Chapter – 4
Feature Extraction and Data Processing Using IoT

4.1 Overview

Dataset contains hundreds of attributes. However, not all attributes are required to complete the machine learning task. A feature extraction and selection algorithms are used to determine the importance of the attributes. Rather than processing all attributes, only relevant attributes are included in the machine learning process. This reduces processing time and also improves the performance of the task. Therefore, attribute extraction and selection algorithms are applied before applying data learning tasks such as classification, clustering, and outlier analysis.

Feature extraction is used to extract features from the collected data. Feature selection is the extension of feature extraction. Feature selection is the process of selecting a subset of relevant features (variables, predictors) to use in model building [Chong and Ng 2016]. In normal situations, domain knowledge plays a key role and allows you to select the features that seem most important. For example, when predicting sales of Furniture, the type of furniture, size of furniture and budget of customer may be important. Feature selection follows Feature extraction which simply selects required features and remove unwanted or redundant features from the Data set[Coutard 2014]. Feature selection performs following functions.

1. Remove Features with missing values
2. Remove highly uncorrelated features
3. Remove Features with low variance

Feature extraction and data processing using IoT involves extracting relevant information or features from the data collected by IoT devices and processing them for further analysis or application. Feature extraction is an important step in machine learning, and its goal is to reduce the dimensionality of input data while preserving important information. Extracting relevant features is important to improve the efficiency of machine learning algorithms, reduce computational complexity, and improve model performance. The step-by-step overview of the process is given below:

Data Collection: IoT devices collect data from various sensors, such as temperature, humidity, motion, or light sensors. The data can be collected continuously or at regular intervals and transmitted to a central server or cloud platform for processing.

Preprocessing: Raw data collected from IoT devices often requires preprocessing to remove noise, handle missing values, or normalize the data. This step ensures that the data is in a suitable format for further analysis.

Feature Extraction: Feature extraction involves identifying and extracting relevant features from the preprocessed data. Features are specific measurements or characteristics that capture the essential information for the intended analysis or application. For example, in a smart home scenario, features could include temperature, occupancy status, or energy consumption patterns.

Selection and Dimensionality Reduction: Depending on the application, it may be necessary to select a subset of features or reduce the dimensionality of the data. This step aims to eliminate irrelevant or redundant features, improving computational efficiency and reducing the risk of overfitting in machine learning models.

Data Integration: In some cases, data from multiple IoT devices or sources may need to be integrated to derive meaningful insights. Integration can involve combining data from various sensors, time synchronization, or merging data from different locations or devices.

Data Analytics: Once the relevant features have been extracted and processed, various analytics techniques can be applied to gain insights or make predictions. This can include statistical analysis, data mining, machine learning algorithms, or artificial intelligence models.

Visualization and Reporting: The processed data and analytics results can be visualized using charts, graphs, or dashboards to provide a clear representation of the information. Visualizations aid in understanding patterns, trends, or anomalies in the data. Additionally, reports or alerts can be generated to notify users or stakeholders of important findings or events.

Real-Time Processing: IoT systems often require real-time processing to enable timely decision-making or immediate actions based on the collected data. Real-time processing involves analyzing data as it arrives and generating responses or triggers in near real-time.

Feedback Loop: The insights or actions derived from the processed data can be used to provide feedback and optimize the IoT system's performance. For example, adjusting sensor thresholds, improving predictive models, or triggering automated responses based on the analysis results.

Overall, feature extraction and data processing using IoT play a crucial role in transforming raw data collected from IoT devices into meaningful information and actionable insights for various applications such as smart homes, industrial monitoring, healthcare, or environmental monitoring.

Algorithm: Combined Feature Selection, Evaluation, and Attribute Selection

1. **Load Data:**
    - Load the dataset from a given file path.
2. **Evaluate Feature Selection Methods:**
    - For each chosen feature selection method (like CfsSubset Eval, GainRatio Attribute Eval, etc.):
    - Apply the method to the dataset to find the most important attributes.
    - Store the list of selected attributes for each method.
3. **Combine Selected Attributes**:
    - Create an empty list to hold combined selected attributes.
    - For each list of selected attributes from different methods:
        - Add the attributes to the combined list, avoiding duplicates.
4. **Apply Feature Selection on Combined Data**:
    - Use the CfsSubsetEval method on the dataset with the combined selected attributes.
    - Create a new dataset with the chosen attributes.

5. **Train Classifier**:
    - Choose a classifier, like Linear Regression.
    - Train the classifier using the data obtained after combined feature selection.

6. **Rank and Percentile Calculation**:
    - Calculate the attribute performance scores based on the classifier's coefficients.
    - Rank the attributes based on their performance scores.
    - Calculate the percentile of each attribute's performance score.

7. **Select Most Appropriate Attributes**:
    - Choose a threshold percentile for attribute selection (e.g., 80%).
    - Select attributes that have performance percentiles above the threshold.

8. **Display Results**:
    - Display the matrix of selected attributes from each method.
    - Display the list of combined attributes.
    - Display the coefficients of the trained classifier.
    - Display the ranked attributes along with their performance scores and percentiles.
    - Display the final selected attributes based on the threshold.
9. **End**.

Machine learning algorithms are the core components of machine learning systems. They allows computers to learn from data and make predictions and decisions without being explicitly programmed for the task. Overall, the model offers an advanced approach to attribute selection, enabling the creation of more accurate and interpretable classification models across various domains and applications. The various Algorithms for extracting useful information are discussed in the following paragraphs.

4.2 Feature Selection Methods

Feature selection methods are essential techniques in data analysis and machine learning that streamline datasets by identifying the most pertinent attributes. Among these methods, CfsSubsetEval evaluates attribute relevance while considering inter-

feature relationships, Gain Ratio AttributeEval assesses attributes' class discrimination ability, One R creates simple rules to predict classes, Relief F handles noisy data by evaluating attribute influence on nearest neighbors' classification, and Symmetrical Uncertainty calculates mutual information between attributes and classes while accounting for class imbalance. Employing these methods enhances data quality, improves model performance, and ensures more effective feature selection, leading to more accurate and interpretable results. When choosing a feature selection method, it is important to consider the type of data, the characteristics of the problem, and the specific requirements of the machine learning task. It is often useful to try multiple techniques and compare their effects on model performance to determine the most effective approach for a particular scenario. The goal is to improve model performance, reduce computational complexity, and reduce the risk of overfitting. 21 attributes used for Analysis are shown in the Table 4.1.

Table 4.1: Attribute Names and ID

Attribute ID	**Attribute Name**
1	TIME
2	REGION_ID
3	SPEED
4	REGION
5	BUS_COUNT
6	NUM_READS
7	HOUR
8	DAY_OF_WEEK
9	MONTH
10	DESCRIPTION
11	RECORD_ID
12	WEST
13	EAST
14	SOUTH
15	NORTH
16	NW_LOCATION
17	SE_LOCATION
18	COMMUNITY AREAS
19	ZIP CODES
20	WARDS
21	CLASS LABEL

4.2.1 Info Gain Attribute Eval

Information gain is a metric used in the context of decision tree-based algorithms, specifically for feature selection. This helps determine the relevance of features in classifying or predicting the target variable. Attribute evaluation using information gains is commonly used in decision tree algorithms. Information gain measures the effectiveness of an attribute in classifying a dataset. It is based on the concept of entropy, which quantifies the uncertainty or disorder in a dataset. The information gain of an attribute is calculated by comparing the entropy of the dataset before and after partitioning based on that attribute.

In Weka, the 'InfoGainAttributeEval' feature selection method is utilized to assess attribute importance within a dataset. This technique quantifies the value of attributes by measuring the reduction in uncertainty they bring to the classification or regression task, particularly in decision tree algorithms. By comparing the entropy before and after splitting the data based on an attribute, the 'InfoGainAttributeEval' method determines the most informative attributes, aiding in improving model performance by focusing on key features. This approach is an essential tool in Weka's arsenal for enhancing data preprocessing and model building processes.

Evaluator: weka.attributeSelection.InfoGainAttributeEval

Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

Search Method: Attribute ranking.

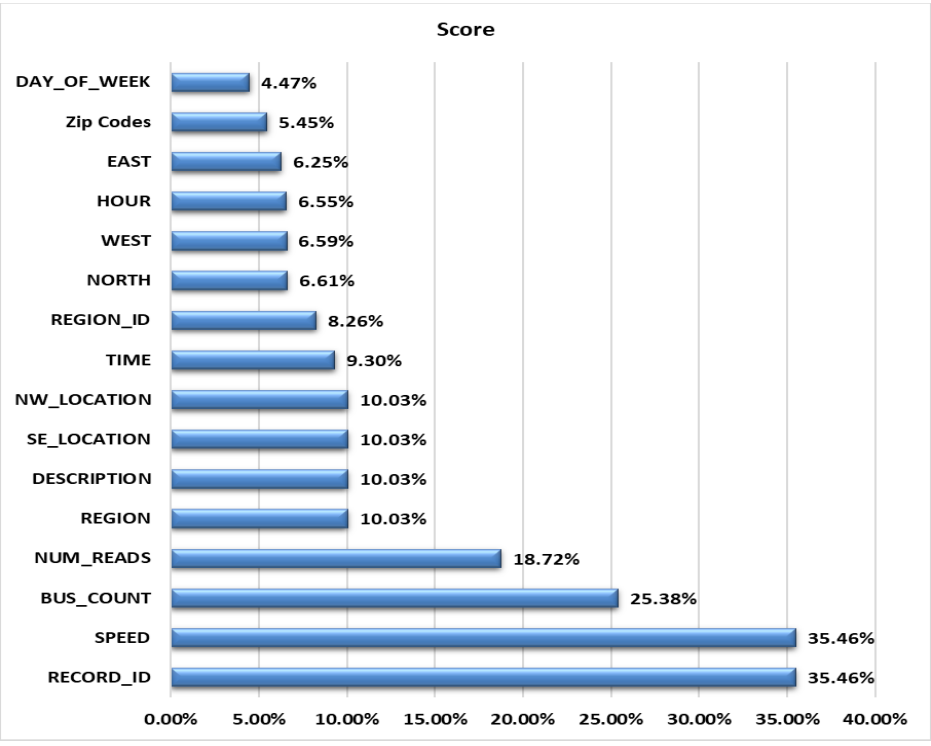
Attribute Evaluator (supervised, Class (nominal)): Information Gain Ranking Filter.

Attribute Scores: Shown in the Table 4.2

Table 4.2 : Attribute Score for Info Gain Attribute Eval

Score	Attribute/ Feature	Attribute ID
0.3546	RECORD_ID	11
0.3546	SPEED	3
0.2538	BUS_COUNT	5
0.1872	NUM_READS	6
0.1003	REGION	4
0.1003	DESCRIPTION	10
0.1003	SE_LOCATION	17
0.1003	NW_LOCATION	16
0.093	TIME	1
0.0826	REGION_ID	2
0.0661	NORTH	15
0.0659	WEST	12
0.0655	HOUR	7
0.0625	EAST	13
0.0545	ZIP CODES	19
0.0447	DAY_OF_WEEK	8

Figure 4.1: Attribute Score for Info Gain Attribute Eval



The results confirm that the top five high scoring attributes are RECORD_ID, SPEED, BUS_COUNT, NUM_READS and REGION with values 0.3546, 0.3546, 0.2538, 0.1872 and 0.1003 respectively. The low scoring attributes are TIME, REGION_ID, NORTH, WEST, HOUR, EAST, ZIP CODES and DAY_OF_WEEK with values 0.093, 0.0826, 0.0661, 0.0659, 0.0655, 0.0625, 0.0545 and 0.0447 respectively. Based on the figure above the 16 selected attributes are as shown below:

Selected attributes: 11,3,5,6,4,10,17,16,1,2,15,12,7,13,19 and 8.

Total No. of selected attributes: 16

4.2.2 Correlation Attribute Eval

Correlation Attribute Eval method serves as a feature selection technique aimed at evaluating attribute significance within a dataset by gauging their correlation with the class variable. By calculating the correlation between each attribute and the class labels, this approach helps identify attributes that bear the most relevance to the classification task. This process aids in refining model performance by retaining attributes that demonstrate strong connections to the class variable and discarding those with weaker correlations.

Evaluator: weka.attributeSelection.CorrelationAttributeEval

Search: weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N 16

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

Search Method: Attribute ranking.

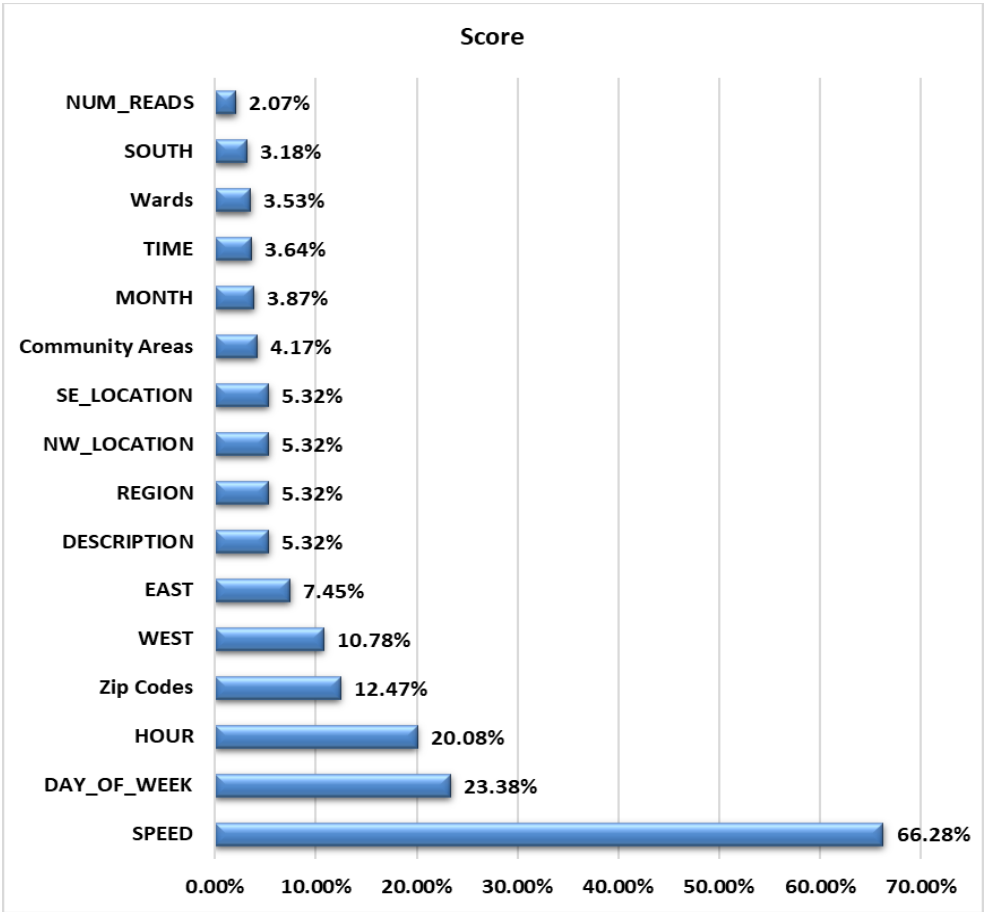
Attribute Evaluator (supervised, Class (nominal)): Correlation Ranking Filter.

Attribute Scores: Shown in Table 4.3

Table 4.3 : Attribute Score for Correlation Attribute Eval

Score	Attribute/ Feature	Attribute ID
0.6628	SPEED	3
0.2338	DAY_OF_WEEK	8
0.2008	HOUR	7
0.1247	ZIP CODES	19
0.1078	WEST	12
0.0745	EAST	13
0.0532	DESCRIPTION	10
0.0532	REGION	4
0.0532	NW_LOCATION	16
0.0532	SE_LOCATION	17
0.0417	COMMUNITY AREAS	18
0.0387	MONTH	9
0.0364	TIME	1
0.0353	WARDS	20
0.0318	SOUTH	14
0.0207	NUM_READS	6

Figure 4.2: Attribute Score for Correlation Attribute Eval



The outcome confirm that the top five high scoring attributes are SPEED, DAY_OF_WEEK, HOUR, ZIP CODES and WEST with values 0.6628, 0.2338,0.2008, 0.1247, 0.1078 respectively. The low scoring attributes are EAST, DESCRIPTION, REGION, NW_LOCATION, SE_LOCATION, COMMUNITY AREAS, MONTH, TIME, WARDS, SOUTH and NUM_READS with values 0.0745, 0.0532, 0.0532, 0.0532, 0.0532, 0.0417, 0.0387, 0.0364, 0.0353, 0.0318 and 0.0207 respectively. Based on the figure above the 16 selected attributes are as shown below:

Selected attributes: 3,8,7,19,12,13,10,4,16,17,18,9,1,20,14 and 6.

Total No. of selected attributes: 16

4.2.3 Classifier Attribute Eval

Attribute selection algorithms are applied before applying data mining tasks such as classification, clustering, and outlier analysis. Classifier Attribute Eval is a feature selection method used to evaluate the importance of attributes (features) with respect to a classifier's performance. This technique helps to identify and select the most relevant attributes for building a predictive model.

Evaluator: weka.attributeSelection

Search: weka.attributeSelection

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal)): Classifier feature evaluator

Attribute Scores: Shown in Table 4.4

Table 4.4 : Attribute Score for Classifier Attribute Eval

Rank	Attribute/ Feature	Attribute ID
1	WARDS	20
2	HOUR	7
3	DAY_OF_WEEK	8
4	ZIP CODES	19
5	NUM_READS	6
6	BUS_COUNT	5
7	REGION	4
8	SPEED	3
9	REGION_ID	2
10	MONTH	9
11	DESCRIPTION	10
12	RECORD_ID	11
13	NW_LOCATION	16
14	COMMUNITY AREAS	18
15	SE_LOCATION	17
16	NORTH	15

The results confirm that the top five high scoring attributes or ranked attributes are Wards, HOUR, DAY_OF_WEEK, Zip Codes and NUM_READS with ranks 1, 2, 3, 4, and 5 respectively. The low ranked attributes are BUS_COUNT, REGION, SPEED, REGION_ID, MONTH, DESCRIPTION, RECORD_ID, NW_LOCATION, COMMUNITY AREAS, SE_LOCATION and NORTH with ranks 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 respectively. Based on the table above the 16 selected attributes are as shown below:

Selected attributes: 20,7,8,19,6,5,4,3,2,9,10,11,16,18,17 and 15

Total No. of selected attributes: 16

4.2.4 Cfs Subset Eval

Correlation-based Feature Selection Subset Evaluator it is feature selection method that evaluates the relevance and redundancy of attributes within a dataset based on their correlation with the class variable. The goal is to select a subset of attributes that are highly correlated with the class variable while minimizing redundancy among them.

Evaluator: weka.attributeSelection.CfsSubsetEval

Search: weka.attributeSelection.GreedyStepwise

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

Search Method: Greedy Stepwise (forwards).

Attribute Subset Evaluator (supervised, Class (nominal)): CFS Subset Evaluator

The results confirm that the top three high scoring attributes are SPEED, NUM_READS and MONTH, DAY_OF_WEEK, Zip Codes and NUM_READS with ranks 1, 2 and 3 respectively. The following 3 selected attributes are as follows:

Selected attributes: 3,6 and 19 (SPEED, NUM_READS and Zip Codes)

Total No. of selected attributes: 3

4.2.5 Gain Ratio Attribute Eval

Gain Ratio Attribute Eval is one of the method used for evaluating the attributes and finding the most appropriate one using the ability to discriminate between various classes. This method takes the concept of gain ratio basically which take into consideration the intrinsic information of the attribute and also the potential information gain.

Evaluator: weka.attributeSelection.GainRatioAttributeEval

Search: weka.attributeSelection.

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

Search Method: Attribute ranking.

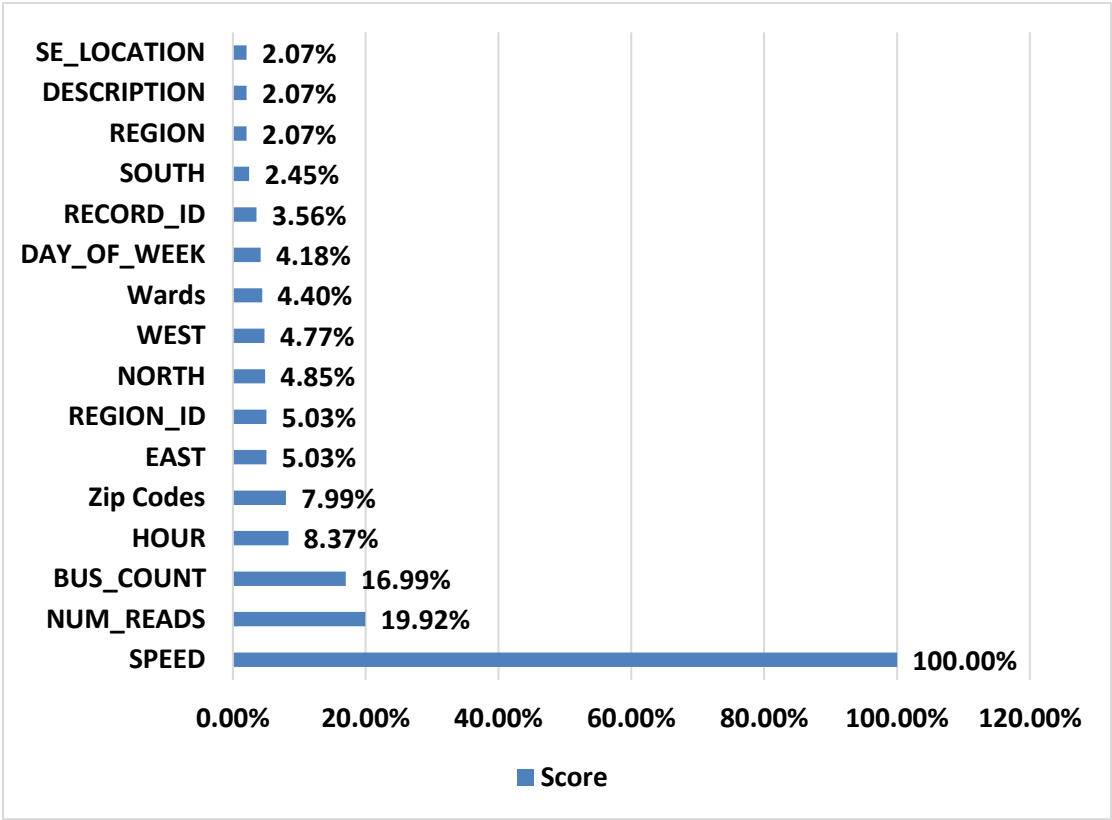
Attribute Evaluator (supervised, Class (nominal)): Gain Ratio feature evaluator

Attribute Scores: Shown in Table 4.5

Table 4.5: Attribute Score for Gain Ratio Attribute Eval

Score	Attribute/ Feature	Attribute ID
1	SPEED	3
0.1992	NUM_READS	6
0.1699	BUS_COUNT	5
0.0837	HOUR	7
0.0799	ZIP CODES	19
0.0503	EAST	13
0.0503	REGION_ID	2
0.0485	NORTH	15
0.0477	WEST	12
0.044	WARDS	20
0.0418	DAY_OF_WEEK	8
0.0356	RECORD_ID	11
0.0245	SOUTH	14
0.0207	REGION	4
0.0207	DESCRIPTION	10
0.0207	SE_LOCATION	17

Figure 4.3: Attribute Score for Gain Ratio Attribute Eval



The outcome confirm that the top five high scoring attributes are SPEED, NUM_READS, BUS_COUNT, HOUR and ZIP CODES with values 1, 0.1992, 0.1699, 0.0837 and 0.0799 respectively. The low scoring attributes are EAST, REGION_ID,

NORTH, WEST, Wards, DAY_OF_WEEK, RECORD_ID, SOUTH, REGION, DESCRIPTION and SE_LOCATION with values 0.0503, 0.0503, 0.0485, 0.0477, 0.044, 0.0418, 0.0356, 0.0245, 0.0207, 0.0207 and 0.0207 respectively. Based on the figure above the 16 selected attributes are as shown below:

Selected attributes: 3,6,5,7,19,13,2,15,12,20,8,11,14,4,10 and 17

Total No. of selected attributes: 16

4.2.6 OneR Attribute Eval

OneR Attribute Eval is a feature selection method that's based on the One Rule (OneR) classifier. The OneR algorithm is a simple and interpretable rule-based classification algorithm that selects a single attribute as the best predictor for classifying instances. The OneR Attribute Eval method evaluates the quality of attributes by measuring how well they serve as rules for classifying instances. OneR or "One Rule" is a simple, interpretable classification algorithm that is often used as a fast base model or as a benchmark for more complex algorithms. Attribute evaluation in the context of OneR refers to the process of selecting the best attributes to create classification rules. The goal is to find the attribute that by itself provides the most accurate prediction. For each attribute value, this algorithm creates a simple rule based on that value. Created rule is used to count the number of correct and incorrect classifications for calculating total error. The attribute with lowest error is selected as the One Rule. Finally classification rule is created on chosen attribute.

Evaluator: weka.AttributeSelection.OneRAttributeEval.

Search: weka. Attribute Selection.

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

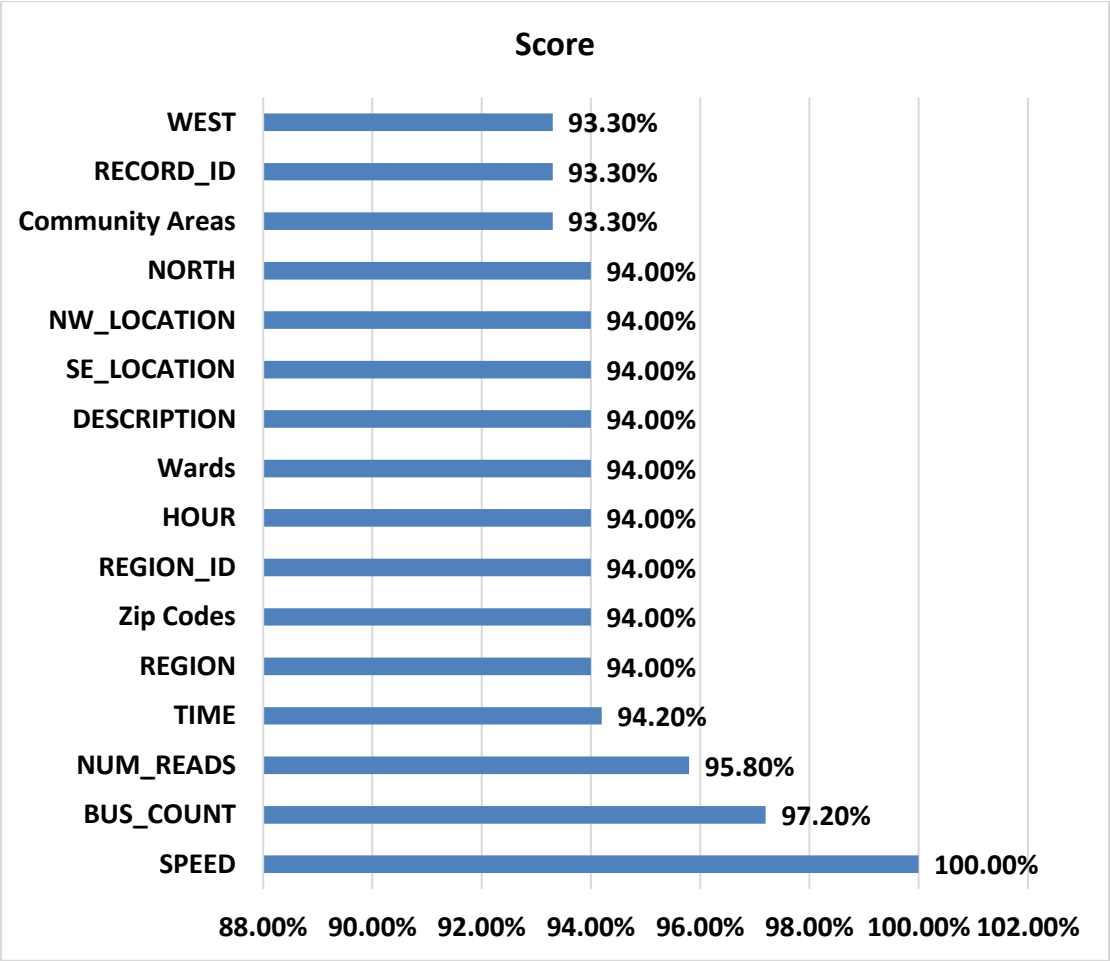
Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal)): OneR feature evaluator.

Table 4.6: Attribute Score for OneR Attribute Eval

Score	Attribute/ Feature	Attribute ID
1.00	SPEED	3
0.972	BUS_COUNT	5
0.958	NUM_READS	6
0.942	TIME	1
0.94	REGION	4
0.94	ZIP CODES	19
0.94	REGION_ID	2
0.94	HOUR	7
0.94	WARDS	20
0.94	DESCRIPTION	10
0.94	SE_LOCATION	17
0.94	NW_LOCATION	16
0.94	NORTH	15
0.933	COMMUNITY AREAS	18
0.933	RECORD_ID	11
0.933	WEST	12

Figure 4.4: Attribute Score for OneR Attribute Eval



The results confirm that the top five high scoring attributes are SPEED, BUS_COUNT, NUM_READS, TIME and REGION with values 1.00, 0.972, 0.958, 0.942 and 0.94 respectively. The low scoring attributes are ZIP CODES, REGION_ID, HOUR, WARDS, DESCRIPTION, SE_LOCATION, NW_LOCATION, NORTH, COMMUNITY AREAS, RECORD_ID and WEST with values 0.94, 0.94, 0.94, 0.94, 0.94, 0.94, 0.94, 0.933, 0.933 and 0.933 respectively. Based on the figure above the 16 selected attributes are as shown below:

Selected attributes: 3,5,6,1,4,19,2,7,20,10,17,16,15,18,11 and 12

Total No. of selected attributes: 16

4.2.7 ReliefF Attribute Eval

ReliefF Attribute Eval is a feature selection method based on the ReliefF algorithm. The ReliefF algorithm is designed to assess the importance of attributes in a dataset for classification tasks, particularly in the context of feature selection and ranking. It focuses on measuring the relevance and quality of attributes by considering the nearest neighbors of instances.

Evaluator: weka.attributeSelection.ReliefFAttributeEval

Search: weka.attributeSelection.

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

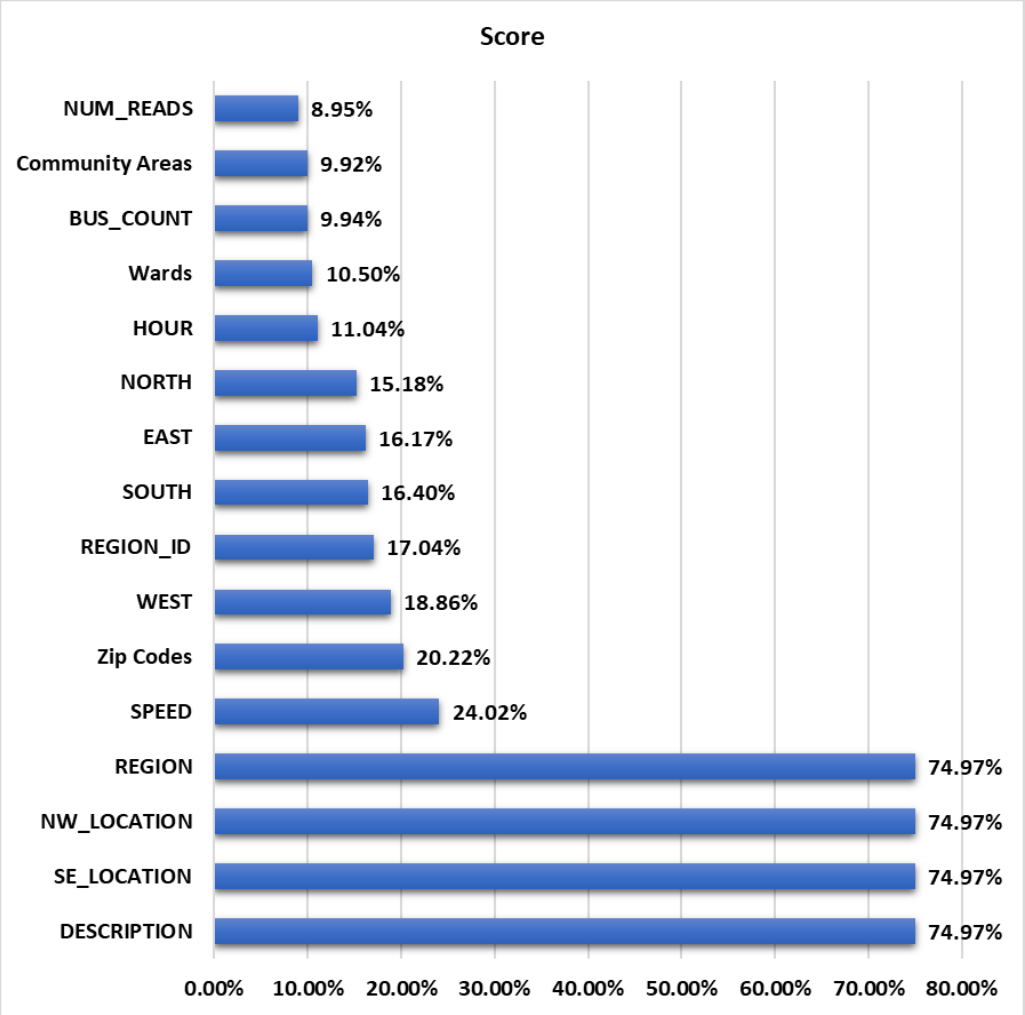
Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal)): ReliefF Ranking Filter

Table 4.7: Attribute Score for ReliefF Attribute Eval

Score	Attribute/ Feature	Attribute ID
0.7497	DESCRIPTION	10
0.7497	SE_LOCATION	17
0.7497	NW_LOCATION	16
0.7497	REGION	4
0.2402	SPEED	3
0.2022	ZIP CODES	19
0.1886	WEST	12
0.1704	REGION_ID	2
0.164	SOUTH	14
0.1617	EAST	13
0.1518	NORTH	15
0.1104	HOUR	7
0.105	WARDS	20
0.0994	BUS_COUNT	5
0.0992	COMMUNITY AREAS	18
0.0895	NUM_READS	6

Figure 4.5: Attribute Score for ReliefF Attribute Eval



The outcome confirm that the top five high scoring attributes are DESCRIPTION, SE_LOCATION, NW_LOCATION, REGION and SPEED with values 0.7497, 0.7497, 0.7497, 0.7497 and 0.2402 respectively. The low scoring attributes are ZIP CODES, WEST, REGION_ID, SOUTH, EAST, NORTH, HOUR, WARDS, BUS_COUNT, COMMUNITY AREAS and NUM_READS with values 0.2022, 0.1886, 0.1704, 0.164, 0.1617, 0.1518, 0.1104, 0.105, 0.0994, 0.0992 and 0.0895 respectively. Based on the figure above the 16 selected attributes are as shown below:

Selected attributes: 10,17,16,4,3,19,12,2,14,13,15,7,20,5,18 and 6

Total No. of selected attributes: 16

4.2.8 Symmetrical Uncert Attribute Eval

Symmetrical Uncert Attribute Eval Is one of the selection methods which follows the symmetric uncertainty principle. The symmetric uncertainty is basically a measure that finds the amount of information which is being shared between the two variables and that is two being very useful for finding the relevance of various attributes in a classification context. This approach mainly identifies the attributes or features which have a strong relationship with the class label variable While taking in consideration the potential interactions.

Evaluator: weka.attributeSelection.SymmetricalUncertAttributeEval

Search: weka.attributeSelection.

Relation: Chicago_Traffic_1000

Instances: 1000

Attributes: 21

Evaluation mode: Evaluate on all training data

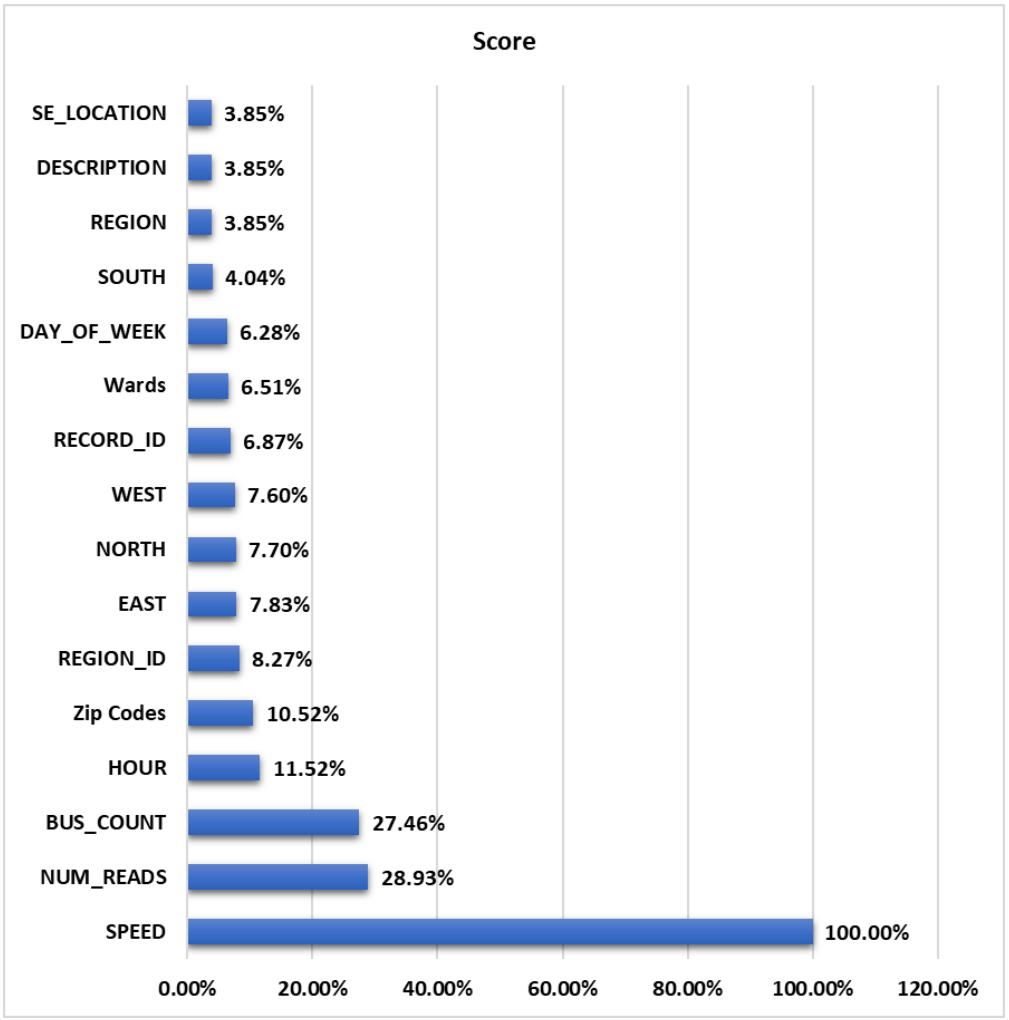
Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal)): Symmetrical Uncertainty Ranking Filter

Table 4.8: Attribute Score for Symmetrical Uncert Attribute Eval

Score	Attribute/ Feature	Attribute ID
1	SPEED	3
0.2893	NUM_READS	6
0.2746	BUS_COUNT	5
0.1152	HOUR	7
0.1052	ZIP CODES	19
0.0827	REGION_ID	2
0.0783	EAST	13
0.077	NORTH	15
0.076	WEST	12
0.0687	RECORD_ID	11
0.0651	WARDS	20
0.0628	DAY_OF_WEEK	8
0.0404	SOUTH	14
0.0385	REGION	4
0.0385	DESCRIPTION	10
0.0385	SE_LOCATION	17

Figure 4.6: Attribute Score for Symmetrical Uncert Attribute Eval



The result confirm that the top five high scoring attributes are SPEED, NUM_READS, BUS_COUNT, HOUR and ZIP CODES with values 1, 0.2893, 0.2746, 0.1152 and 0.1052 respectively. The low scoring attributes are REGION_ID, EAST, NORTH, WEST, RECORD_ID, WARDS, DAY_OF_WEEK, SOUTH, REGION, DESCRIPTION and SE_LOCATION with values 0.0827, 0.0783, 0.077, 0.076, 0.0687, 0.0651, 0.0628, 0.0404, 0.0385, 0.0385 and 0.0385 respectively. Based on the figure above the 16 selected attributes are as shown below:

Selected attributes: 10,17,16,4,3,19,12,2,14,13,15,7,20,5,18 and 6

Total No. of selected attributes: 16

4.3 Feature Extraction Using Multiple Regression

The scope of multiple regression can be expanded from finding the relationship between dependent and independent variables to getting insights for finding the appropriate attributes that is for doing feature selection.

Table 4.9: Variables Entered / Removed

Variables Entered/Removed[a]			
Model	**Variables Entered**	**Variables Removed**	**Method**
1	Wards, Hour, North, Month, Community Areas, Speed, Day_Of_Week, West, Num_Reads, East, Bus_Count, South[b]	.	Enter
a. Dependent Variable: Class Label			
b. All requested variables entered.			

Table 4.10: Model Summary

Model Summary				
Model	**R**	**R Square**	**Adjusted R Square**	**Std. Error of the Estimate**
1	.675[a]	.455	.445	.182
a. Predictors: (Constant), Wards, HOUR, NORTH, MONTH, Community Areas, SPEED, DAY_OF_WEEK, WEST, NUM_READS, EAST, BUS_COUNT, SOUTH				

Table 4.11: ANOVA Summary

ANOVA[a]						
Model		**Sum of Squares**	**df**	**Mean Square**	**F**	**Sig.**
1	Regression	18.323	12	1.527	46.262	.000[b]
	Residual	21.949	665	.033		
	Total	40.273	677			
a. Dependent Variable: Class Label						
b. Predictors: (Constant), Wards, HOUR, NORTH, MONTH, Community Areas, SPEED, DAY_OF_WEEK, WEST, NUM_READS, EAST, BUS_COUNT, SOUTH						

Table 4.12: Coefficients Summary

Coefficients[a]						
Model		**Unstandardized Coefficients**		**Standardized Coefficients**	**t**	**Sig.**
		B	**Std. Error**	**Beta**		
1	(Constant)	27.656	12.608		2.194	.029
	Speed	-.023	.001	-.561	-15.761	.000
	Bus_Count	.003	.001	.276	1.828	.068
	Num_Reads	.000	.000	-.179	-1.167	.244
	Hour	-.010	.002	-.212	-5.860	.000
	Day_Of_Week	.048	.012	.143	4.159	.000
	Month	-.040	.012	-.102	-3.223	.001
	West	1.955	.442	.502	4.419	.000
	East	-1.953	.536	-.461	-3.643	.000
	South	2.759	.766	.923	3.604	.000
	North	-3.396	.821	-1.072	-4.134	.000
	Community Areas	.000	.000	-.023	-.759	.448
	Wards	.000	.001	-.011	-.354	.724
a. Dependent Variable: Class Label						

The provided regression model summary indicates the relationships between predictor variables (Speed, Bus_Count, Num_Reads, Hour, Day_Of_Week, Month, West, East, South, North, Community Areas, Wards) and a dependent variable ("Class Label"). The coefficients show how changes in predictor variables affect the predicted outcome. Notably, variables such as "Speed" and "Hour" have moderate negative impacts, while "Day_Of_Week" and "West" have moderate positive impacts, all with statistical significance. "East" and "North" have strong negative impacts, while "South" has a

strong positive impact, all statistically significant. Other variables like "Bus_Count," "Num_Reads," "Month," "Community Areas," and "Wards" have smaller effects with varying statistical significance. These interpretations aid in understanding the importance and direction of influence of each predictor on the dependent variable. So indirectly the identified selected attributes are as follows:

Selected attributes: 3,5,6,7,8,9,12,13,14,15,18 and 20

Total No. of selected attributes: 12

4.4 Combined Feature Selection Matrix

The proposed combined feature selection matrix basically is an arrangement between types of feature selection method and the identified attributes after analysis. The Attribute names and their ID's are given below for the reference.

Table 4.13: Attribute Names and ID

Attribute ID	Attribute Name
1	TIME
2	REGION_ID
3	SPEED
4	REGION
5	BUS_COUNT
6	NUM_READS
7	HOUR
8	DAY_OF_WEEK
9	MONTH
10	DESCRIPTION
11	RECORD_ID
12	WEST
13	EAST
14	SOUTH
15	NORTH
16	NW_LOCATION
17	SE_LOCATION
18	COMMUNITY AREAS
19	ZIP CODES
20	WARDS
21	CLASS LABEL

Table 4.14: Combined Feature Selection Matrix

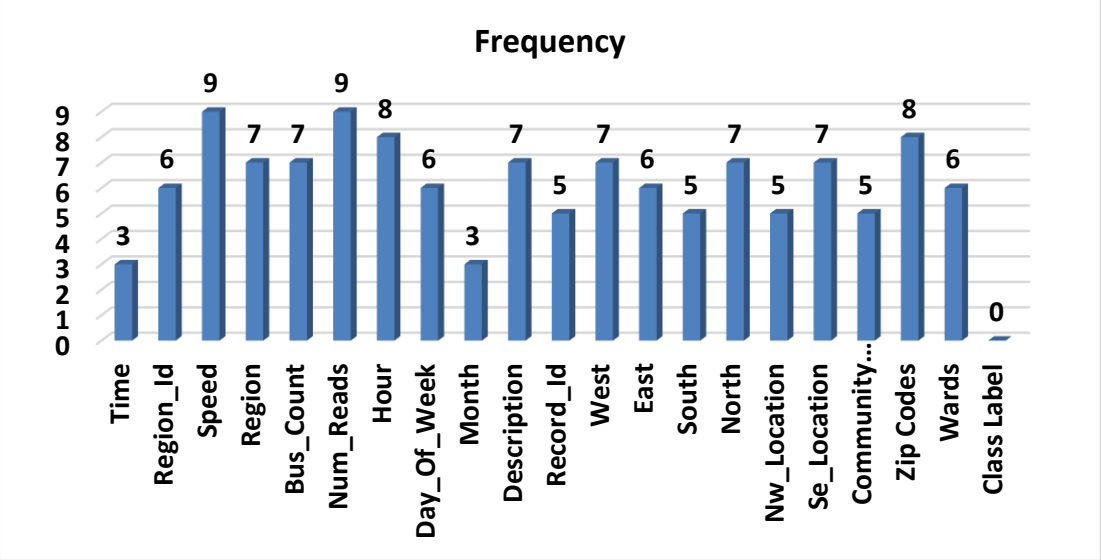
Info Gain Attribute Eval	**Correlation Attribute Eval**	**Classifier Attribute Eval**	**Cfs Subset Eval**	**Gain Ratio Attribute Eval**	**OneR Attribute Eval**	**ReliefF Attribute Eval**	**Symmetrical Uncert Attribute Eval**	**Multiple Regression**
11	3	20	3	3	3	10	3	3
3	8	7	6	6	5	17	6	5
5	7	8	19	5	6	16	5	6
6	19	19	-	7	1	4	7	7
4	12	6	-	19	4	3	19	8
10	13	5	-	13	19	19	2	9
17	10	4	-	2	2	12	13	12
16	4	3	-	15	7	2	15	13
1	16	2	-	12	20	14	12	14
2	17	9	-	20	10	13	11	15
15	18	10	-	8	17	15	20	18
12	9	11	-	11	16	7	8	20
7	1	16	-	14	15	20	14	-
13	20	18	-	4	18	5	4	-
19	14	17	-	10	11	18	10	-
8	6	15	-	17	12	6	17	-

The above table shows the 9 feature selection methods and various attributes being selected by them which are basically shown through the attribute ID in Table 4.14

Table 4.15: Attribute and Count

Attribute ID	Attribute Name	Count
1	TIME	3
2	REGION_ID	6
3	SPEED	9
4	REGION	7
5	BUS_COUNT	7
6	NUM_READS	9
7	HOUR	8
8	DAY_OF_WEEK	6
9	MONTH	3
10	DESCRIPTION	7
11	RECORD_ID	5
12	WEST	7
13	EAST	6
14	SOUTH	5
15	NORTH	7
16	NW_LOCATION	5
17	SE_LOCATION	7
18	COMMUNITY AREAS	5
19	ZIP CODES	8
20	WARDS	6
21	CLASS LABEL	0

Figure 4.7: Attribute and Frequency



The number of times the frequency of the particular attribute is identified from the combined feature selection matrix. From the figure above it is clear that attributes or features Speed and Num_Reads is having the highest frequency or occurrence with value 9 followed by attributes Hour, Zip_Codes with value 8 whereas the number of occurrences of attribute Class_Label, Month and Time is lowest.

Table 4.16: Overall Attribute Performance in Percentage (%)

Applied Attribute Evaluator ID	Attribute Name	Count	Overall Attribute Performance in Percentage (%)
1	TIME	3	14.29
2	REGION_ID	6	28.57
3	SPEED	9	42.86
4	REGION	7	33.33
5	BUS_COUNT	7	33.33
6	NUM_READS	9	42.86
7	HOUR	8	38.10
8	DAY_OF_WEEK	6	28.57
9	MONTH	3	14.29
10	DESCRIPTION	7	33.33
11	RECORD_ID	5	23.81
12	WEST	7	33.33
13	EAST	6	28.57
14	SOUTH	5	23.81
15	NORTH	7	33.33
16	NW_LOCATION	5	23.81
17	SE_LOCATION	7	33.33
18	COMMUNITY AREAS	5	23.81
19	ZIP CODES	8	38.10
20	WARDS	6	28.57
21	CLASS LABEL	0	0.00

The overall attribute contribution is being evaluated in percentage as shown above in the table. The performance percentages demonstrate the assessed significance of each attribute, ranging from 0% to 42.86%. Attributes like "Speed," "Num_Reads," "Hour," and "Zip Codes" received relatively higher performance percentages, implying they have notable influence or relevance in the context of the evaluation.

4.5 Rank and Percentile

Rank and percentile approach is one of the valuable techniques for the feature selection which also supports the identification of the most relevant attributes for the data analysis and modelling.

Table 4.17: Rank & Percentile Approach Results

Attribute Name	Attribute ID /Point	Overall Attribute Performance in Percentage (%)	Rank	Percentile
SPEED	3	42.86	1	95.00%
NUM_READS	6	42.86	1	95.00%
HOUR	7	38.10	3	85.00%
ZIP CODES	19	38.10	3	85.00%
REGION	4	33.33	5	55.00%
BUS_COUNT	5	33.33	5	55.00%
DESCRIPTION	10	33.33	5	55.00%
WEST	12	33.33	5	55.00%
NORTH	15	33.33	5	55.00%
SE_LOCATION	17	33.33	5	55.00%
REGION_ID	2	28.57	11	35.00%
DAY_OF_WEEK	8	28.57	11	35.00%
EAST	13	28.57	11	35.00%
WARDS	20	28.57	11	35.00%
RECORD_ID	11	23.81	15	15.00%
SOUTH	14	23.81	15	15.00%
NW_LOCATION	16	23.81	15	15.00%
COMMUNITY AREAS	18	23.81	15	15.00%
TIME	1	14.29	19	5.00%
MONTH	9	14.29	19	5.00%
CLASS LABEL	21	0.00	21	0.00%

The rank and percentile method were being use for further analysis of the various attributes using the overall attribute performance. Accordingly, the table above shows the evaluated rank and percentile. Evaluation seems to gauge the significance of each attribute in the context of the analysis. The "Rank" column indicates the attribute's rank based on performance, while the "Percent" column indicates the percentile of its performance among all attributes. Attributes like "Speed" and "Num_Reads" achieved the highest overall performance of 42.86%, securing the top rank and a percentile of 95.00%. "Hour" and "Zip Codes" follow closely with 38.10% performance and a joint

rank of 3, corresponding to an 85.00% percentile. Attributes such as "Region," "Bus_Count," "Description," "West," "North," and "Se_Location" share a performance of 33.33%, ranking 5th with a 55.00% percentile. Some attributes, including "Time," "Month," and "Class Label," had lower performance percentages, ranking lower with 14.29% and 0.00% performance, respectively. This ranking and percentile analysis offers insights into the relative importance of attributes within the dataset, helping to identify attributes that strongly contribute to the analysis and those with lesser impact.

4.6 Summary

Eight feature selection methods within WEKA machine learning software were executed to select and extract highly correlated features from Chicago_Traffic_1000 dataset with twenty one features. First six features along with class label were short listed after removing redundant and uncorrelated features for traffic congestion prediction model development. The selected features listed below.

1. SPEED
2. NUM_READS
3. HOUR
4. ZIP CODES
5. REGION
6. BUS_COUNT
7. CLASS LABEL