CSE454 Data Mining Project

Emre YILMAZ - 1901042606

***Abstract*—In this project, through various data mining applications, meaningful information has been tried to be obtained by looking at various characteristics of traffic accidents.**

***Keywords*—data mining, clustering, dbscan, classification, decision tree, preprocessing**

I. Problem Definition

The data consists of approximately 250,000 rows and 70 columns related to various traffic accidents that occurred in the UK, obtained from Kaggle. Among these 70 features, there is a wide range of information including accident location, road conditions, weather conditions, driver characteristics, characteristics of those affected by the accident, and characteristics of the vehicles involved in the accident. In this study, although not all of these features may be used, the dataset and this project conducted on it are important for identifying accident characteristics and taking necessary precautions.

In this study, the preprocessing process was initially performed to complete missing information in the dataset. Following that, a classification model was developed, and finally, a clustering model was developed to extract various meaningful insights. Although these processes may appear basic, various data mining techniques were attempted to be used during the implementation.

II. Data Overview

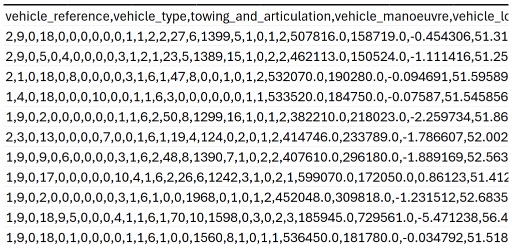
The data is available on Kaggle in the form of a CSV file, consisting of 250,000 rows and 70 columns with various features. Most of the features in the dataset are categorical (nominal) values, but there are also numeric values. It's worth noting that the initial download from Kaggle contains many rows with missing information. The categorical values are always represented in an ordered manner, such as 1, 2, 3, and so on.

III. Preprocessing

As mentioned earlier, two strategies were used to handle missing information in the dataset for this project:

A. Removing Lines

It was observed that some rows had a significant amount of missing information. A threshold was set, and rows with missing information exceeding this threshold were removed from the dataset. This threshold is indicated as 7 at the beginning. If a line includes more than 7 missing values, it is removed.



```
vehicle_reference,vehicle_type,towing_and_articulation,vehicle_manoeuvre,vehicle_l
2,9,0,18,0,0,0,0,0,0,1,1,2,2,27,6,1399,5,1,0,1,2,507816.0,158719.0,-0.454306,51.31
2,9,0,5,0,4,0,0,0,0,3,1,2,1,23,5,1389,15,1,0,2,2,462113.0,150524.0,-1.111416,51.25
2,1,0,18,0,8,0,0,0,0,3,1,6,1,47,8,0,0,1,0,1,2,532070.0,190280.0,-0.094691,51.59589
1,4,0,18,0,0,0,10,0,0,1,1,6,3,0,0,0,0,0,1,1,533520.0,184750.0,-0.07587,51.545856
1,9,0,2,0,0,0,0,0,0,1,1,6,2,50,8,1299,16,1,0,1,2,382210.0,218023.0,-2.259734,51.86
2,3,0,13,0,0,0,0,7,0,0,1,6,1,19,4,124,0,2,0,1,2,414746.0,233789.0,-1.786607,52.002
1,9,0,9,0,6,0,0,0,0,3,1,6,2,48,8,1390,7,1,0,2,2,407610.0,296180.0,-1.889169,52.563
1,9,0,17,0,0,0,0,0,10,4,1,6,2,26,6,1242,3,1,0,2,1,599070.0,172050.0,0.86123,51.412
1,9,0,2,0,0,0,0,0,0,3,1,6,1,0,0,1968,0,1,0,1,2,452048.0,309818.0,-1.231512,52.6835
1,9,0,18,9,5,0,0,0,4,1,1,6,1,70,10,1598,0,3,0,2,3,185945.0,729561.0,-5.471238,56.4
1,9,0,18,0,1,0,0,0,0,1,1,6,1,0,0,1560,8,1,0,1,1,536450.0,181780.0,-0.034792,51.518
```

Şekil 1: Data Overview.

B. Using Correlation Analysis

To fill in some of the missing information, the mean imputation method was used, but instead of taking the mean of the entire dataset, a more sophisticated algorithm was employed. The literature research indicated that correlation analysis can be used to fill in missing values [1]. This information was used at a basic level in the project:

- All features containing missing information were identified.
- For each of these identified features, correlation analysis was performed with all other features.
- The top 5 features that appeared most correlated with each identified feature were determined.
- Then, for each missing row, all rows with the same values for the top 5 correlated columns were found.
- The mode of the missing column was calculated using these rows to fill in the missing information.

IV. Correlation Analysis

A. Implementation

Cramer's V [1] [2] was used for correlation analysis in the preprocessing phase. Cramer's V correlation is a statistical metric that measures the strength of the relationship between two nominal (categorical) variables. This metric is calculated using the results of the chi-square test, but Cramer's V normalizes the chi-square statistic based on the number of variables and sample size. This allows for comparable results across tables of different sizes and data with different sample sizes.

978-1-5386-1501-0/18/$31.00 © 2018 IEEE

The chi-square test assesses whether there is independence between two categories. If there is no independence between two categories, then it can be said that there is a relationship. The chi-square test measures how different the observed frequencies are from the expected frequencies. The result of the test is evaluated based on a chi-square distribution, producing a p-value. This p-value indicates whether the observed difference is due to random variation or not.

Cramer's V, using the chi-square value, measures the strength of the relationship between two nominal variables. The value ranges from 0 to 1. A value of 0 indicates no relationship, while a value of 1 indicates a perfect relationship. This metric provides a more standardized measure for tables of different sizes because it adjusts the chi-square value based on the sample size and the number of categories.

The calculation of Cramer's V involves the following steps:

1) Calculate the Chi-square statistic (χ^2) using the formula:

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} is the observed frequency and E_{ij} is the expected frequency for each cell in the contingency table.

2) Determine the sample size (n), which is the total number of observations in all cells of the table. Also, find the number of rows (r) and the number of columns (k) of the table.

3) (Optional - Bias Correction) Adjust the Chi-square value to correct for bias in small or unbalanced samples:

$$\phi^2_{corr} = \max(0, \chi^2/n - ((k-1)(r-1))/(n-1))$$

Additionally, adjust the counts of rows and columns:

$$r_{corr} = r - \frac{(r-1)^2}{n-1}, \quad k_{corr} = k - \frac{(k-1)^2}{n-1}$$

4) Calculate Cramer's V:

- Without Bias Correction:

$$V = \sqrt{\frac{\chi^2}{n \times \min(k-1, r-1)}}$$

- With Bias Correction (if step 3 was applied):

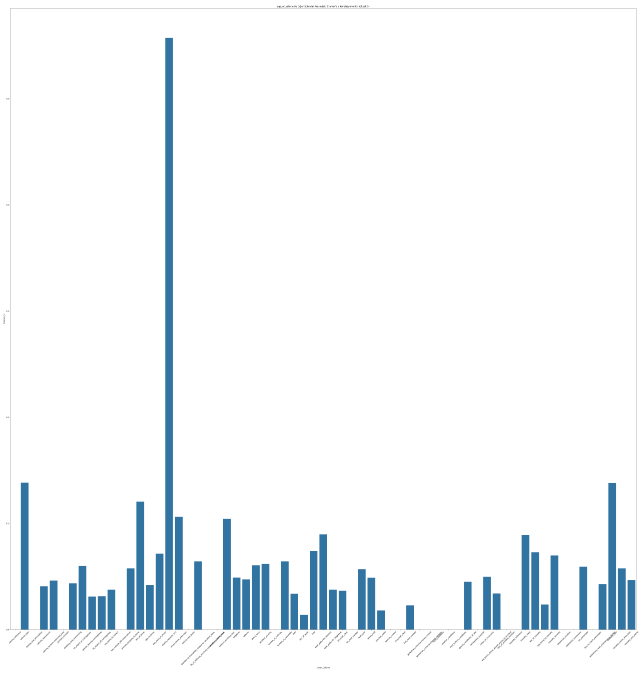
$$V = \sqrt{\frac{\phi^2_{corr}}{\min(k_{corr}-1, r_{corr}-1)}}$$

You can access the implementation from "Additions" of this paper

B. Results

As I mentioned before, I calculated the correlation tables of each column with missing information against all other columns. I will now highlight some noteworthy findings. You can access the complete results in the "Additions" section.

As you can see, age of vehicle is highly correlated with engine capacity. We can prove that correlation method is working by looking graphs. You can access all correlation graphs in source code folder at the end of the paper.



Şekil 2: Result of Age of Vehicle Correlation.

V. Classification

In the Classification step, a decision tree was implemented. I will explain the basic steps of this decision tree step by step. The splitting algorithm was configured to work with both the GINI Index and Entropy, and the results were compared. Here's a detailed analysis:

A. Gini Index and Entropy

Gini Index: The Gini Index measures the purity (homogeneity) of a dataset at each node of a decision tree. Its value ranges from 0 to 1, where 0 indicates complete purity (homogeneity), and 1 indicates complete heterogeneity. The Gini Index is calculated as:

$$\text{Gini Index} = 1 - \sum (p_i)^2$$

where p_i is the proportion of the samples that belong to class i at a specific node.

Entropy: Entropy is a measure of the disorder or uncertainty in a dataset. A lower entropy value indicates a more orderly dataset (less uncertainty), while a higher value indicates more disorder. Entropy is calculated as:

$$\text{Entropy} = -\sum p_i \log_2(p_i)$$

where p_i is the proportion of the samples that belong to class i.

Information Gain: Information Gain is the amount of "purity increase" achieved by splitting a dataset based on a feature. A feature with high information gain is a good candidate for a node in the decision tree.

B. Weight Mechanism

Weight Mechanism: To better handle imbalanced datasets where some classes are underrepresented, a weight mechanism was introduced in the decision tree. This mechanism assigns higher weights to rarer classes, thereby increasing the sensitivity of the decision tree to these classes. The weights are incorporated into the calculations of Gini Index and Entropy, effectively biasing the tree towards splits that correctly classify the rarer classes. This approach is crucial for datasets where certain class outcomes are more critical despite their lower frequency.

calculate_weighted_metric: This function's approach to calculate Gini Index or Entropy, which factors in the class imbalance by weighting the classes differently. This approach is particularly useful for ensuring that the decision tree does not become biased towards the majority class in imbalanced datasets.

C. Implementation

check_purity Function: Checks if the data split is pure (only one class remains).

classify_data Function: Classifies data by majority vote.

get_potential_splits Function: Determines potential splitting points for the data.

split_data Function: Splits the data into two parts based on a threshold value.

calculate_entropy and calculate_gini Functions: Calculates the entropy or Gini Index of a particular split.

calculate_overall Function: Calculates the weighted metric (Gini/Entropy) of two splits.

determine_best_split Function: Determines the best split using the specified metric (Gini/Entropy).

DecisionNode Class: Represents each node of the decision tree.

decision_tree_algorithm Function: Implements the fundamental steps of the decision tree algorithm. It first checks if the dataset is pure, then determines the best split, and recursively constructs each subtree of the tree.

Decision Tree Algorithm This code constructs a decision tree by splitting the dataset in a way that maximizes the purity of the dataset. At each step, the best split for the current dataset (the feature and threshold value yielding the highest information gain) is found using the Gini Index or Entropy. The dataset is then split into two subsets based on this threshold value, and the process is recursively repeated for each subset. This continues until the dataset becomes pure or the maximum depth is reached.

Decision trees can model complex structures in datasets with simple decision rules, hence are widely used in classification and regression problems.

D. Test

This section outlines the selection of specific attributes for the purpose of predicting "accident severity" using a decision tree model. In the context of traffic safety analysis, certain attributes can be particularly indicative of the severity of an accident.

Selected Attributes for Predicting Accident Severity:

The following 13 attributes have been selected based on their relevance and potential predictive power for "accident severity":

1) **vehicle_type:** The type of vehicle involved could significantly impact both accident and casualty severity.
2) **age_of_driver:** Age can be a crucial factor in driving behavior and accident outcomes.
3) **sex_of_driver:** Gender may have different risk profiles in accidents.
4) **road_type:** Different road types can have varying risks associated with accidents.
5) **speed_limit:** Higher speed limits might be correlated with more severe accidents.
6) **weather_conditions:** Adverse weather can contribute to the severity of accidents.
7) **light_conditions:** Poor lighting could lead to more severe accidents.
8) **road_surface_conditions:** Slippery or poor road conditions can increase accident severity.
9) **junction_detail:** The characteristics of junctions can influence the likelihood and severity of accidents.
10) **vehicle_manoeuvre:** The maneuver the vehicle was performing before the accident.
11) **day_of_week:** The day of the week might correlate with different traffic conditions and accident severities.
12) **time:** The time of day could be related to visibility conditions, traffic density, and the likelihood of impaired driving.
13) **urban_or_rural_area:** Accidents in urban or rural areas may differ in severity due to varying road conditions, traffic, and response times of emergency services.

These attributes are chosen based on common factors that are often considered in traffic safety analyses. They encompass a range of factors including driver demographics, vehicle characteristics, environmental conditions, and road infrastructure. The use of these attributes in a decision tree model aims to provide insights into the factors that significantly influence accident severity, thereby aiding in effective traffic safety management.

Firstly, it's important to emphasize that the data is distributed in an unbalanced way: Label 3 (Slight): 8504 Label 2 (Serious) 1359 Label 1 (Fatal) 123

Therefore, a weighting mechanism has been used in the algorithm. Weighting mechanisms increase the weight of examples in the minority class, making these examples perceived as "more important" by the model. This approach helps the model better recognize and focus on examples in the minority class. [4], [5]

The weighting mechanism used in this implementation has been enhanced at the algorithm level.

Additionally, the decision tree was designed to work with both the GINI Index and Entropy splitting approaches.

E. Test - 1

Let’s see the results for each case with 0.05 of data:

GINI Index, max_depth = 3: Accuracy = 0.85

GINI Index, max_depth = 6: Accuracy = 0.84

GINI Index, max_depth = 9: Accuracy = 0.84

Entropy, max_depth = 3: Accuracy = 0.84

Entropy, max_depth = 6: Accuracy = 0.85

Entropy, max_depth = 9: Accuracy = 0.84

F. Test - 2

Let’s see the results for each case with 0.1 of data:

Entropy, max_depth = 7: Accuracy = 0.85

G. Results

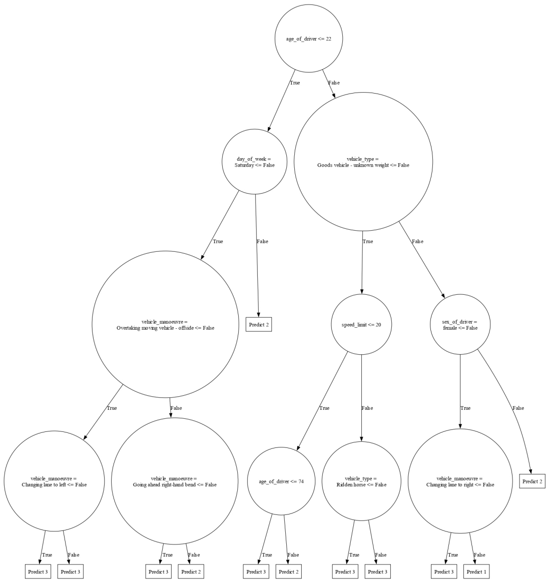
Maximum Depth and Overfitting: Generally, as the maximum depth increases, overfitting of the model to the training dataset is expected. However, in this case, as the maximum depth increases, a significant decrease in accuracy rates has not been observed. This may indicate that the dataset is sufficiently represented or that increasing the depth of the tree does not unnecessarily increase the model’s complexity.

Comparison of GINI Index and Entropy: There is no significant difference in performance between the GINI Index and Entropy. Both metrics have provided similar accuracy rates. This indicates that the dataset is not influenced significantly by these two different purity measures.

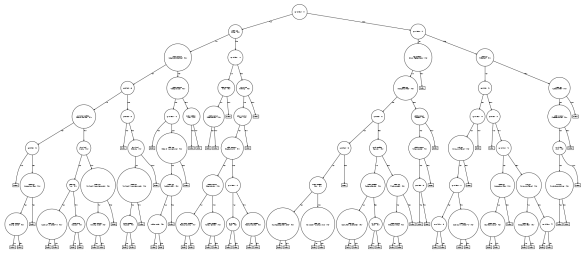
The decision tree was constructed using small samples due to limitations in computer performance when dealing with large samples. This choice might result from the data not being complex enough when working with smaller samples. Therefore, the excellent results obtained by the decision tree can be interpreted as expected. It’s essential to test the model on the entire dataset, but it’s clear that it has performed well in its current state.

There is a tree example with max_depth=4, and there is a tree example with max_depth = 8. By looking this trees, several conclusions can be drawn from the accident data set at hand:

- **Driver’s Age:** According to the decision tree, the driver’s age is an important feature. It suggests that the age of the driver plays a significant role in determining the outcome of accidents.
- **Day of Week:** The day of the week, is highlighted as an important feature.
- **Maneuver Type:** The way the vehicle maneuvers is identified as a critical determinant of accident severity. In particular, the type of lane change appears to have an impact on the accident’s severity.
- **Sex of Driver:** When looking at a deeper decision tree (max_depth=8) that achieves an accuracy of 0.83 based on our data, it is interesting to observe that the driver’s gender appears to have an impact on the severity of accidents. This is actually a result contrary to the general belief because we would typically assume that women, being more cautious and driving at slower speeds, would be involved in less severe accidents.



Şekil 3: An Example of Tree.



Şekil 4: An Example of Tree.

- **Weather Conditions:** Weather conditions do impact the severity of accidents, especially rain and snow-wind, which are seen to affect the severity of accidents.

These findings provide valuable insights into the factors that contribute to accident outcomes based on the dataset, highlighting the importance of driver age, the day of the week, and maneuver type in predicting accident severity. I would like to emphasize once again that to obtain the most accurate results, it is recommended to build the tree with the entire dataset. Unfortunately, my resources were not sufficient for this.

H. Comparing with Logistic Regression

I test same train and test set using Logistic Regression model. I use sklearn library to do this.

Accuracy: 0.89

We can clearly see that the linear regression model yields better results. Here are the possible reasons:

- **Lower Risk of Overfitting:** Decision trees, especially without pruning or depth control, can overfit the training data. Logistic regression, generally being a simpler model, might have a lower risk of overfitting.
- **Scalability and Robustness:** Logistic regression works well with large datasets and various feature types and is relatively robust against outliers.
- **Small Datasets:** Logistic regression might perform better with datasets that have many features but few observations, as decision trees tend to overfit in these scenarios.

It has been observed that classification techniques yield high success in most conditions. The reason for this is discussed under the 'Discussion' section at the end of the paper.

VI. Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a popular clustering method that is particularly adept at identifying clusters of arbitrary shape in datasets and handling noise. It groups together points that are closely packed together and marks as outliers points that lie alone in low-density regions.

Let's see the implementation first:

A. Euclidean Distance Function

- **Purpose:** Calculates the Euclidean distance between two points, which is the straight-line distance in a multi-dimensional space.
- **Usage in DBSCAN:** DBSCAN uses the Euclidean distance to determine the closeness of points in a dataset, which is crucial for defining the ε-neighborhood of a point.

B. Radius Neighbors Function

- **Purpose:** Identifies all points within a radius ε of each point in the dataset.
- **Usage in DBSCAN:** Essential for finding the neighbors of each point within a given radius, which is a key step in forming clusters.

C. DBSCAN Function

- **Purpose:** Implements the core DBSCAN clustering algorithm.
- **Parameters:**
  - X: Dataset containing points.
  - *eps*: Maximum distance between two samples for one to be considered as in the neighborhood of the other.
  - $min_samples$: The number of samples in a neighborhood for a point to be considered a core point.
- **Process:**
  - $CalculateNeighbors$: First, it calculates the neighbors for each point in X using the radius_neighbors function.
  - $ClusterFormation$: The algorithm then iterates through each point and forms clusters based on the density (i.e., if the number of neighbors within the distance eps is greater than or equal to min_samples).
  - $NoiseIdentification$ :: Points that do not meet these criteria are labeled as noise (-1).
  - $ExpansionofClusters$: For each core point, it recursively adds all directly density-reachable points to the cluster.
  - $Output$: The function returns an array labels where the index represents the original data point in X and the value at each index is the cluster label the point belongs to. Noise points are labeled as -1.

In summary, this implementation of DBSCAN works by identifying core points (based on eps and min_samples), forming clusters around these core points, and labeling non-core points as noise or part of a cluster. The algorithm is effective for datasets where clusters have varying shapes and densities, and where noise may exist. It's a widely used clustering algorithm in applications such as anomaly detection, spatial data analysis, and image segmentation.
You can access the implementation from "Additions" of this paper

VII. Test Process

Firstly, we converted categorical values into the correct format for clustering using one-hot encoding.

A. General Process

The first step involves preparing the longitude and latitude data for clustering. This step is crucial for ensuring the quality and accuracy of the clustering results.

B. Elbow Method for EPS Parameter Estimation

- **Function:** `k_nearest_neighbor`
- **Purpose:** To compute the distances of the k-nearest neighbors for each point in the dataset.
- **Usage:** This function aids in finding the optimal eps value for DBSCAN. The method involves plotting the distance to the k-th nearest neighbor and identifying the 'elbow' point where the rate of increase changes sharply.

C. Applying DBSCAN Clustering

- **Procedure:** Run the DBSCAN algorithm on the coordinates with `eps` set to 0.5 and `min_samples` to 40.
- **Output:** An array `labels` containing the cluster labels for each point.
- **Objective:** To perform spatial clustering based on the density of points, where each point is either assigned to a cluster or marked as noise (-1).

D. Counting Samples in Each Cluster

- **Function:** Count and print the number of samples in each cluster.
- **Purpose:** To understand the distribution of data points across the clusters formed by DBSCAN.

E. Visualization of Clusters

The final step involves visualizing the spatial clusters formed. This visualization helps in interpreting the spatial patterns and understanding the geographical distribution of the data points.

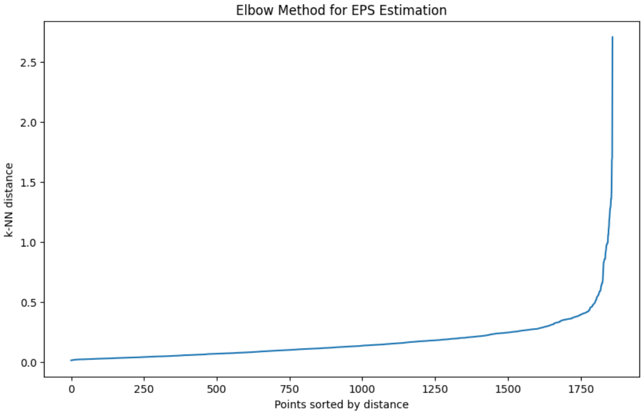
VIII. Results

A. Test - 1 Clustering

Clustering of data points based on their geographical locations (longitude and latitude). It involves the steps of determining an appropriate eps value for DBSCAN using the Elbow Method, applying the DBSCAN algorithm to identify clusters, and visualizing these clusters. The code is particularly useful for understanding spatial patterns and distributions in a dataset.

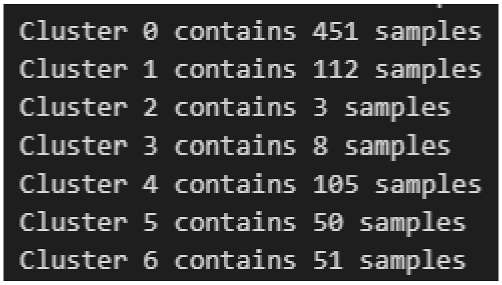
It is tested by using 0.005 of data.

The result of this test will show us where accidents are predominantly clustered. Here are the results:



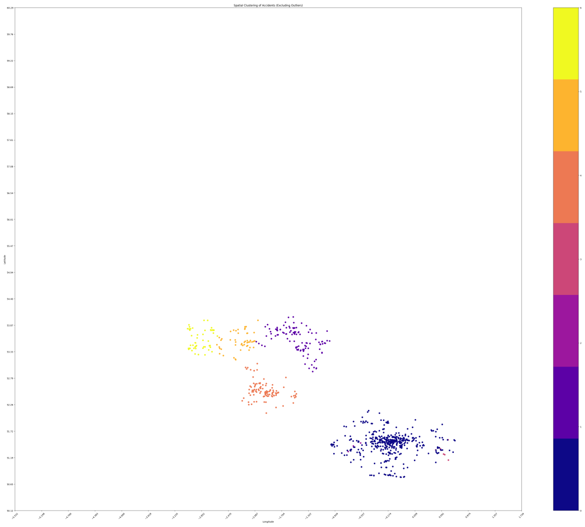
Şekil 5: Result of Elbow Method.

As you see from Elbow graph, the appropriate EPS parameter is 0.4, so we are going to use it.

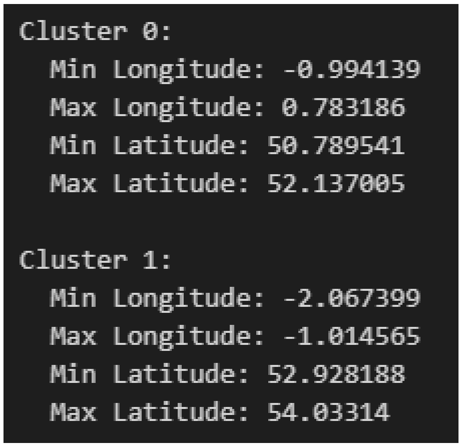


```
Cluster 0 contains 451 samples
Cluster 1 contains 112 samples
Cluster 2 contains 3 samples
Cluster 3 contains 8 samples
Cluster 4 contains 105 samples
Cluster 5 contains 50 samples
Cluster 6 contains 51 samples
```

Şekil 6: Points of Clustering.



Şekil 7: Result of Clustering.



```
Cluster 0:
  Min Longitude: -0.994139
  Max Longitude: 0.783186
  Min Latitude: 50.789541
  Max Latitude: 52.137005

Cluster 1:
  Min Longitude: -2.067399
  Max Longitude: -1.014565
  Min Latitude: 52.928188
  Max Latitude: 54.03314
```

Şekil 8: Stats of Greatest Clusters.

As seen from the results, the clusters appear to be spherical. We could have obtained a similar result using a different clustering algorithm. However, due to the high number of outliers in our dataset, I found it appropriate to use DBScan. DBScan is quite successful in identifying outliers [6]. And we can see from the statistics where accidents are concentrated.

IX. Test - 2 Clustering

In the second test, I attempted clustering using multiple categorical features. Clustering using 'vehicle_manoeuvre', 'pedestrian_movement', 'vehicle_location-restricted_lane', 'junction_location' allows for a detailed analysis of the interaction between driver behavior, pedestrian movement, and road infrastructure in causing accidents. It can provide actionable insights for improving road safety and reducing accident rates.

Finding the correct EPS value with the elbow method is not a reasonable solution for categorical values. Therefore, I looked into the literature and thought that applying the Silhouette Score method could be successful [7].

A. Silhouette Score

Silhouette score is a metric used to evaluate the efficiency of a clustering algorithm. This score measures how well each data point fits within its own cluster and how well it is separated from other clusters. The Silhouette score ranges from -1 to 1, where high values indicate that data points fit well within their clusters and are well-separated from other clusters.

The calculation of the Silhouette score is done as follows:

Cohesion: For each data point, the average distance to other points within the same cluster is calculated. This value represents the data point's membership within its own cluster.

Separation: For each data point, the average distance to the nearest neighboring cluster is calculated. This shows how far it is from points in a different cluster.

Silhouette Score: For each data point, the score is calculated using the formula (separation - cohesion) / max(cohesion, separation). This score ranges between -1 and 1. Values close to 1 indicate good clustering, values close to 0 indicate ambiguity between clusters, and values close to -1 indicate misclassification

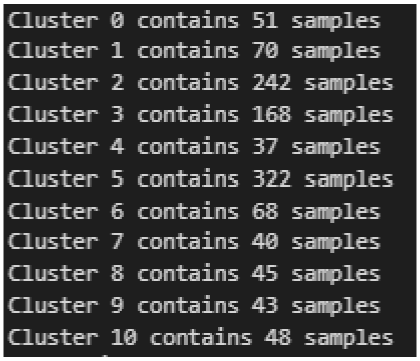
Some tests are conducted to find the most appropriate EPS and min_pts parameters with different parameters:

The most successful EPS parameter with min_pts = 25: Eps = 0.1, Score = 0.49

The most successful EPS parameter with min_pts = 15: Eps = 0.1, Score = 0.58

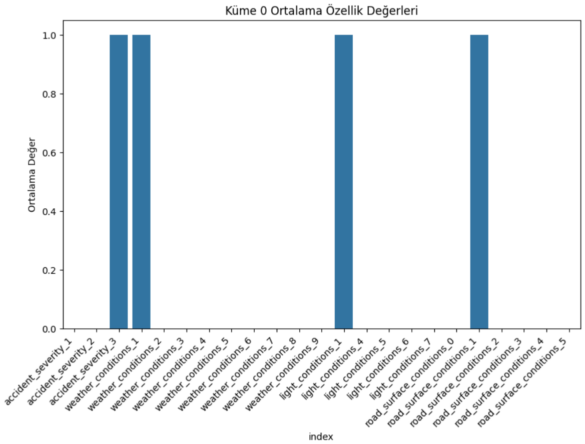
The most successful EPS parameter with min_pts = 35: Eps = 0.1, Score = 0.48

Here are the results for the most appropriate clusters:



```
Cluster 0 contains 51 samples
Cluster 1 contains 70 samples
Cluster 2 contains 242 samples
Cluster 3 contains 168 samples
Cluster 4 contains 37 samples
Cluster 5 contains 322 samples
Cluster 6 contains 68 samples
Cluster 7 contains 40 samples
Cluster 8 contains 45 samples
Cluster 9 contains 43 samples
Cluster 10 contains 48 samples
```

Şekil 9: Stats of Greatest Clusters.



Şekil 10: Stats of Greatest Clusters.

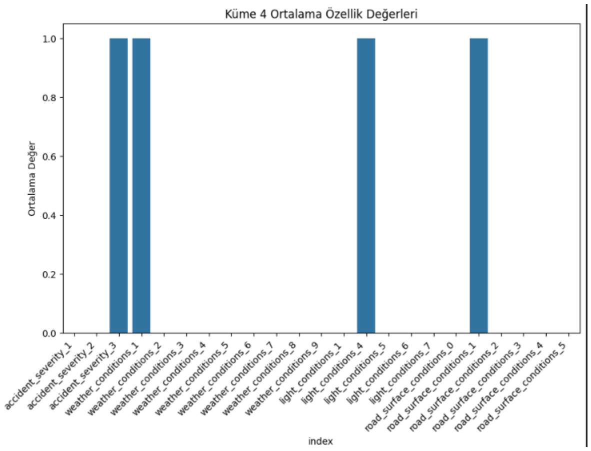
B. Cluster Information of the Greatest 2 Clusters

vehicle_manouver_18 = Going Ahead Other

pedestrian_movement_0 = No Pedastrian

vehicle_location_restricted_lane_0 = Not In Restricted Lane

junction_location_8 = Mid Junction - on roundabout or on main road



Şekil 11: Stats of Greatest Clusters.

vehicle_manouver_18 = Going Ahead Other

pedestrian_movement_0 = No Pedastrian

vehicle_location_restricted_lane_0 = Not In Restricted Lane

junction_location_8 = Not at or within 20 metres of junction

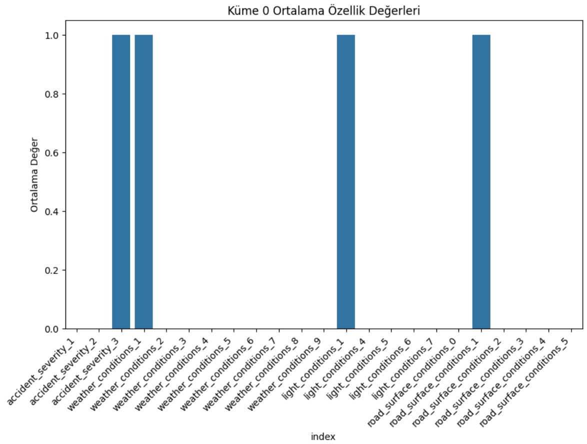
C. Conclusion of Test - 2 Clustering

Looking at the results, we can interpret how the accidents are clustered. Accidents generally occur in the form of head-on collisions, do not involve pedestrians, and take place in an unrestricted lane. They occur at a roundabout or on a straight road

X. Test - 3 Clustering

In this test, I applied clustering using multiple categorical features such as accident severity, weather conditions, light conditions, and road surface conditions. Clustering these features allows us to analyze in detail how various factors that influence the causes and severity of traffic accidents interact with each other.

In particular, examining the effects of weather and light conditions on accidents provides important information about the environmental conditions in which the accidents occur. Understanding the role of road surface conditions on the severity of accidents can guide the development of road safety measures. This analysis could provide actionable insights for reducing traffic accidents and enhancing road safety.



Şekil 12: Stats of Greatest Clusters.

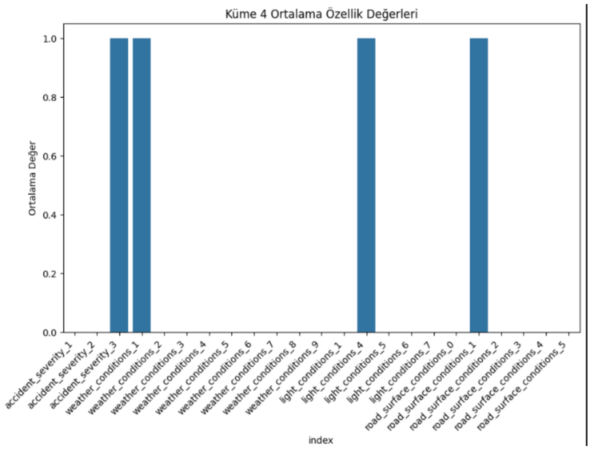
A. *Cluster Information of the Greatest 2 Clusters*

accident_severity_3 = Slight Accident

weather_conditions_1 = Fine No High Winds

light_conditions_4 = Darkness - Light Lit

road_surface_conditions_1 = Dry



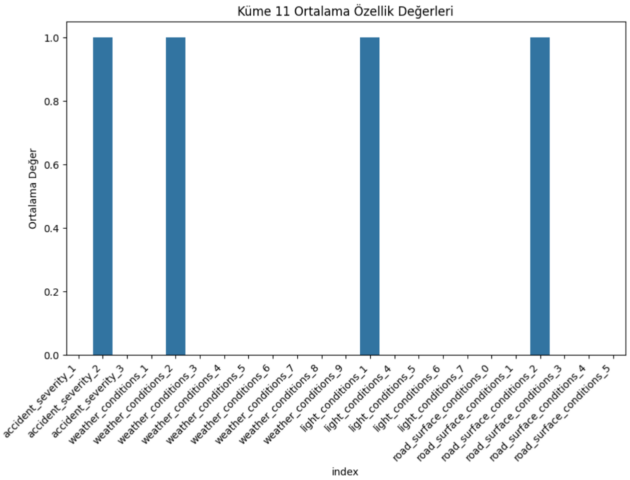
Şekil 13: Stats of Greatest Clusters.

accident_severity_3 = Slight Accident

weather_conditions_1 = Fine No High Winds

light_conditions_1 = Daylight

road_surface_conditions_1 = Dry



Şekil 14: Stats of The Smallest Cluster.

accident_severity_3 = Serious Accident

weather_conditions_1 = Raining No Wind

light_conditions_1 = Daylight

road_surface_conditions_1 = Wet

B. *Conclusion of Test - 3 Clustering*

Looking at the results, it can be observed that accidents with lower severity cluster in places with general and seemingly problem-free conditions such as dry road surface, daylight, and straight roads. This may indicate that drivers tend to be more cautious during rainy and foggy weather conditions.

XI. Discussion

It has been particularly noted that classification techniques consistently yield high success. The reason for this can be understood from the results of the clustering analysis. Accidents generally occur under similar conditions. For example, we might predict that accidents are likely to happen in rainy and foggy weather, but in practice, it's observed that they mostly occur under optimal conditions. Therefore, for the analyses to give the most accurate results, training and analysis should be done with the entire dataset. This is because when working with small samples, the tendency of the dataset is usually towards the same meaning.

Kaynaklar

[1] Estimating missing data using novel correlation maximization based methods, Amir Masoud Sefidian, Negin Daneshpour.

[2] The Measurement of Association in Industrial Geography , L. Cramer, 1946

[3] Statistical Power Analysis for the Behavioral Sciences, J. Cohen, 1988)

[4] He, H., Garcia, E. A. (2009). "Learning from Imbalanced Data." IEEE Transactions on Knowledge and Data Engineering.

[5] Barandela, R., Sánchez, J. S., García, V., Rangel, E. (2003). "Strategies for learning in class imbalance problems."

[6] Ester, M., Kriegel, H. P., Sander, J., Xu, X. (1996). "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96).

[7] Kaufman, L. ve Rousseeuw, P.J. (1990), "Finding Groups in Data: An Introduction to Cluster Analysis", Wiley-Interscience.

XII. Video

https://youtu.be/fPun_EE3AR4

XIII. Source Code and All Related Files

https://drive.google.com/drive/folders/1RENBYGp-se8SmdhOsQ-edwRpS2kynpu2?usp=sharing