Machine Learning Project Report

1. Introduction

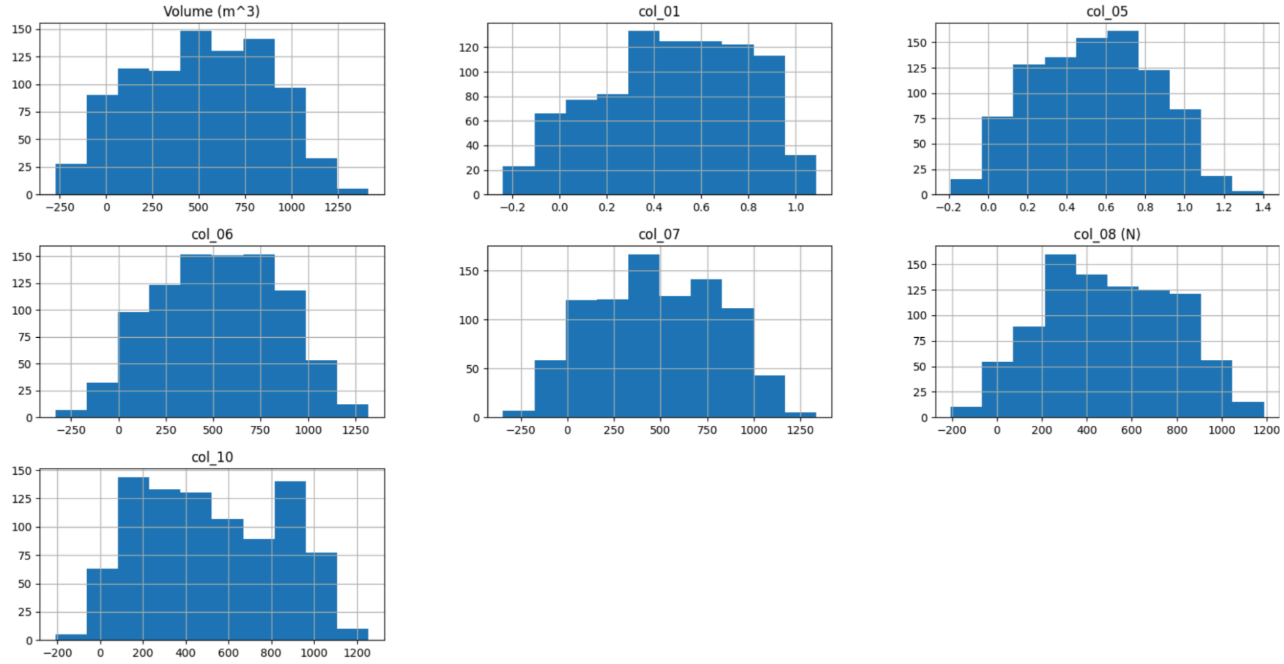
The initial given dataset consists of 11 total columns, and 902 data entries (rows). With the exception of the first column that provides a numeric label for each row, titled "label," the columns do not have a specific label that describes their data. Instead, they are labeled "col_00" through "col_10." The "label" column's data type is already assigned as integers (int64), and "col_01" and "col_06" are already assigned as floating-point values (float64). The rest of the columns, however, are simply assigned as objects. "Col_00" consists mainly of integer values in units of cubic meters, suggesting that these values represent some sort of volume measurement. "Col_01," "col_05," and "col_06," contain both negative and positive floating-point values without specific units of measurement. "Col_02," "col_03," "col_04," and "col_09" are made up of strings. "Col_02" consists of single city names, while "col_03" consists of single cities or lists of multiple cities. "Col_04" and "col_09" are made up of names that are repeated throughout both columns. This possibly suggests that the sample size of this dataset is smaller and more local. "Col_07," "col_08," and "col_10" are made up of both negative and positive integer values, with "col_08" having units of "N." It is unclear what this unit of measurement represents, especially given the content of the other columns. In this analysis, my goal is to determine what this data represents and predict the labels of the response variable.

2. Data Cleaning

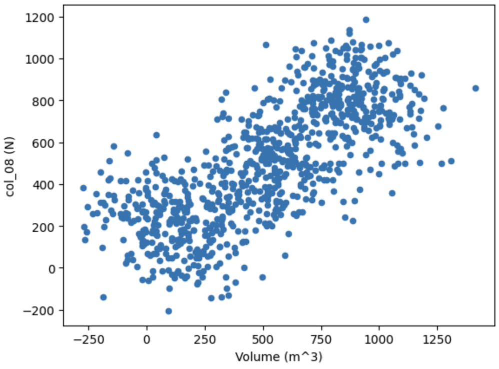
When cleaning the data, I first ensured that the data columns were not useless by attempting to drop columns who had an empty value rate of 50%. This did not drop any of the columns, confirming that all of the columns had sufficient data. Next, I extracted numbers from the numeric columns, removing trailing and leading whitespace, as well as removing all units of measurement from the data to make the numeric data machine readable. I then temporarily renamed col_00, col_02, col_03, col_04, and col_08 to "Volume (m^3)," "Location 1," "Location 2," "Person 1," and "Person 2" respectively to minimize confusion and improve readability. I then assigned data types to each column that is assigned the "object" data type, allowing future calculations to be performed. I then cleaned the string data columns by deleting rows with missing string values (marked as "?"). The reason behind this is because unlike floating-point values or integers, these missing string values are not easily cleanable by simply replacing them with zeroes or another non-skewing value such as the column's mean. Thus, it is necessary to delete these rows instead. After this, I fixed inconsistencies in the strings' capitalization by going through each string and calling python's built-in title() method, which converts the first letter in every word to a capital letter, and any capital letters that are not the first letter in a word are converted to lowercase. Finally, I cleaned the numeric data by replacing missing values (represented as numpy.nan) with the column's average. The reason behind this is to minimize any unintentional skewing that may come from simply replacing the missing values with zeros. I did not choose to delete these rows with missing values because deleting these rows would possibly remove too many data entries with existing valuable data. After data cleaning, the dataset was reduced to 898 rows.

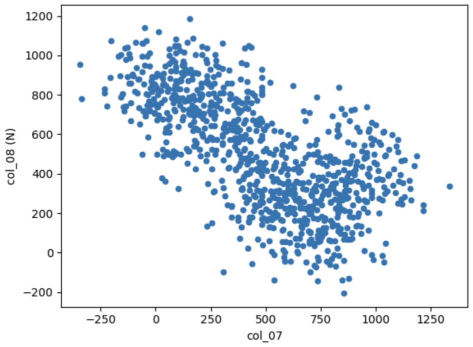
3. Data Visualization

After cleaning my data, I chose to run some basic visualization methods on my data, first translating my numerical data to histograms. Upon doing this, I noticed that my numerical data generally followed a standard distribution, with only a slight left skew in col_01. Below are the histograms.



Additionally, I created several scatter plots between the various numeric columns to try to find any correlation between values. First, I found it interesting that Volume (formerly col_00) has a positive correlation with col_08. Additionally, the values in col_07 have a negative correlation with the values in col_08. Below are the scatter plots of the two relationships.





4. Modeling

For the modeling portion, I applied three classifiers: Logistic Regression, K-Nearest Neighbors, and Decision Tree. My logistic regression classifier was configured with a maximum of 400 iterations and a random state of 30. For the KNN classifier, I set the number of neighbors to 20 to balance the trade-off between overfitting and underfitting. Finally, for the decision tree model, I set the random state to 30. The performance of these classifiers was then measured using a 5-fold cross-validation to measure the accuracy of each run of cross validation. I had to prepare the dataset for validation by scaling features using the StandardScalar() method. This normalized the data to have a mean of 0 and a standard deviation of 1. The results of cross validation indicated that the KNN classifier achieved the highest average accuracy, closely followed by logistic regression. While still high, the decision tree had a slightly lower performance than the other two classifiers. Both the KNN and logistic regression classifiers had the lowest standard deviations, indicating a more consistent performance across each run of cross validation. While only slightly higher, the decision tree recorded the largest standard deviation, suggesting variability in its predictions. Applying a Student's t-test on the classifiers indicated statistical significance between logistic regression and the decision tree, as well as between KNN and the decision tree. However, there was no statistical significance between logistic regression and KNN, suggesting that logistic regression and KNN are the clearly more accurate and reliable classifiers.

Model	**Mean Accuracy**	**Standard Deviation of Accuracy**
Logistic Regression	0.993	0.007
K-Nearest Neighbor	0.995	0.007
Decision Tree	0.953	0.010

5. Analysis

In my modeling stage, the KNN classifier emerged as the most accurate classifier out of the three, while maintaining the lowest standard deviation. However, KNN can tend to be computationally taxing and sensitive to hyperparameters. Close behind follows logistic regression, which is a solid, reliable classifier. Finally, while not completely performing poorly, the decision tree classifier was the lowest performing out of the three due to its tendency to overfit the data. This also explains the decision tree yielding a statistical significance between both the logistic regression classifier and the KNN classifier. While extremely marginal, I was able to slightly improve the decision tree's mean accuracy to 0.955 by setting the max_depth parameter to 7. This differed from my initial visualization, where I did not set a max_depth, which decreased average accuracy due to overfitting. While my data was thoroughly cleaned and smoothed out initially, it could have been possibly improved on by addressing outliers outside 2 standard deviations, in which there were a significant amount of data points. Overall, I applied many concepts from this class in this project. When data cleaning, I applied HO3's concepts. The extract_numbers() function in HO3 proved to be extremely helpful in removing unnecessary units of measurement in numerical data. Additionally, HO3's drop_sparse_columns was useful in ensuring that I had sufficient data in each data column. Finally, HO3's guess_types function allowed me to assign the appropriate data type to each column, enabling the data set to be machine readable. HO1's data visualization methods such as histograms and scatter plots proved to be crucial in my understanding of correlation between some of the data's variables. Finally, HO4's walkthrough of sklearn helped me create the different classifiers for the data set.

6. Conclusion

It still remains extremely difficult to derive meaningful conclusions and interpretations from the data set. The most probable interpretation is that the data set tracks specific statistics related to transportation and travel, but inconsistencies and holes within the data, like the arbitrary unit of measurement "N," makes it difficult to verify this interpretation. Through my visualization, I uncovered the standard distribution throughout the numerical data. Most interestingly, I discovered a possible positive correlation between the values of col_00 and the values of col_08, as well as a negative correlation between the values of col_07 and the values of col_08. Through my modeling of the data, logistic regression, k-nearest neighbors, and a decision tree all proved to be relatively accurate in capturing relationships between the data, with k-nearest neighbors emerging as the most reliable classifier. While a decision tree proved to be the most unreliable, it was slightly improved by modifying the max_depth to 7. There still exist areas for further improvement when it comes to my data cleaning, most specifically addressing outliers. Overall, further investigation into the context of the data is necessary to draw any meaningful conclusions.