**Predicting Vehicle Composition on Road Segments Using Multinomial Logistic Regression**

Model’s Goal:

The goal is to predict the vehicle composition on various road segments, identifying whether a segment is likely to experience "Heavy Traffic," "Light Traffic," or "Mixed Traffic." Predicting these compositions helps in making informed decisions about infrastructure changes, traffic management, and resource allocation.

The model chosen for this task is **Multinomial Logistic Regression** because it is specifically designed for **multi-class classification problems**, where the outcome can be one of several categories (one of several possible outcomes). In this case, the model predicts which traffic composition (heavy traffic, light traffic, mixed traffic) class a road segment will experience based on features like speed limit, vehicle counts, time of day, and road location.

It provides clear insights into how each feature (e.g., speed limit, vehicle counts) influences the likelihood of each traffic class (by generating coefficients). For each road segment, multinomial logistic regression predicts the **probability distribution** over the possible vehicle composition classes. The multinomial logistic regression model learns relationships between road segment features (e.g., speed limits, time of day, vehicle counts) and predicts the likelihood of each traffic composition class. It selects the class with the highest probability as the predicted outcome.

**Input Variables and Target Variable**

Input variables are data points that the model uses to make predictions about the target variable.

**Target Variable (y)**: The output we want to predict (vehicle composition: "Heavy Traffic," "Light Traffic," or "Mixed Traffic").

**Input Variables (X)**: The features or characteristics that the model uses to make its predictions (e.g., speed limit, vehicle counts, location, time of day).

The relationships between these variables are what the models learn during training.

Step 1: Data Preprocessing

**Defining Light and Heavy Vehicles**

I needed to define what counts as light vehicles and heavy vehicles to categorize road segments based on their traffic compositions.

Calculated proportions of light and heavy vehicles for each road segment:

**Total Vehicle Count**: First, I calculated the total number of vehicles by summing the counts of light vehicles, heavy vehicles, and other categories.

**Calculating Light Vehicle Proportion**: I calculated the ratio of light vehicles by considering the counts of light vehicles and light vehicles with trailers relative to the total number of vehicles.

**Calculating Heavy Vehicle Proportion**: Similarly, I calculated the ratio of heavy vehicles by summing the counts of trucks, buses, and other heavy vehicles relative to the total number of vehicles.

****

**Creating the Target Variable: Vehicle Composition (Y)**

Once I had the proportions of light and heavy vehicles, I defined the **vehicle composition** variable based on these ratios:

* **"Heavy Traffic"**: If heavy vehicles made up more than 50% of the total.
* **"Light Traffic"**: If light vehicles accounted for more than 70% of the total.
* **"Mixed Traffic"**: If neither of the above conditions was met.

A screenshot of a computer code

Description automatically generated

**Verifying vehicle composition column (checking distribution of categories across the dataset):**

**A screenshot of a computer

Description automatically generated**

**Defining Input Variables (X)**

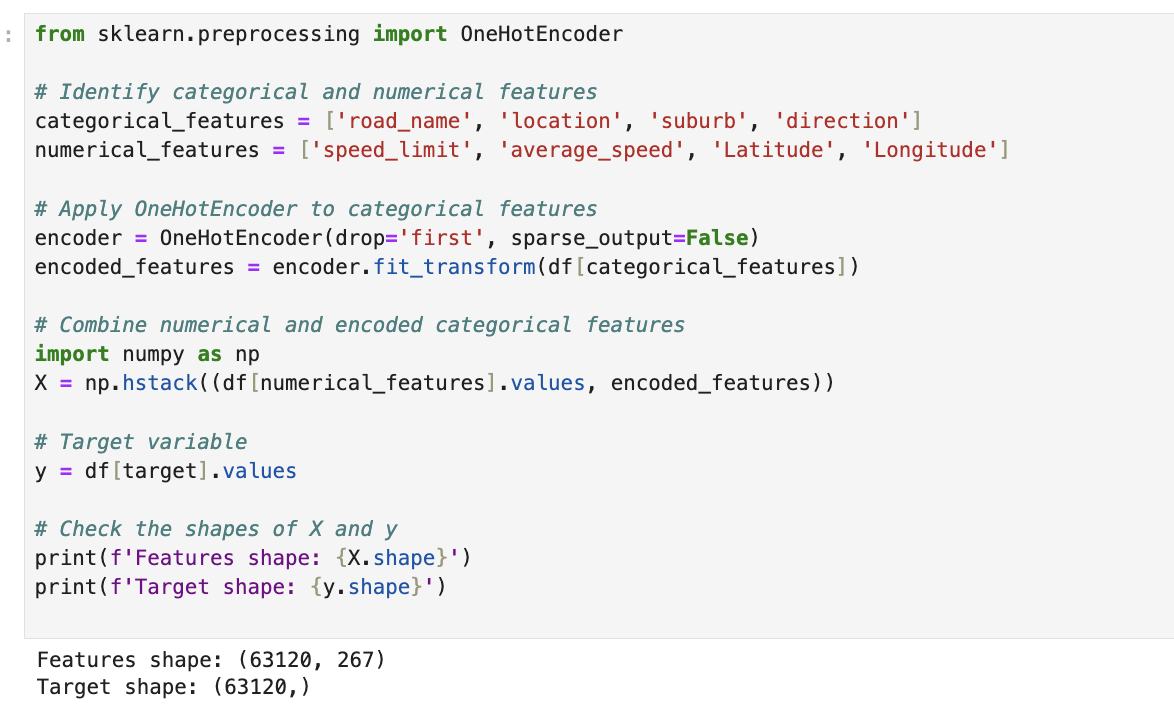
The input variables used in the model to predict the vehicle composition included several characteristics of each road segment:

1. **Speed Limit**: Maximum speed allowed on the road.
2. **Average Speed**: Observed average speed of vehicles.
3. **Latitude** and **Longitude**: Geographic coordinates of the road.
4. **Road Characteristics**: Road name, suburb, location.
5. **Direction**: The direction of traffic flow on the road segment.
6. **Vehicle Ratios**: Ratios of light and heavy vehicles.

These input features were used to train the model to predict the likelihood of each traffic composition class for each road segment.



**Identifying categorical and numerical features:**

****

**Train-Test Split**

To properly train and evaluate the model, I split the dataset into training and testing sets. I used 80% of the data for training and reserved 20% for testing to evaluate how well the model performs on unseen data.

This split ensures that the model is evaluated on data it has never seen during training, providing a more realistic measure of its performance.



**Standardizing the Input Variables**

Before training the model, I standardized the input features to ensure they had a mean of 0 and a standard deviation of 1. This was important because features like geographic coordinates and vehicle ratios have different ranges, and standardizing helps the model converge more effectively.

**Model Training: Multinomial Logistic Regression**

Once the data was pre-processed, I trained a **Multinomial Logistic Regression** model. This model is ideal for this task since it can handle more than two outcome classes (in this case, heavy, light, or mixed traffic).

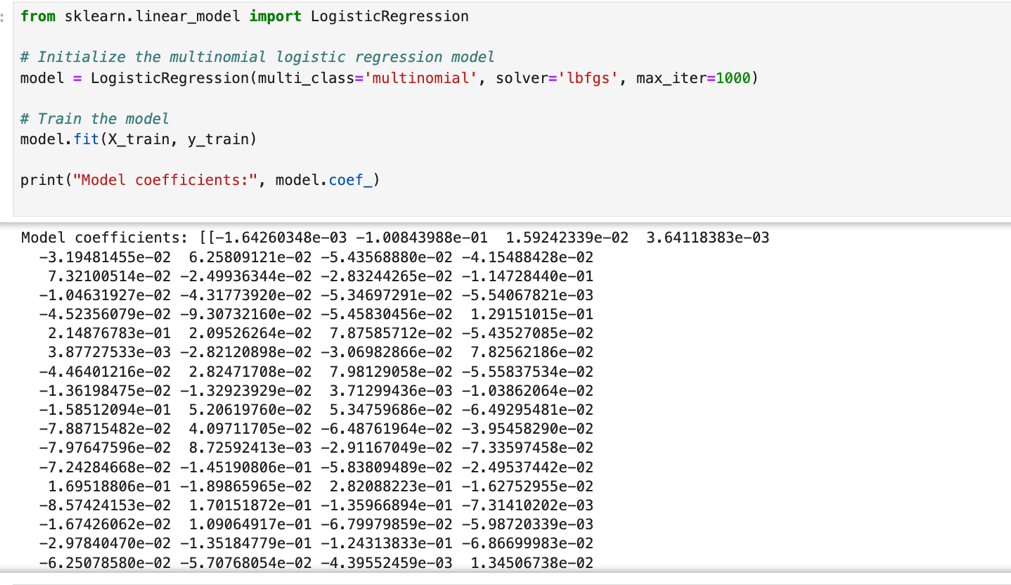
**Multi\_class= ‘multinomial’ : dealing with multi-class classification**

Set max\_iter= 1000 (setting maximum number of iterations to be 1000).

The solver iterates through the training data multiple times, adjusting the model's coefficients to minimize the prediction error.

**fit()**: This method is used to train the model. It takes two inputs:

* X\_train\_scaled: The input features for the training data that have been standardized (scaled to have a mean of 0 and a standard deviation of 1). These features describe the road segments (e.g., speed limit, vehicle ratios, etc.).
* y\_train: The target variable for the training data. This is the **vehicle composition** (i.e., "Heavy Traffic," "Light Traffic," or "Mixed Traffic") for each road segment.

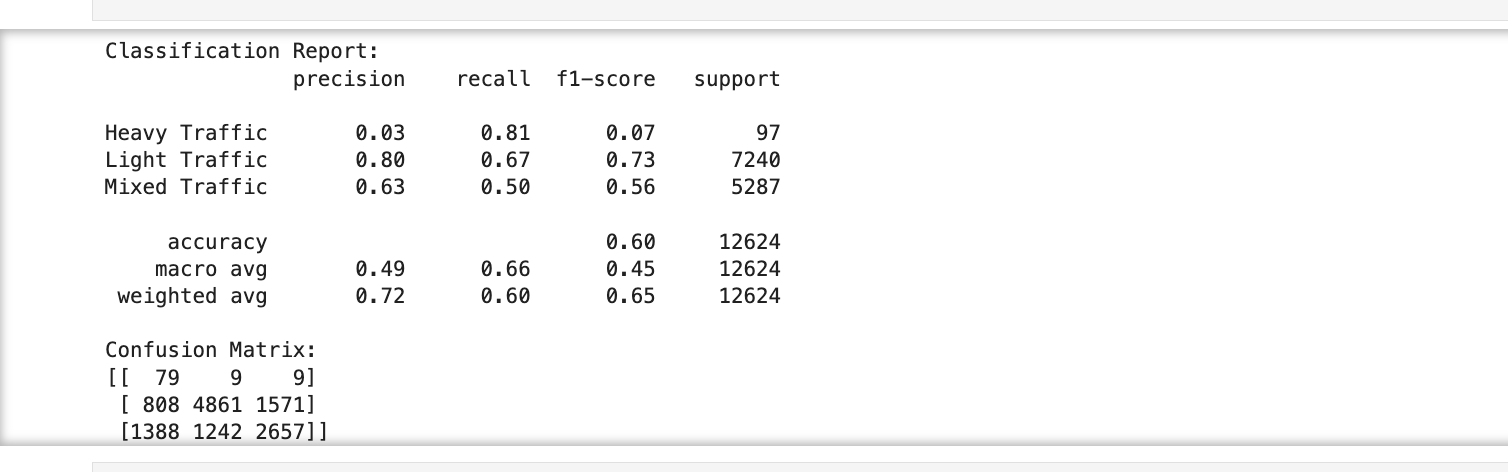


What are Coefficients in Logistic Regression?

Each coefficient corresponds to one of the input features (like speed limit, average speed, vehicle ratios, etc.) and tells you how much that feature affects the probability of predicting a certain class (either "Heavy Traffic," "Light Traffic," or "Mixed Traffic").

* A **positive coefficient** means that as the value of that feature increases, the odds of the model predicting that specific traffic class **increase**.
* A **negative coefficient** means that as the value of that feature increases, the odds of the model predicting that specific traffic class **decrease**.

1. **Rows in the Coefficients Table**: Each row corresponds to a feature (input variable) in your model, such as speed limit, average speed, or vehicle ratios. Each of these rows has three values—one for each traffic class.
2. **Columns in the Coefficients Table**: Each column corresponds to a different traffic class ("Heavy Traffic," "Light Traffic," and "Mixed Traffic"). So the values tell you how much each feature influences the probability of predicting one of these traffic classes.



**Classification report and confusion matrix**

The classification report shows how well the model predicts the three traffic classes: **Heavy Traffic**, **Light Traffic**, and **Mixed Traffic**.

The classification report provides metrics such as precision, recall, and F1-score for each of the traffic categories: Heavy Traffic, Light Traffic, and Mixed Traffic. These metrics are based on the model's performance on the test set.

1. **Precision**: How many of the predicted labels were actually correct.
2. **Recall**: How many actual instances of a class were correctly identified by the model.
3. **F1-score**: A balance between precision and recall, giving an overall performance measure.
4. **Support**: The number of actual instances for each class.

Here’s what each row in your report tells you:

* **Heavy Traffic**:
  + **Precision**: 0.03 – Only 3% of the times the model predicted "Heavy Traffic" was it actually correct.
  + **Recall**: 0.81 – The model found 81% of all actual "Heavy Traffic" cases.
  + **F1-score**: 0.07 – A low score overall because precision is so low, meaning the model struggles with correctly predicting this class.
* **Light Traffic**:
  + **Precision**: 0.80 – 80% of the times the model predicted "Light Traffic" was it correct.
  + **Recall**: 0.67 – The model identified 67% of actual "Light Traffic" cases.
  + **F1-score**: 0.73 – A good balance between precision and recall, indicating the model performs well for this class.
* **Mixed Traffic**:
  + **Precision**: 0.63 – 63% of "Mixed Traffic" predictions were correct.
  + **Recall**: 0.50 – The model identified 50% of the actual "Mixed Traffic" cases.
  + **F1-score**: 0.56 – Moderately good performance, but there’s room for improvement.
* **Overall (accuracy, macro avg, weighted avg)**:
  + **Accuracy**: 60% – The model correctly predicted the class for 60% of all cases.
  + **Macro avg**: 0.49 – This is the average performance across all classes, treating them equally.
  + **Weighted avg**: 0.72 – This average gives more importance to classes with more instances, which is why it's higher.

**Confusion matrix**

A confusion matrix helps visualize the number of correct and incorrect predictions made by the model.

A screenshot of a graph

Description automatically generated

* **Rows** represent the **actual labels** (what the true class is).
* **Columns** represent the **predicted labels** (what the model predicted).

For example:

* **First row** (Heavy Traffic):
  + **79**: The model correctly predicted "Heavy Traffic" 79 times.
  + **9**: The model incorrectly predicted "Light Traffic" instead of "Heavy Traffic" 9 times.
  + **9**: The model incorrectly predicted "Mixed Traffic" instead of "Heavy Traffic" 9 times.
* **Second row** (Light Traffic):
  + **4861**: The model correctly predicted "Light Traffic" 4861 times.
  + **808**: The model incorrectly predicted "Heavy Traffic" instead of "Light Traffic."
  + **1571**: The model incorrectly predicted "Mixed Traffic" instead of "Light Traffic."
* **Third row** (Mixed Traffic):
  + **2657**: The model correctly predicted "Mixed Traffic" 2657 times.
  + **1388**: The model incorrectly predicted "Heavy Traffic" instead of "Mixed Traffic."
  + **1242**: The model incorrectly predicted "Light Traffic" instead of "Mixed Traffic."

**Interpretation:**

1. **Heavy Traffic**:
   * The model is **good at recalling** (finding) most of the "Heavy Traffic" cases (81% recall), but it **struggles with precision** (only 3% of its predictions were correct). This means the model is often confused when predicting this class.
2. **Light Traffic**:
   * The model is **best at predicting Light Traffic** with an F1-score of 0.73. It has the highest precision (80%) and recall (67%), meaning it does a decent job of correctly predicting and identifying this class.
3. **Mixed Traffic**:
   * The model’s performance on "Mixed Traffic" is moderate. The F1-score is 0.56, meaning it correctly predicts 50% of the actual instances and gets 63% of its predictions for "Mixed Traffic" right.

**Overall Conclusion:**

* The model performs well for **Light Traffic**, but has difficulty with **Heavy Traffic** (low precision) and **Mixed Traffic** (moderate performance).

-Need to handle imbalance for better prediction of heavy traffic

**Cross Validation**

-testing model’s performance more thoroughly by splitting data into parts: (5 parts)

A screenshot of a computer code

Description automatically generated

**Average cross-validation score**: The average score across all 5 folds is **73.63%**, which means, on average, the model performs with about **73.63% accuracy**.

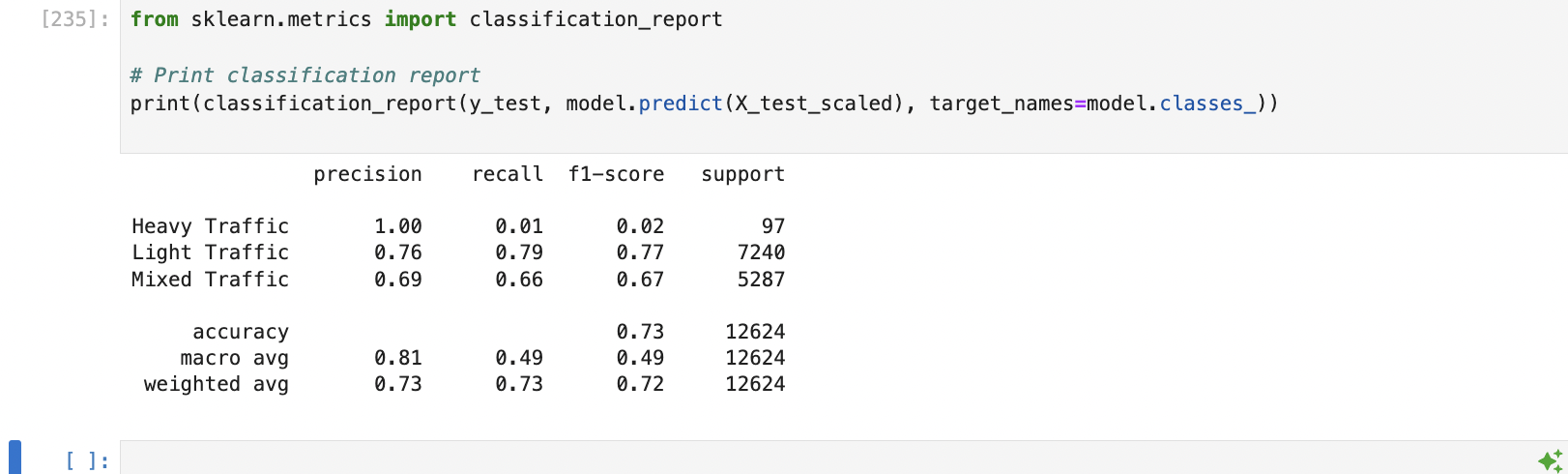
**Testing on rows for demonstration**

**Real World Example**: Take a hypothetical scenario:

* "Imagine we are using this model for a major highway during rush hour. Based on historical data and current road conditions, the model predicts that there is an 85% chance of Heavy Traffic, which could indicate the need for more lanes or traffic regulation."

The model takes in this data (features like speed limit, road segment, vehicle types, etc.) and predicts the **probability of each traffic composition** (Heavy, Light, or Mixed Traffic).

 **Prediction Output**: For example, for a specific road segment at a certain time of day, the model may predict that there’s a 60% chance of "Heavy Traffic," 30% chance of "Light Traffic," and 10% chance of "Mixed Traffic."



**Interpretation of Your Results:**

**1. Heavy Traffic:**

* **Precision**: 1.00 (meaning whenever the model predicted Heavy Traffic, it was correct 100% of the time).
* **Recall**: 0.01 (the model identified only 1% of the actual Heavy Traffic instances).
* **F1-score**: 0.02 (low F1-score indicates a poor balance between precision and recall).
* **Support**: 97 (there are only 97 actual instances of Heavy Traffic in your test set).

**Explanation**: The model is highly confident when it predicts Heavy Traffic, but it rarely predicts this class, meaning it's missing most actual Heavy Traffic cases (low recall).

**2. Light Traffic:**

* **Precision**: 0.76 (out of all predicted Light Traffic cases, 76% were correct).
* **Recall**: 0.79 (out of all actual Light Traffic cases, 79% were correctly predicted).
* **F1-score**: 0.77 (good balance between precision and recall).
* **Support**: 7,240 (this is the most common class in your test set).

**Explanation**: The model performs reasonably well for Light Traffic. Precision and recall are close, which indicates that the model is well-calibrated for this class.

**3. Mixed Traffic:**

* **Precision**: 0.69 (out of all predicted Mixed Traffic cases, 69% were correct).
* **Recall**: 0.66 (the model caught 66% of the actual Mixed Traffic cases).
* **F1-score**: 0.67 (moderate balance between precision and recall).
* **Support**: 5,287 (this class is also common in the test set).

**Explanation**: The model performs moderately well for Mixed Traffic. It's less accurate than for Light Traffic, but still provides usable predictions.

**Overall Model Performance:**

* **Accuracy**: 0.73 (The model correctly predicted the traffic class in 73% of cases overall).
* **Macro avg**: 0.81 precision, 0.49 recall, and 0.49 F1-score. This average gives equal weight to all classes, showing that Heavy Traffic is dragging down the overall recall.
* **Weighted avg**: 0.73 precision, 0.73 recall, and 0.72 F1-score. This average is weighted by the number of instances in each class (meaning Light Traffic has a bigger impact).

Things to work further on:

Based on Time

-how each segment are affected

-predict for each segment

-use a filter

Predicting vehicle composition for each specific segment