür Özlük

I hereby state that I have examined this graduation project by Your Name (Title Format) which is accepted by his supervisor. This work is acceptable as a graduation project and the student is eligible to take the graduation project examination.

05/09/2021

Prof. Dr. Özgür Özlük

Director
of
Big Data Analytics Program

We hereby state that we have held the graduation examination of _____ and agree that the student has satisfied all requirements.

THE EXAMINATION COMMITTEE

Committee Member	Signature
1. Prof. Dr. Özgür Özlük	……..
2. Dr. Tuna Çakar	……..

ACADEMIC HONESTY PLEDGE

I promise not to collaborate with anyone, not to seek or accept any outside help, and not to give any help to others.

I understand that all resources in print or on the web must be explicitly cited.

In keeping with MEF University's ideals, I pledge that this work is my own and that I have neither given nor received inappropriate assistance in preparing it.

Taylan Polat 05/09/2021 Signature

EXECUTIVE SUMMARY

SEGMENTATION WITH UNSUPERVISED LEARNING: AN APPLICATION USING THE WALKER'S DATA

Taylan Polat

Advisor: Prof. Dr. Özgür Özlük

SEPTEMBER, 2021, 31 pages

In this project, the Walkers suitable for the service were filtered by using the dataset shared by the DogGo company. Then, unsupervised machine learning methods such as K-Means, Gaussian, Principal Component Analysis were used to score and cluster the most suitable walkers according to performance, willingness, and experience.

DogGo is the first mobile application in Turkey that provides pet walking and grooming services to its customers in a safe and professional manner. DogGo provides a professional service where dogs are taken care of in dog families' own homes or at the caretaker's home for any need of dog families. DogGo Company wants to provide the best matching of walkers and animals, using Machine Learning algorithms, through a 5-step acquisition process for their walkers.

While the results of the K-means models created on the unique sliders were compared with the help of the Elbow method and the Silhouette score, the results of the Gaussian models were compared with the AIC and BIC method. In addition, an RFM scoring in a classical structure has also been created. When the results of the study were examined considering the Elbow and Silhouette scores, it was shown that the model created with K-Means gave the best results, and the number of clusters was decided as 2.

Key Words: Clustering, K-Means, Gaussian, Principal Component Analysis, AIC and BIC, Elbow Method

ÖZET

DENETİMSİZ ÖĞRENME İLE SEGMENTASYON: GEZDİRİCİ VERİSİNİ KULLANAN BİR UYGULAMA

Taylan Polat

Proje Danışmanı: Prof. Dr. Özgür Özlük

EYLÜL, 2021, 32 sayfa

Bu projede DogGo şirketinin paylaşmış olduğu data üzerinde hizmete uygun gezdiricilerin filtrelenmesinden sonra ilgili müşteriye talepte bulunmuş en uygun gezdiricinin performans, isteklilik ve tecrubesine gore puanlandirilmasi ve kümelenmesi için KMeans, Gaussian, Temel Bileşenler Analizi gibi denetimsiz makine öğrenmesi yöntemleri kullanılmıştır.

DogGo, evcil hayvan sahiplerine güvenli ve profesyonel köpek gezdirme ve bakım hizmeti sağlayan Türkiye'nin ilk mobil uygulamasıdır. DogGo, köpek ailelerinin her türlü ihtiyacı için köpek ailelerinin kendi evlerinde veya bakıcının evinde köpeklerin bakıldığı profesyonel bir hizmet sunmaktadır. DogGo Şirketi, gezdiricileri için 5 aşamalı alım sürecinden geçirip, Makine Öğrenmesi algoritmaları da kullanarak en iyi gezici ve hayvan eşleştirmesini sağlamak istemektedir. Eşsiz gezdiriciler üzerinden oluşturulan K-means modellerinin sonuçları Elbow methodu ve Silhouette puanı yardımıyla karşılaştırılırken, Gaussian modellerinin sonuçları AIC ve BIC methoduyla karşılaştırılmıştır. Ayrıca, klasik yapıda bir RFM skorlaması da oluşturulmuştur. Elbow ve Silhouette puanları dikkate alınarak çalışmanın sonuçları incelendiğinde, K-Means ile oluşturulan modelin en iyi sonuçları verdiğini göstermiş olup, küme sayısı 2 olarak kararlaştırılmıştır.

Anahtar Kelimeler: Kümeleme, K-Means, Gaussian, Temel Bileşenler Analizi, AIC ve BIC, Elbow Yöntemi

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Online pet grooming services, including pet grooming apps, have been increasing in popularity lately. They can also provide you with an experienced person who knows more about pets.

Relying on friends or family to care for pets may not always produce good results because they are not pet care professionals. For this reason, online pet grooming companies continue to improve themselves to improve pet grooming services.

After the 2000s, developments in technology such as smart phones or mobile applications and the difficulty of carrying computers have led to significant developments worldwide (Myles, 2020). DogGo has also become the first mobile application of Turkey, which pet walking and grooming services to its customers in a safe and professional manner by adapting to the times.

The aim of this study is to improve the ability of DogGo company to understand the needs of its customers, to develop customized marketing programs for customers, to find the best match between walkers and dogs by using machine learning algorithms. As of this issue, we can say that the matching system of the DogGo application is like the dating applications. Dating apps use important algorithms that match their members based on their similar specialties (Myles, 2020).

While trying to segment Walkers, it has been progressed as if an ordinary customer segmentation is being created. Segmenting a business customer base into segments of customers with similar and different market characteristics or categories is called customer segmentation (Ezenkwu & Ozuomba, 2015). While segmenting the walkers; We took into related factors such as the frequency of matching of walkers, the score of given by the dog owner, the time of dog spent with the walker, and the location.

While creating walker segmentation, we will train our data with clustering models, one of the unsupervised machine learning algorithms. Clustering: It is an unsupervised machine learning model that is widely used in pattern recognition, medicine or computer science, which is used to divide observations, features or data groups into specific groups (Ezenkwu & Ozuomba, 2015).

1.1. Literature Review on Segmentation with Unsupervised Learning: An Application Using the Walker's Data

While creating the segmentation models, a walker segmentation was first created over the RFM scoring. For walkers, the total number of days was used as the recency parameter, the total number of walks as the frequency parameter, and the lifetime variable as the monetary parameter. The RFM study provides to divide customers or employees into behavioral groups similarities based on recency, frequency, and monetary values. It is thought that segmentation studies increase customer revenues. At the same time, it is believed that retaining existing customers is more important than acquiring new customers (Christy et al., 2018).

As another segmentation method, it was decided to create walker segmentation by using the K-Means and Gaussian Mixture Model machine learning algorithms. K-Means algorithm, taking parameters and number of clusters as input, provides a defined number of partitions with high similarity and characteristics within the clusters. K-Means algorithm is an iterative approach that calculates the value of centroids before each iteration, moving data points between different clusters based on the centers calculated at each iteration. (Christy et al., 2018)

Another applied unsupervised machine learning, Gaussian Mixture Model (GMM) is a data clustering method, and it can both be seen as a linear combination of different Gaussian components and divide the observations into different components by predicting the parameters of each cluster (Liu at al.).

The Gaussian mixture model is constructed by combining multivariate Gaussian distributions, each with different mean and covariance, and each of these component distributions is a set of distribution (Bouman, 2005). Clustering algorithms can be expressed in two groups as model and similarity-based. Similarity-based clustering algorithms, taking as a basis the similarity function between the data observations, model-based methods, uses mixtures of distribution to fit the data (Liu at al.).

1.2. Literature Review on Semi-supervised Learning: An Application Using the Walker's Data

We aim to develop a semi-supervised learning methodology to demonstrate classification performance using the labeled data in the "Puan" feature in the dataset shared by the DogGo team. For this reason, we tried to predict missing labels using the Decision Tree and KNN models. Considering that it has no significant contribution to the study, this feature was not included in the model at the final stage.

Decision tree models are one of the most useful models in the field of data mining as they provide reasonable accuracy for possible outcomes compared to other models and are relatively inexpensive to calculate. Most decision tree classifiers consist of two stages, Tree generation and Tree pruning, and recursively dividing the training data according to the most appropriate criterion until all or most of the records for each segment are labeled with the same class. In decision trees, tree pruning is used to prune leaves and branches (Du & Zhan, 2002).

In the K Nearest Neighbor algorithm, an object is classified according to its properties or the distances of its neighbor points, and this object is assigned the largest class among its k nearest neighbors. The input value consists of the closest trained k samples in the feature space. In this case, if k is 1, the object is simply assigned to the nearest neighbor class. We can say that it is a non-parametric method for K Nearest Neighbor algorithm features and is used in both classification and regression problems (Suyambu, 2017).

1.3 Literature Review on Applied Measurement Metrics and Data Cleaning Methods: An Application Using the Walker's Data

While the Shapiro–Wilk test is a more suitable method for small sample sizes, it can also be applied to larger samples, while Kolmogorov–Smirnov test is used for cases where the sample is larger than 50. For both tests, the null hypothesis states that the data are taken from the normally distributed population. When the P value is greater than 0.05, the null hypothesis is accepted, and the data are considered normally distributed (Mishra et al., 2019).

Chi square test used to compare the segments created because of the model with the "Puan" feature, represents the different states "Puan" on the rows, the states of clusters variable on the columns, and the cells contain integer occurrences in that state row and column of the this two variables. It is a statistical test that uses probability tables to compare different populations in terms of frequency at which different species are represented (Barceló, 2018).

When the boxplot for the data is plotted, the whiskers in the boxplot represent the minimum and maximum values when they are within 1.5 times the IQR value from both ends of the box, while values 1.5 and 3 times the IQR value are considered outliers (Mishra et al., 2019).

One of the methods applied while determining the number of clusters is the Elbow Method. The elbow method is based on the idea that the number of clusters should be determined by looking at the percentage of variance explained for the number of clusters, so that adding another cluster does not result in better modeling of the data. Because the first clusters add so much information, at some point the marginal gain drops significantly and an

angle forms on the chart. At this point, the "elbow criterion" is determined by choosing the correct "k", that is, the number of clusters (Bholowalia & Kumar, 2014)

The other method, Silhouette index does not need a training set to evaluate the clustering results, making it more suitable for the clustering task. The value of the silhouette width can range from -1 to 1. The larger the (positive) value of an element, the higher the probability of clustering in the right group (Shutaywi & Kachouie, 2021)

2. PROJECT DEFINITION

DogGo is the first mobile application in Turkey that provides pet walking and grooming services to its customers in a safe and professional manner. They are currently serving in various districts of Istanbul. DogGo was founded by Mehmet Oğul Gürsoy and Ömercan Dede in Istanbul in June 2017, and its mobile application became active on January 3, 2019. Rover and Wag are among the most well-known dog walking companies in the world. DogGo is in the same market area as these companies. Within the scope of this capstone project, a segmentation and prediction study will be carried out on the existing walker data of DogGo company. The aim of this project is to achieve the best results when matching DogGo's own walkers with the dogs of its customers. Segmentation and prediction studies were carried out by using "walks.csv" and "walkerParameters.csv" files shared by DogGo company. The "walks.csv" file consists of 55541 rows and 35 features, while the "walkerParameters.csv" file consists of 2198 rows and 48 features. When the files were examined, it was observed that the number of unique walkers was 1382. First, before starting the segmentation and prediction models, data cleaning, data preparation and data analysis processes were carried out.

3. EXPLORATORY DATA ANALYSIS AND PREDICTING CLUSTERS

3.1. Data Description

The Walkerparameters data shared by the DogGo company used in this project consists of 2198 rows and 40 columns. Each Walker has its own unique ID. In the table below in Table1 and in Table2, you can see the names and types of some variables in the Walkerparameters data.

Table 1: Variable Types

Variables	**Variable Types**
gender	**Categorical**
walkerType	
totalWalks	**Numerical**
todayFirstWalk	
todayLastWalk	
firstWeekWalks	
firstMonthWalks	
averageEarlyFinished	
averageBadDistance	
averageLateStart	
averageDistance	
lifetime	
averageWalking	
feedbackAverage	
comWithDogAvg	
comWithMeAvg	
timeAccuracyAvg	
feedbackCount	
lastDemand	
totalDemands	
firstDemand	
todayLastDemand	
todayFirstDemand	
firstWeekDemands	
firstMonthDemands	
lastWeekDemands	
lastMonthDemands	
avgmeetingrate	
avgsittingrate	
avgwalkingrate	
activeLifeTime	
activeDaysCount	
differentDistrictCount	
dailyWalkAverage	
walkFrequency	
negativeFeedbackCount	
negativeFeedbackRatio	
AdHoc	
Planned	
Package	

Table 2: Variable Names and Definitions

Variable Names	Variable Definition
Id/Walkerıd	each walker's unique id
Gender	gender of a walker
Signuptime	sign up time of walker
Lastwalk	last walk time of a walker
Firstwalk	first walk time of a walker
Totalwalks	total number of walkings done by walker
Todayfirstwalk	time difference between today and first walk time (Milliseconds)
Todaylastwalk	time difference between today and last walk time (milliseconds)
Firstweekwalks	number of walks done within the 7 days after sign up
Firstmonthwalks	number of walks done within 30 days after sign up
Serveddogs	ids of each dog that a walker serves
Averageearlyfinished	early finish is the walk that finishes before Duration(45) - 3 minutes - averageEarlyFinished is total numbers of earlyFinishedWalks / totalWalks
Averagebaddistance	badDistance is more than 8 minutes and duration is less than 1.5 km averageBadDistance is total number of badDistanceWalks / total Walks
Averagedistance	total distance done by walker / totalWalks
Lifetime	the time between signup time and lastWalk time (Days)
Averagewalking	total number of walks/lifetime (think it as total walks in one day)
Totaldemands	total number of demands that a walker applies for orders(walks)
Lastdemand	last demand time of a walker applies for an order(walk)
Firstdemand	first demand time of a walker applies for an order
Todaylastdemand	time between today and last demand time (Milliseconds)
Todayfirstdemand	time between today and first demand time (milliseconds)
Firstweekdemands	total number of demands within 7 days after signup time
Firstmonthdemands	total number of demands within 30 days after signup time
Feedbackcount	number feedbacks given for specific walker
Feedbackaverage	after each walk, the owner gives 3 feedbacks communication with me, communication with dog and time accuracy the mean of these 3 columns creates the walkerFeedbacks feedbackAverage is the average of feedback values

Data includes additional Walker Score shared by DogGo Company. Here, it was seen that only 205 Walker scores were assigned out of 2198 observations. Score values missing at this stage will be updated with the help of Semi-Supervised in the future but, this filling process was not used in the final segmentation stage. Distribution of "Puan" feature is shown in Figure 1:

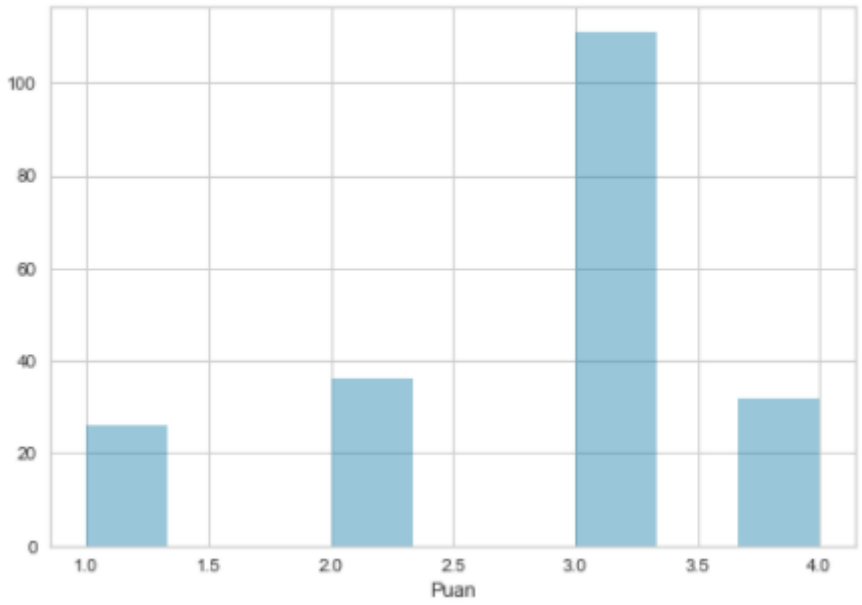

Figure 1: Distribution of "Puan" feature

When the Walks data about the walks shared by the DogGo Company is examined, it was observed that it mostly takes place in Kadiköy, Besiktas, Sisli districts. It is seen that the rate of use is high due to the high awareness and the size of the districts due to Kadiköy, Besiktas,Sisli being the first districts where the DogGo company was opened. Distribution of distance shown in Figure 2:

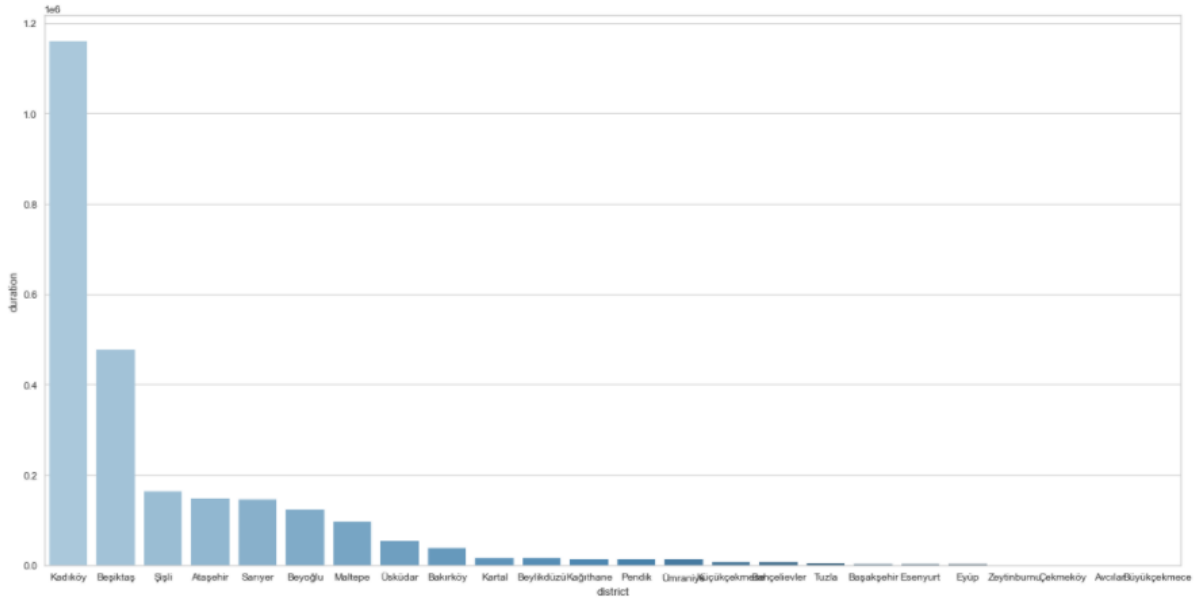

Figure 2: Distribution of Distance for Districts

At the same time, it was observed that the most walking distance was in cloudy, clear, and rainy weather, respectively. It can be said that less walking of dogs in open weather due to hot weather causes dog walking times to be longer in cloudy weather. Distribution of weather and distance relationships shown in Figure 3:

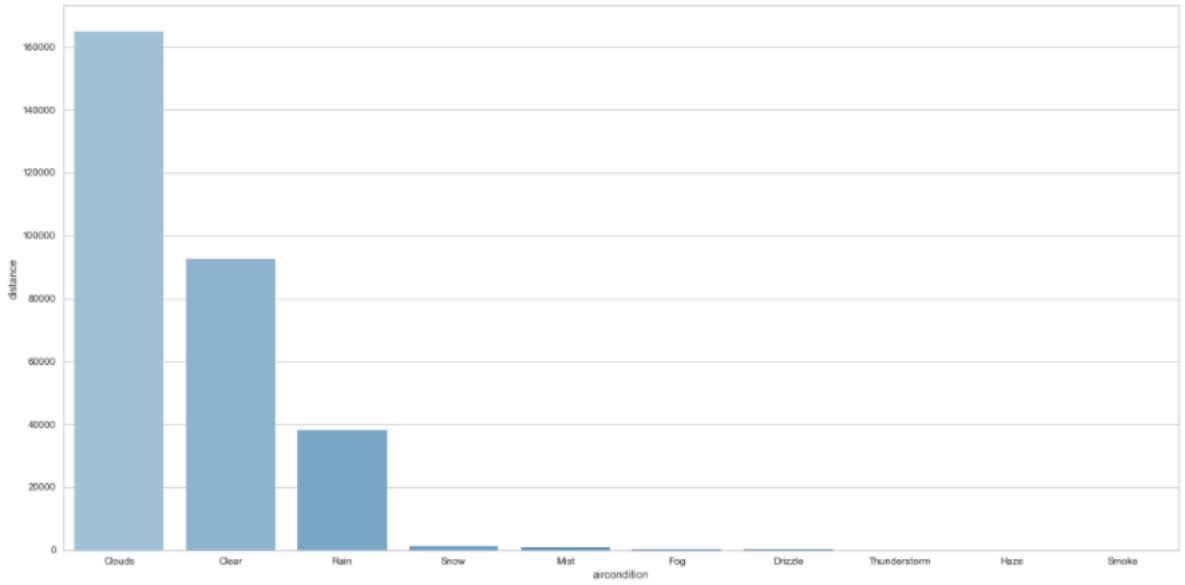

Figure 3: Distribution of Weather & Distance

Figure 4 shows the relationship between total distance and total time spent quarterly. While the y-axis on the left represents Distance, the line, that is, the y-axis on the right, represents Spending Time. It can be seen that the time and distance spent change each year depending on the seasons, and we can say that it increases as time progresses.

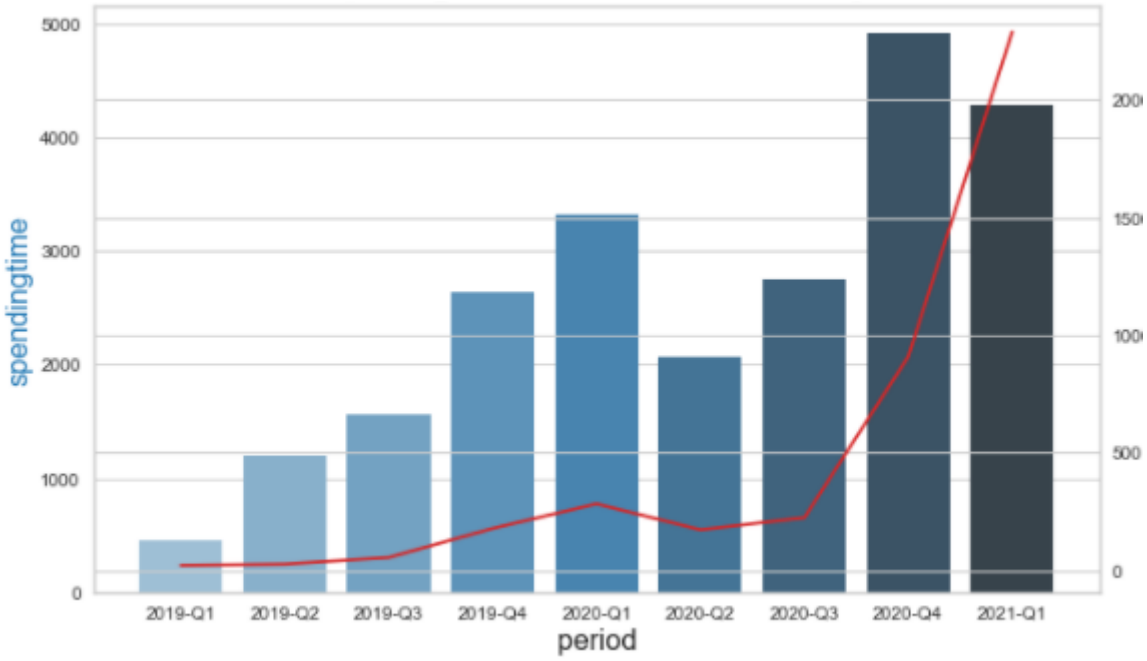

Figure 4: Time period of Distance

3.2. Data Preparation

Features with a missing value of 80 percent or more were excluded from the data. While filling in the missing observations, the distribution of the variable was examined. It was observed that "feedbackAverage", "comWithDogAvg", "comWithMeAvg", "timeAccuracyAvg" feature are binary variables, and zero is given for missing values. The distribution of “feedbackAverage” variable is shown in Figure 5. While filling the missing values in the other variables in the data, filling with the mean and median was tried, but due to the high number of missing observations, preference was made over the Mice and KNN imputation methods. In order to apply Mice and Knn imputation,"totalDemands","todayLastDemand", "todayFirstDemand", "firstWeekDemands","firstMonthDemands","Puan" features were deemed appropriate.

Knn and Mice imputation were performed separately for each feature and when the data distribution was examined, it was decided to proceed with Mice imputation. As an example,

Figure 6 and Figure 7 show the distribution after mice imputation for the "totalDemands" and "firstMonthDemands" features.

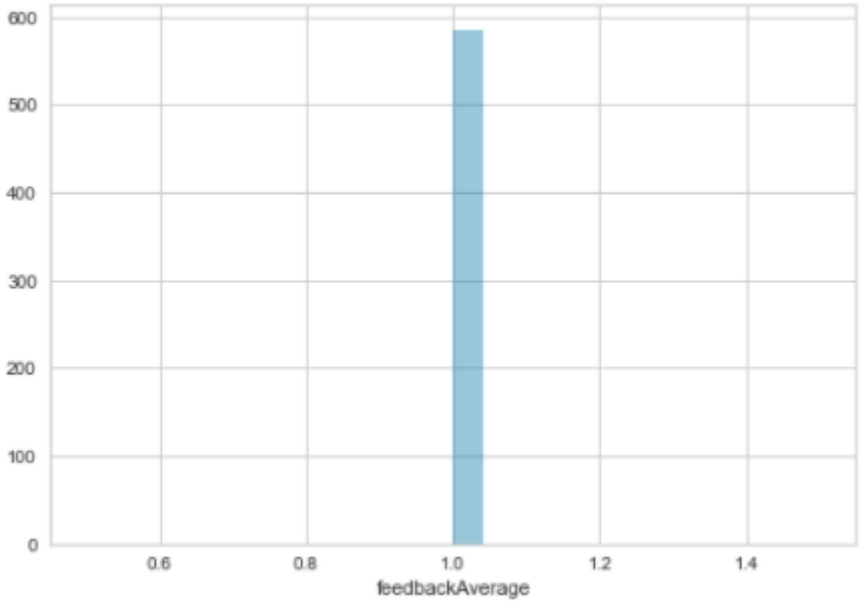

Figure 5: Binary Missing Imputation

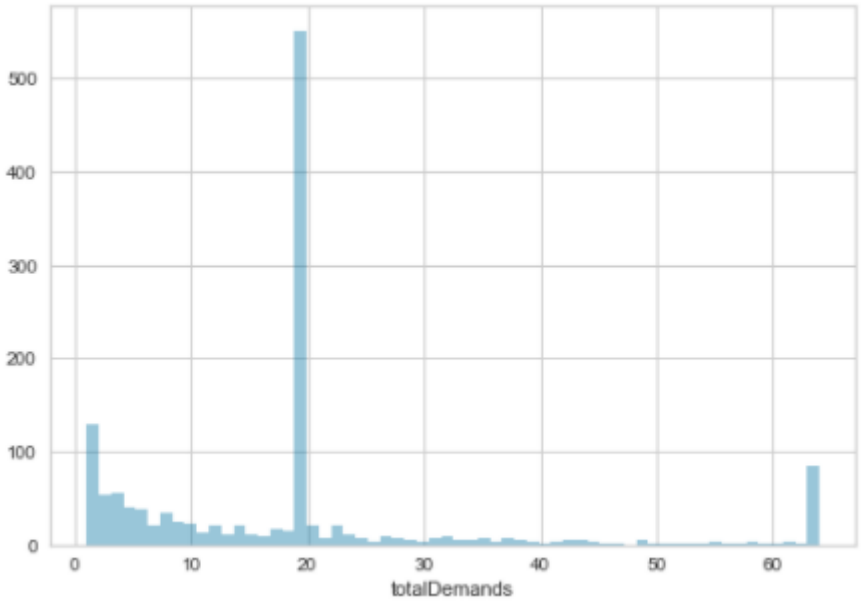

Figure 6: Mice Imputation for “totalDemands” Feature

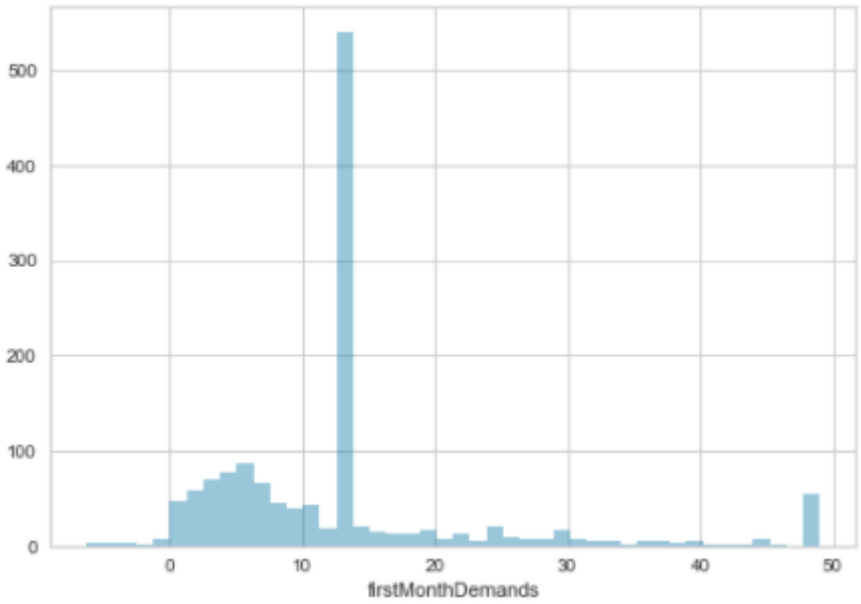

Figure 7: Mice Imputation for “firstMonthDemands” Feature

"Unnamed:0.1.1.11","Unnamed:0.1.1.1.1.1","applicantid""Unnamed:0.1","Unnamed:0.1.1","Unnamed:0.1.1.1","lastWalk","firstWalk","todayFirstWalk","servedDogs","lastDemand","firstDemand","avgsittingrate","lastWeekDemands","Unnamed:0" are excluded from the data because they do not make any sense for the model.

In order to create the Frequency variable required for the RFM table to be created, firstly, the Spendingday variable was created with the difference of the lastwalk variable by taking the latest date from the checkintime date in the Walk table. For the numerical variables in the dataset, the IQR method was used to handle the outliers. If the outlier observation is above the determined top value, the up_limit value is determined because of IQR, and if it is below the low limit, the low_limit value is used. Boxplot, histogram, and distribution graphs of "totalWalks" and "firstWeekWalks" variables are given below as an example. The IQR method was applied for the variables with outlier observations such as the variables in the example below.

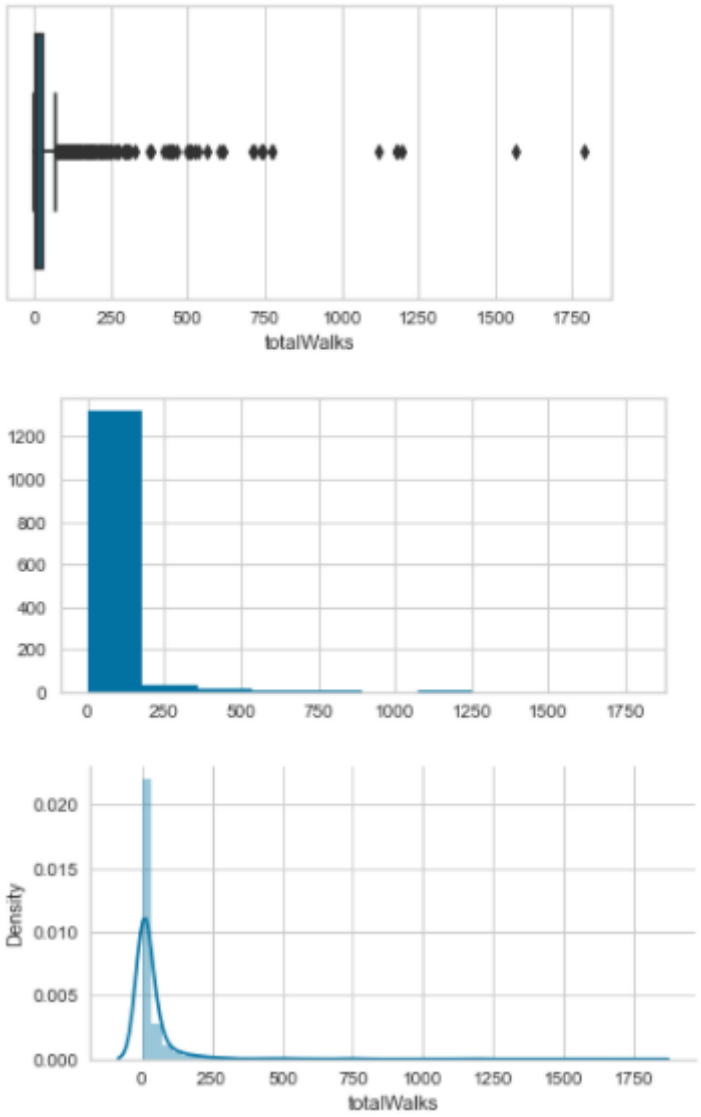

Figure 8: Histogram, Boxplot, Distribution charts of the “totalWalks” Feature

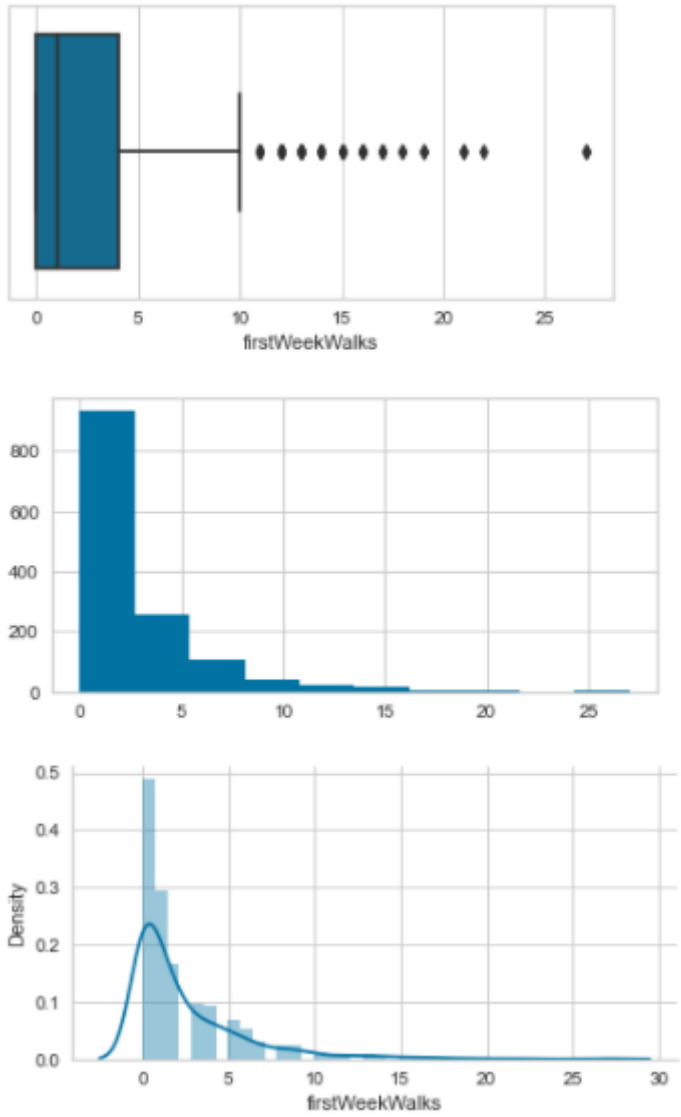

Figure 9: Histogram, Boxplot, Distribution charts of the "firstWeekWalks" Feature

Considering that the "Puan" feature may contribute to the models, unknown "Puan" features were tried to be predicted by using the KNN model for missing observations. Since the "Puan" feature was distributed in an imbalanced way, Smote was first applied to the variable, then this process was abandoned because the model success was less than expected. Model Accuracy results according to the K variability in the KNN model for the "Puan" feature are shown in Figure 10.

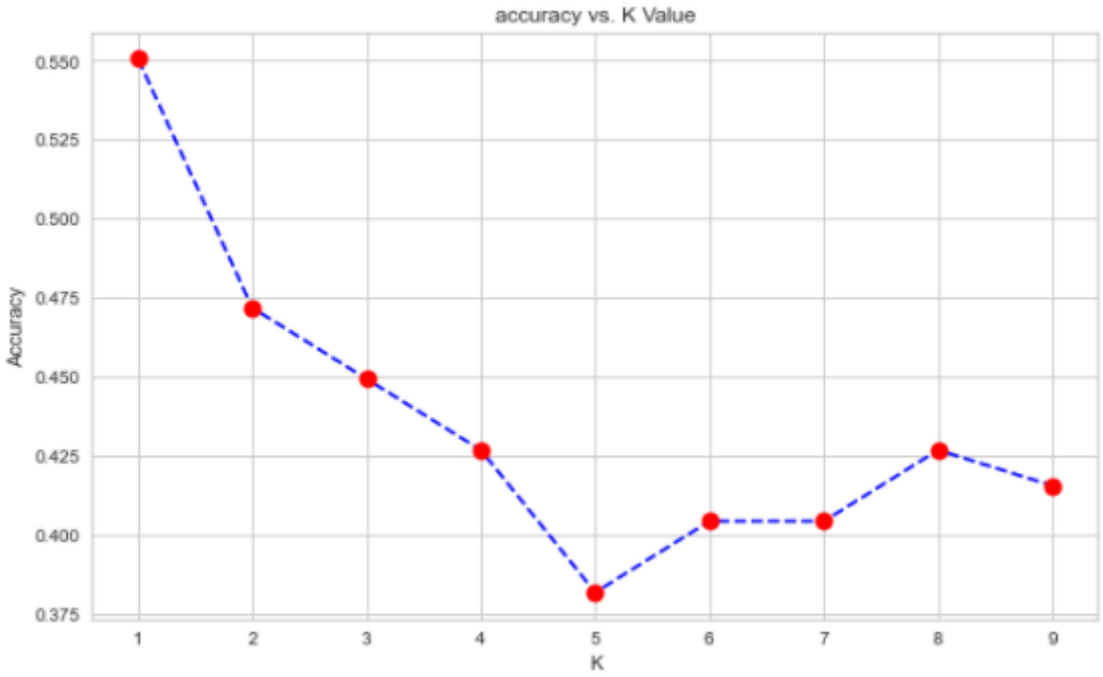

Figure 10: KNN model for Smote applied “Puan” feature

As a result of the negotiations with the DogGo company, while creating the Walker segmentation, it was decided to continue working with both an RFM strategy and the model that gave the most successful results using Machine Learning methods.

In order to observe the distributions of the created clusters in 2D and 3D, functions named silandscat and treed_clustering were created. A new data named “RFMSeg” was created by using the “spendingday” variable for the “Recency” value, the “totalWalks” variable for the “Frequency” value, and the “lifetime” variable for the “Monetary” value. In cases where the RFM study was performed, values greater than 0 were used for the "Recency" and "Monetary" values. It is applied for recency, frequency, and monetary parameters by specifying 3 quantiles (0.25,0.50.0.75). For the recency parameter, a class of 1 is determined for values less than 0.25 and 0.25, 2 for values less than 0.5 and greater than 0.25, 3 for values less than 0.75 and greater than 0.5, and 4 for the remaining values. For frequency and monetary parameters, 4 for values less than 0.25 and 0.25, 3 for values less than 0.5 and greater than 0.25, 2 for values less than 0.75 and greater than 0.5, and 1 for remaining values are determined.

Table 3: RFM Quantiles

	recency	frequency	monetary
0.25	71.286458	3.0	13.168663
0.50	192.843750	8.0	42.275486
0.75	401.177083	28.0	135.067326

Then, these determined classes are brought together to form the RFM Score. The head of five rows of the RFM scoring are shown in the Table 4.

Table 4: First 5 rows for RFM Table

walkerid	recency	frequency	monetary	R_Quartile	F_Quartile	M_Quartile	RFMScore
00162fd3-21b4-4371-bebe-db6b081d86ef	87.927083	15.0	134.260741	2	2	2	222
0082cfc9-ce2f-47b1-a1ae-5e0c45f903b1	98.177083	7.0	21.000671	2	3	3	233
00e99cd8-9886-4808-8266-7080bcf2ac88	18.052083	2.0	16.027176	1	4	3	143
00fef613-662c-4194-8de1-164942e25fb2	403.298634	25.0	37.355498	4	2	3	423
01adeee1-cb3d-46ff-9514-f4797299698c	0.315729	19.0	44.684572	1	2	2	122

The correlations of the variables in the data prepared for the RFM study were examined. It was observed that the variables of Recency, Frequency and Monetary were not highly correlated with each other. It is desired to develop a different approach by making a clustering with K-Means using the Recency, Frequency and Monetary variables in the RFM data. For this reason, Shapiro-Wilk Test was applied for Recency, Monetary and Frequency variables first. Since the p-values obtained as a result of the test were less than 0.05, Minmax Scaling process was applied.

When the Gaussian Mixture Model algorithm is applied as a second method and the results are examined, the distribution of AIC and BIC results according to segments is shown in Figure 11:

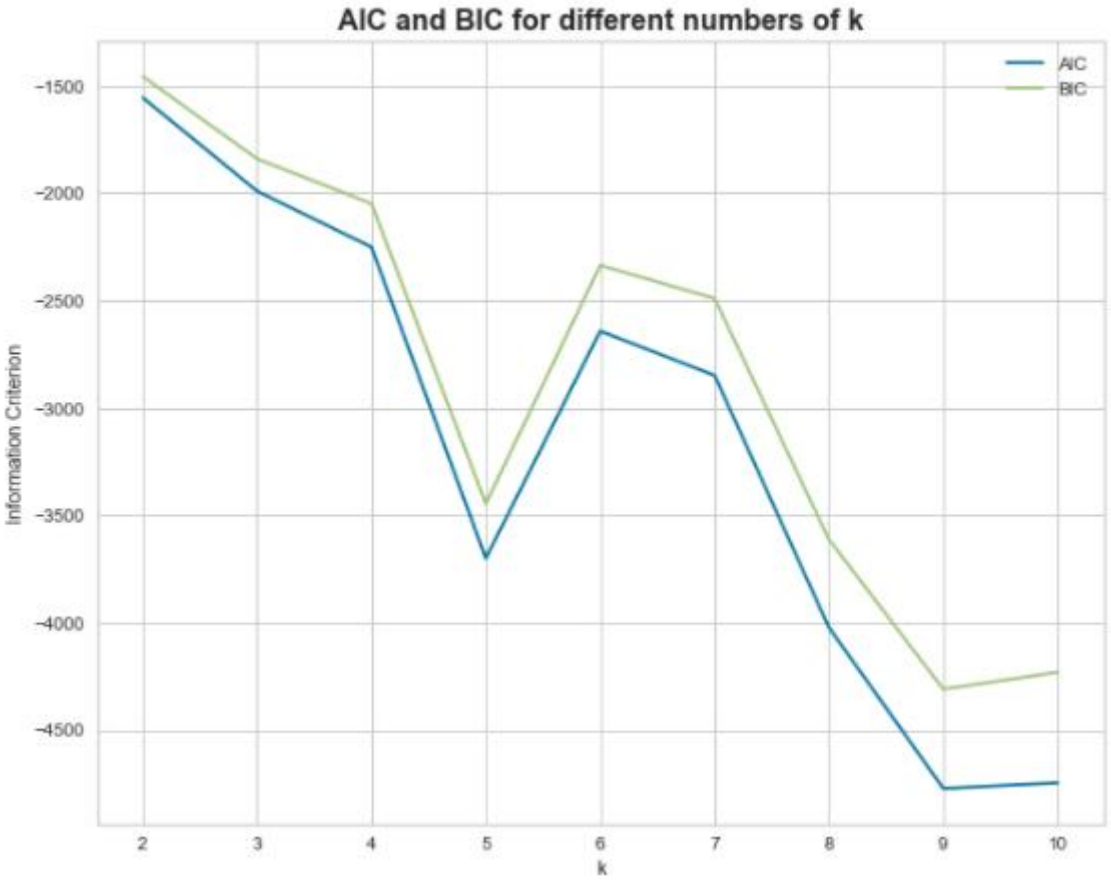

Figure 11: AIC and BIC Scores for Actual Data

Considering that the ["todayLastWalk", "averageEarlyFinished", "averageBadDistance","averageLateStart","feedbackAverage","comWithDogAvg","comWithMeAvg", "timeAccuracyAvg", "todayLastDemand", "todayFirstDemand", "walkFrequency", "Package", "spendingday", "gender_Male", "gender_Other","walkerType_TypeA", "walkerType_TypeB", "walkerType_TypeC"] features may disrupt the segment structure in order to increase the significance of the data, these features were excluded from the Mice Imputed data. Before applying the K-Means algorithm, which was used as the last method, it was decided to use Minmax Scaling and PCA methods on the Mice Imputed data. The chart in Figure 12 was taken as a basis while determining the component because of PCA application. The number of components with approximately 0.90 explanatory power is taken as 8, and the number of components is given as 8 when applying PCA to the dataset.

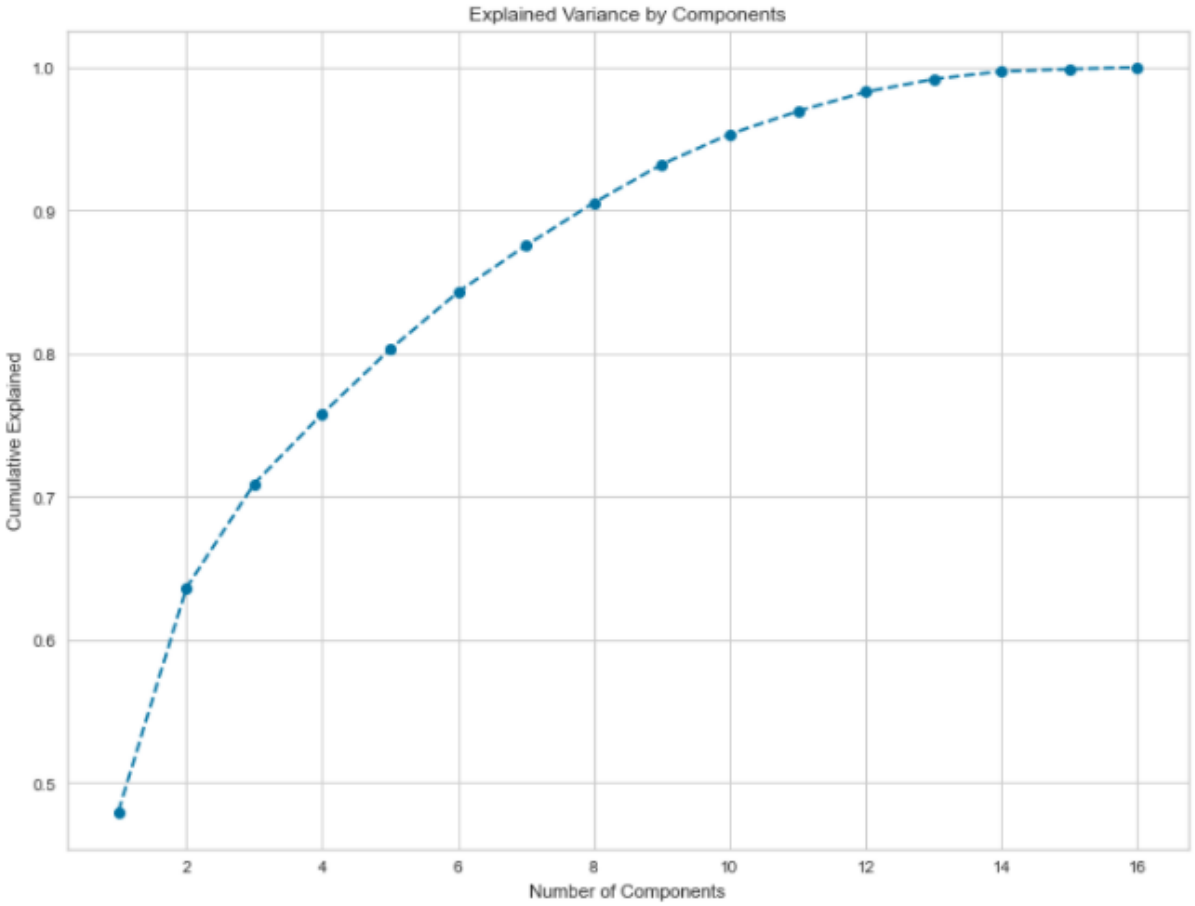

Figure 12: Explained Variance for PCA

Before creating segments with K-Means in the PCA applied data, the number of segments was tried to be reached by applying Silhouette and Elbow methods. Silhouette scores are shown in Figure 13 and Elbow graph is shown in Figure 14. When we examined it with the Elbow method, it was decided to create 2 clusters because it was broken very sharply in the 2nd cluster, and the highest score was obtained in the 2nd cluster in the Silhouette scoring.

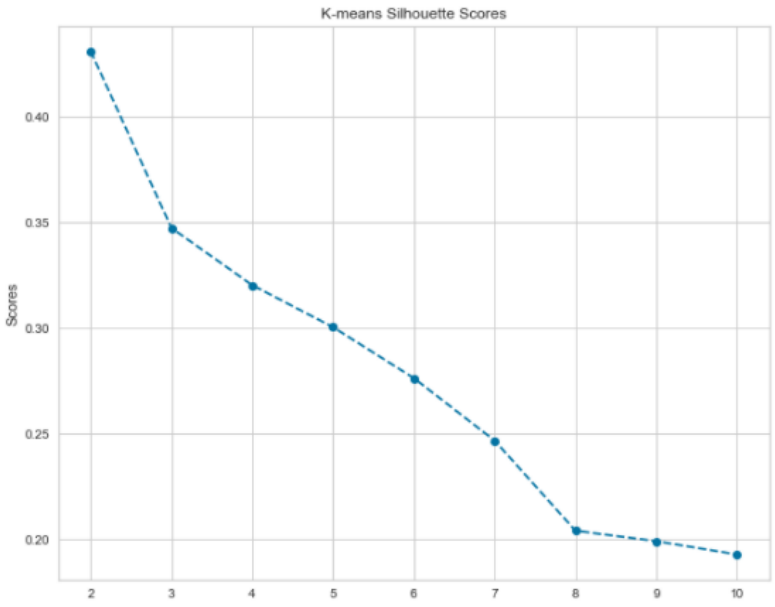

Figure 13: Silhouette Scores

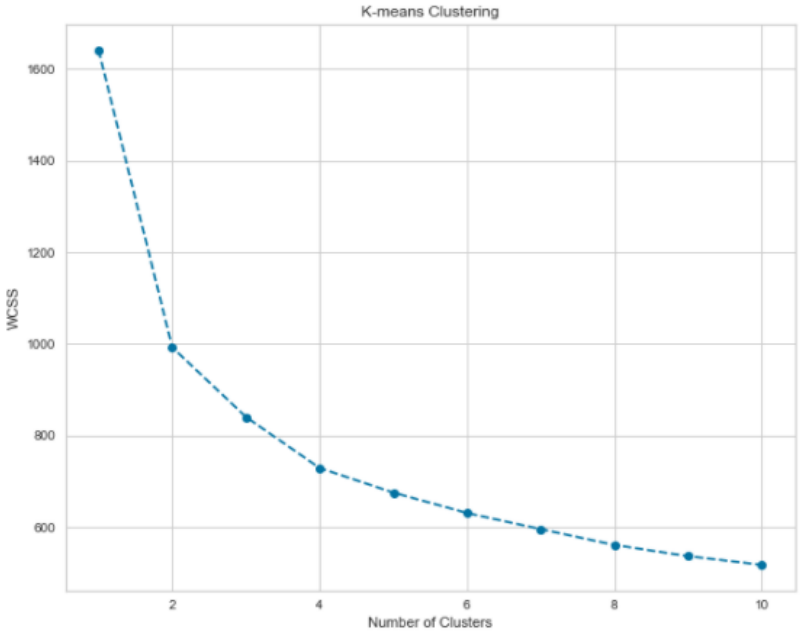

Figure 14: Elbow Chart

When the distribution of the "Score" variable shared by the DogGo team with the clusters created was examined, it was observed that the Active Walkers got the most 3 points. The distribution chart is shown in Figure 15.

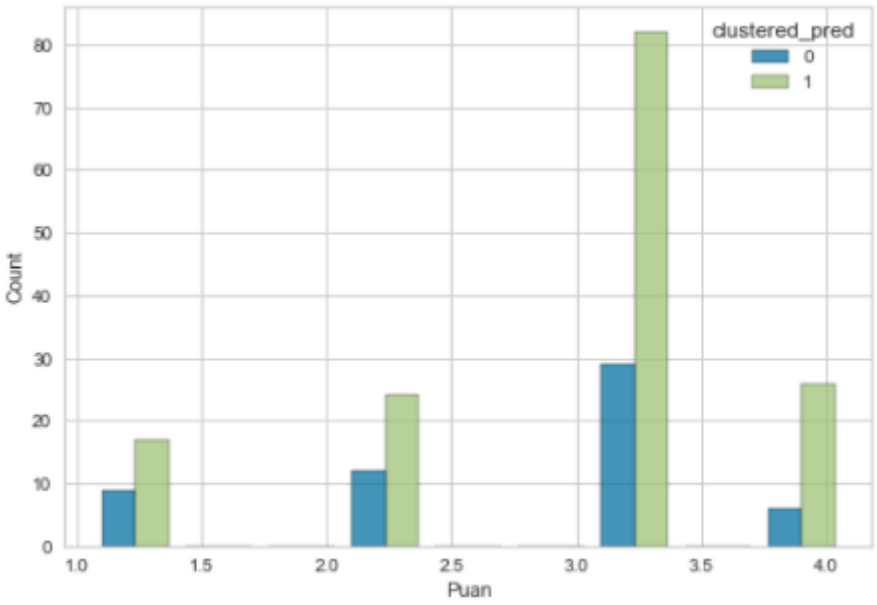

Figure 15: Distribution of "Puan" and 2 Clusters

In order to examine whether there is a difference between the score variable and the determined clusters, the Chi 2 test was applied using Python at a confidence interval of 0.95. As a result of the test, we rejected the H0 hypothesis because the p-value was 0.45. Thus, we can not say there is no significant differences.

Table 5: Chi 2 Results

H0: There is no significant difference between distributions
H1: There is significant difference between distributions

CHI 2 RESULTS	
chi2 statistic	2,62
degrees of freedom	3,00
p-value	0,45

4. CONCLUSION

In this project, the walkers suitable for the service were filtered by using the dataset shared by the DogGo company. Then, unsupervised machine learning methods such as K-Means, Gaussian Mixture Model, and classic RFM strategy were used to score and cluster the most suitable walkers according to performance, willingness, and experience.

Firstly, dataset analysis and EDA processes are applied for this study. Afterwards, Mice Imputation was applied to handle missing values in the data. The IQR method was used to handle the outliers based on the data distributions after the missing imputation. After the data preparations were completed, classical RFM study was carried out and segments were tried to be prepared according to RFM Scores. At the same time, KNN Classifier and Decision Tree Classifier were tried to be used as Semi-Supervised Machine Learning Methods to fill in the missing observations in the Score variable shared by the DogGo team. Considering that the contribution of the imputation process to the model is low, it was decided to work with K-Means and Gaussian Mixture Models.

As a result of weekly meetings with the DogGo team and the determined studies, it was decided that the model created with Machine Learning and K-Means algorithm was more successful in creating segments.

Within the scope of this project, the success of various models has been achieved with Unsupervised Machine Learning methods; active progress has been achieved in processes such as data compliance, data cleaning and data preparation. In order to further develop the project, the size of the data used can be enlarged, different variables can be included in the use for the model, and the results obtained can be supported by fieldwork.

APPENDIX

```
import pandas as pd
import numpy as np
import seaborn as sns
pd.options.display.max_columns = None
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
%matplotlib inline
from scipy import stats
import datetime as dt
from datetime import datetime
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.figure_factory as ff
import matplotlib.pyplot as plt
import plotly.offline as py
from feature_engine.outliers import Winsorizer
from statsmodels.graphics.gofplots import qqplot
from sklearn.preprocessing import StandardScaler,MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_samples, silhouette_score
from yellowbrick.cluster import SilhouetteVisualizer
from mpl_toolkits.mplot3d import Axes3D
from sklearn.decomposition import PCA
from scipy.stats import stats
from scipy import stats
from scipy.stats import norm, skew
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
```

```
from sklearn.model_selection import cross_val_score
from sklearn import svm
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import StratifiedKFold

walkerparameters =
pd.read_csv(r"C:\Users\taylan.polat\Desktop\TP\doggo\walkerParameters.csv")

def showmissing(df):
null_values = df.isnull().mean().sort_values(ascending = False)
null_values = pd.DataFrame(null_values)
plt.figure(figsize = (12,16))
plt.barh(null_values.index,null_values.iloc[:,0].values , align='center', alpha=0.5)
plt.yticks(null_values.index)
plt.xlabel('Missing')
plt.title('Missing Degree')

showmissing(walkerparameters)

walkerscores = pd.read_csv(r"C:\Users\taylan.polat\Desktop\TP\doggo\walkerScores.csv")
walkerscore = walkerscores[["walkerid","Puan"]]
walkerparameters = pd.merge(walkerparameters,walkerscore,on = "walkerid",how = "left")
walk = pd.read_csv(r"C:\Users\taylan.polat\Desktop\TP\doggo\walks.csv")
maxcheckintime = max(walk["checkintime"])
Puan = walkerparameters["Puan"].dropna().describe()
sns.distplot(walkerparameters["Puan"].dropna(),kde = False)

fig = plt.figure(figsize = (8, 4))

# creating the bar plot
plt.bar(["mean","std"],[Puan[["mean","std"]][0],Puan[["mean","std"]][1]] , color ='blue',
width = 0.4)
```

```
plt.xlabel("Mean & Std")
plt.ylabel("Values")
plt.title("Puan")
plt.show()

walkerparameters.drop(["avgsittingrate","lastWeekDemands","Unnamed: 0"],axis = 1,inplace = True)

walkerparameters['lastWalk'] = pd.to_datetime(walkerparameters['lastWalk'], format='%Y-%m-%d %H:%M:%S.%f')
walkerparameters['firstWalk'] = pd.to_datetime(walkerparameters['firstWalk'], format='%Y-%m-%d %H:%M:%S.%f')

maxcheckintime = pd.to_datetime(maxcheckintime, format='%Y-%m-%d %H:%M:%S.%f')

walkerparameters["spendingday"] = maxcheckintime - walkerparameters["lastWalk"]

walkerparameters["spendingday"] = walkerparameters["spendingday"].astype('timedelta64[s]')
walkerparameters["spendingday"] = walkerparameters["spendingday"].astype(float)/86400

walkerparameters = walkerparameters[walkerparameters["totalWalks"].notna()]

def outlier_thresholds(dataframe, variable):
quantile_one = dataframe[variable].quantile(0.25)
quantile_three = dataframe[variable].quantile(0.75)
interquantile_range = quantile_three - quantile_one
up_limit = quantile_three + 1.5 * interquantile_range
low_limit = quantile_one - 1.5 * interquantile_range
return low_limit, up_limit

def replace_with_thresholds(dataframe, variable):
low_limit, up_limit = outlier_thresholds(dataframe, variable)
```

```
dataframe.loc[(dataframe[variable] < low_limit), variable] = low_limit
dataframe.loc[(dataframe[variable] > up_limit), variable] = up_limit

for val in walkerparameters.columns:
if (walkerparameters[val].dtype != 'O') and (walkerparameters[val].dtype != '<M8[ns]'):
replace_with_thresholds(walkerparameters,val)

rfm = walkerparameters[["walkerid","spendingday","totalWalks","lifetime"]].groupby(["walkerid"])
.sum()

rfm.columns = ["recency","frequency","monetary"]

rfm = rfm[(rfm["recency"] > 0) & (rfm["monetary"] > 0)]



walkerparameters.drop(["Unnamed: 0.1","Unnamed: 0.1.1","Unnamed: 0.1.1.1",
"Unnamed: 0.1.1.1.1","Unnamed: 0.1.1.1.1.1","applicantid",
"lastWalk","firstWalk","todayFirstWalk","servedDogs","lastDemand",
"firstDemand"],axis = 1,inplace = True)

walkerparameters.drop(["signuptime","walkerid"],axis = 1,inplace = True)

plt.figure(figsize = (20,15))
s = sns.heatmap(rfmSeg.corr(),
annot = True,
cmap = "viridis",
vmin = -1,
vmax = 1)
s.set_yticklabels(s.get_yticklabels(),rotation = 0,fontsize = 12)
s.set_xticklabels(s.get_xticklabels(),rotation = 90,fontsize = 12)
plt.title("Correlation Heatmap")
```

```
plt.show()

walkerparameters.drop(["lastMonthDemands","avgmeetingrate","avgwalkingrate",
"negativeFeedbackCount","negativeFeedbackRatio"],axis = 1,inplace = True)
walkerparam = walkerparameters.copy()
shapiro_res = []

for col in walkerparameters.columns:
res = stats.shapiro(walkerparameters[col].fillna(0)).pvalue
shapiro_res.append(res)

from impyute.imputation.cs import mice

imp_var = ["totalDemands","todayLastDemand","todayFirstDemand",
"firstWeekDemands","firstMonthDemands","Puan"]

imp_var_value = pd.DataFrame(mice(walkerparam[imp_var].values),columns = imp_var)

walkerparam = walkerparam.reset_index()
walkerparam.drop("index",axis = 1,inplace = True)
walkerparam[imp_var] = imp_var_value

walkerparam["feedbackAverage"] = walkerparam["feedbackAverage"].fillna(0)
walkerparam["comWithDogAvg"] = walkerparam["comWithDogAvg"].fillna(0)
walkerparam["comWithMeAvg"] = walkerparam["comWithMeAvg"].fillna(0)
walkerparam["timeAccuracyAvg"] = walkerparam["timeAccuracyAvg"].fillna(0)
walkerparam.drop("Puan",axis = 1, inplace = True)
walkerparam = pd.get_dummies(walkerparam, drop_first=True)
walkerparam.drop(["todayLastWalk", "averageEarlyFinished", "averageBadDistance",
"averageLateStart",
"feedbackAverage", "comWithDogAvg", "comWithMeAvg", "timeAccuracyAvg",
"todayLastDemand",
```

```
"todayFirstDemand", "walkFrequency", "Package", "spendingday", "gender_Male", "gender_Other",
"walkerType_TypeA", "walkerType_TypeB", "walkerType_TypeC"],axis = 1,inplace = True)

walkerparam.drop(["lastMonthDemands","avgmeetingrate","avgwalkingrate",
"negativeFeedbackCount","negativeFeedbackRatio"],axis = 1,inplace = True)

walkerparam = walkerparam.fillna(0)

scaler = MinMaxScaler()
scaler.fit(walkerparam)
pca_walkerdata = scaler.transform(walkerparam)
pca = PCA()
pca.fit(pca_walkerdata)
plt.figure(figsize = (12,9))
plt.plot(range(1,17),pca.explained_variance_ratio_.cumsum(),marker = "o",linestyle = "--")
plt.title("Explained Variance by Components")
plt.xlabel("Number of Components")
plt.ylabel("Cumulative Explained")

pca = PCA(n_components = 8)
pca.fit(pca_walkerdata)
segmentation_std = pca.fit_transform(pca_walkerdata)

plt.figure(figsize = (10,8))
plt.plot(range(2,11),sscore(2,11,segmentation_std),marker = "o",linestyle="--")
plt.xlabel("Number of Clusters")
plt.ylabel("Scores")
plt.title("K-means Silhouette Scores")
plt.show()

wcss = []
for i in range(1,11):
```

```
kmeans = KMeans(n_clusters = i, init = "k-means++",random_state = 42)
kmeans.fit(segmentation_std)
wcss.append(kmeans.inertia_)

plt.figure(figsize = (10,8))
plt.plot(range(1,11),wcss,marker = "o",linestyle="--")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
plt.title("K-means Clustering")
plt.show()

kmeans = KMeans(2)
kmeans.fit(segmentation_std)
df_norm_clustered = walkerparameters.copy()
df_norm_clustered["clustered_pred"] = kmeans.predict(segmentation_std)

df_norm_clustered.groupby("clustered_pred").agg(['mean','std'])

df_norm_clustered[["Puan","clustered_pred"]].groupby("Puan").count()

chi_table = pd.pivot_table(df_norm_clustered, values='totalWalks',
index=['Puan'],columns='clustered_pred', aggfunc='count')

results_zero = list(chi_table[0])
results_one = list(chi_table[1])

result_table_last = np.array([results_zero,results_one])

from scipy import stats
chi2_stat, p_val, dof, ex = stats.chi2_contingency(result_table_last)

sns.histplot(data=df_norm_clustered, x="Puan", hue="clustered_pred", multiple="dodge",
shrink=.8);
```

REFERENCES

Christy, A.J. & Umamakeswari, A. & Priyatharsini, L. & Neyaa, A. (2018). RFM ranking – An effective approach to customer segmentation. *Journal of King Saud University – Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2018.09.004

Myles D. (2020). Les rencontres amoureuses et sexuelles au temps des algorithmes: Une analyse comparative de Grindr et Tinder. In Piazzesi, C., Blais, M., Lavigne, J. & C. Lavoie Mongrain (Eds). *Intimités et sexualités contemporaines: changements sociaux, transformations des pratiques et des représentations, Presses de l'Université de Montréal,* p. 1-11.

Chinedu Pascal Ezenkwu, Simeon Ozuomba (2015) *International Journal of Advanced Research in Artificial Intelligence, Vol. 4,* No.10.

Charles A. Bouman (2005). CLUSTER: An Unsupervised Algorithm for Modeling Gaussian Mixtures

Jialu Liu, Deng Cai, Xiaofei He. Gaussian Mixture Model with Local Consistency

Du, W. & Zhan, Z. (2002). "Building Decision Tree Classifier on Private Data". *Electrical Engineering and Computer Science. 8.*

Karthick Suyambu (2017). Semi Supervised Hierarchy Forest Clustering and KNN Based Metric Learning Technique for Machine Learning System. *Journal of Advanced Research in Dynamical and Control, Vol. 9.*

Mishra P. & Pandey C.M. & Singh U. & Gupta A. & Sahu C. & Keshri A. (2019). Descriptive statistics and normality tests for statistical data. *Ann Card Anaesth.* 67-72. doi: 10.4103/aca.ACA_157_18.

Juan A. Barceló (2018). Chi-Square Analysis. https://doi.org/10.1002/9781119188230.saseas0090

Bholowalia, P. & Kumar, A. (2014). Article: EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications 105* (9):17-24.

Shutaywi, M. & Kachouie, N.N. (2021). Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering. *Entropy.23* (6):759. https://doi.org/10.3390/e23060759