README: U.S. DOT ROADII Mobility Traffic Counts

1. Project Description

ROADII Background

Research, Operational, and Artificial Intelligence Data Integration Initiative (ROADII) is a multi-year initiative led by the United States Department of Transportation (U.S. DOT) Intelligent Transportation Systems Joint Program Office (ITS JPO).

ROADII's vision is to expand U.S. transportation agencies' (regional, state, local, tribal, etc.) access to advanced data analytics knowledge and resources including Artificial Intelligence (AI) and Machine Learning (ML). The ROADII team:

i) Identifies and evaluates **use cases** that can benefit from advanced data analytics, AI, and ML
ii) Develops **proofs-of-concept** for use cases
iii) **Engages stakeholders** with proofs-of-concept and refine based on stakeholder feedback
iv) **Makes advanced data analytics, AI, and ML tools** available to the public at a central location (e.g., ITS CodeHub)

The processes and tools developed under ROADII will enable data scientists, researchers, and data providers to test and share new transportation-related AI algorithms; to develop high-value and well-documented AI training datasets; to reduce the barriers of applying AI approaches to transportation data; and to train future transportation researchers.

For more information, visit ITS JPO here.

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

ROADII Use Case 29 – Mobility Traffic Counts

Full Title: "High-Resolution Mobility Traffic Count Estimation for Modeling, Planning, and Environmental Impact Applications"

Purpose and goals of the project: The Mobility Traffic Counts code base geographically matches **traffic counting station data** with **probe-collected speed data** on the U.S. National Highway System (NHS), to produce training datasets for roadway traffic volume prediction across the entire road system. The code provides a Graphical User Interface (GUI) to easily load input data, select input and target columns, and train a model using basic AI neural network methods.

Figure 1 shows traffic speed data on NHS roadway links. The speed data originate from U.S. DOT National Performance Management Research Dataset (NPMRDS) managed by the Regional Integrated Transportation Information System (RITIS).

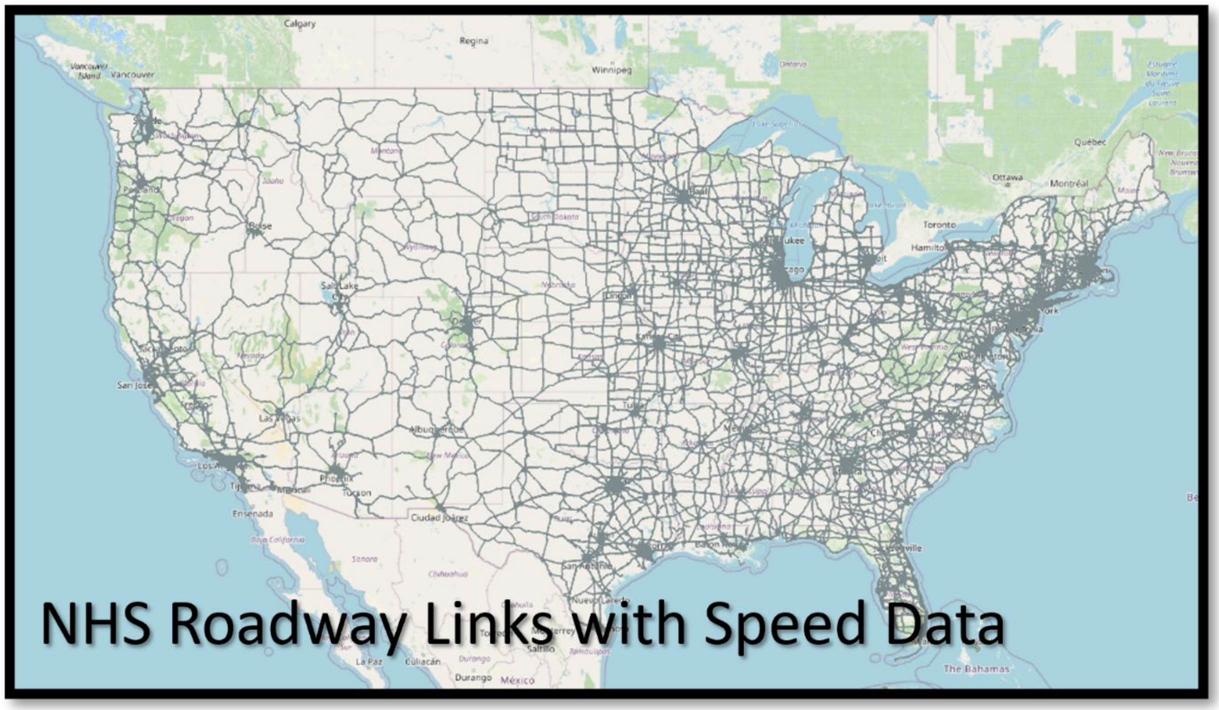


Figure 1. NHS Roadway Links with Speed Data

Figure 2 shows the locations of over 8,000 U.S. Federal Highway Administration (FHWA), Travel Monitoring Analysis System (TMAS) stations for traffic counting and classification.

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

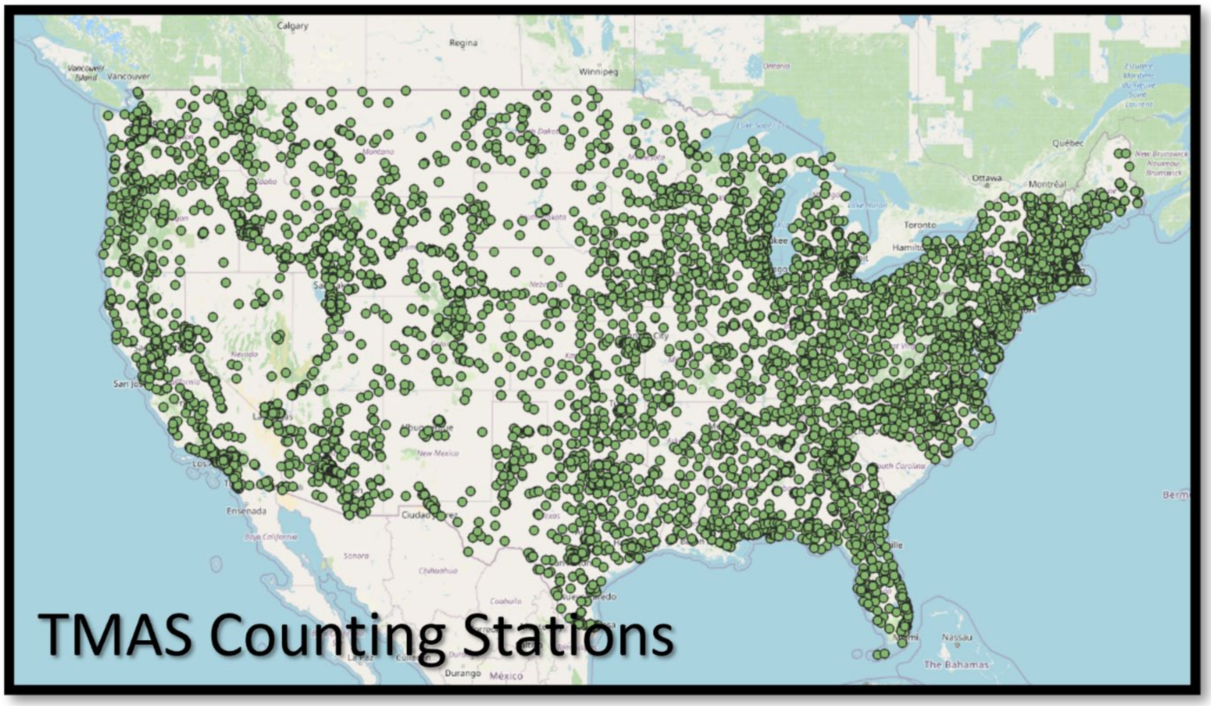
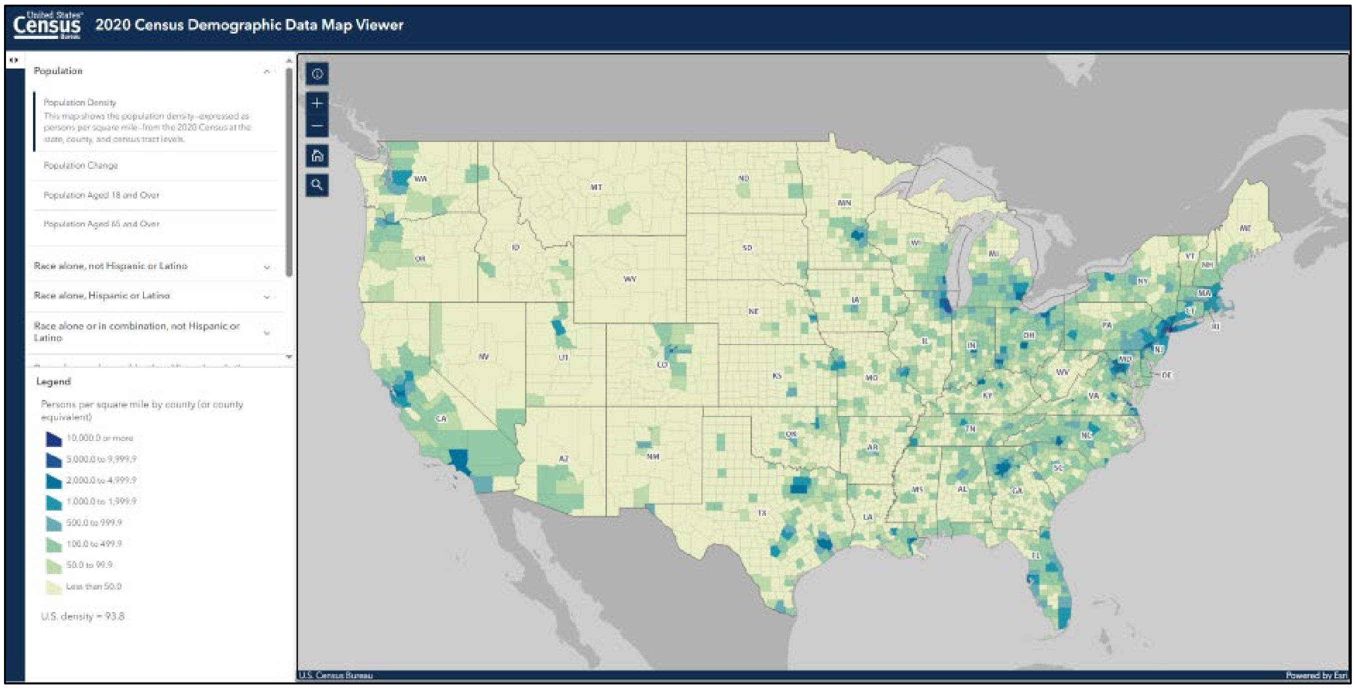


Figure 2. TMAS Traffic Counting Stations

Figure 3 shows U.S. Census 2020 Population Density by County as an example of the Census data used in the code base. The code base uses NHS roadway links where traffic counts and speed data are available, along with Census data; to perform prediction for NHS roadway links having similar Census data characteristics where traffic counts and/or speed data are not available.


United States Census Bureau
2020 Census Demographic Data Map Viewer
Population
Population Density
Population Change
Population Aged 18 and Over
Population Aged 65 and Over
Race alone, not Hispanic or Latino
Race alone, Hispanic or Latino
Race alone or in combination, not Hispanic or Latino
Legend
Persons per square mile by county (or county equivalent)
10,000.0 or more
5,000.0 to 9,999.9
2,000.0 to 4,999.9
1,000.0 to 1,999.9
500.0 to 999.9
100.0 to 499.9
50.0 to 99.9
Less than 50.0
U.S. density = 93.8
U.S. Census Bureau
Powered by Esri

This map shows the population density–expressed as persons per square mile–from the 2020 Census at the state, county, and census tract levels.

Figure 3. U.S. Census 2020 Population Density by County (Retrieved from https://maps.geo.census.gov/ddmv/map.html)

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

Purpose of the source code and how it relates to the overall goals of the project: This code base will make it easier and more approachable for transportation agencies to develop a simple neural network model to output historical traffic count data on NHS roadway links for which real world measured counts are not available. This is the case for most NHS roadway links. The intended user base includes state and local agencies looking to produce and use more complete traffic speed and traffic volume datasets. Applications of these resulting datasets and the code in this repository include highway planning projects and highway management projects, as well as future forecasting efforts.

Length of the project: The code base is currently in development. The ROADII team will update this repository as stable builds of the code are created. Development and testing will likely continue through spring 2024.

2. Prerequisites

Requires:

- Installation of Python 3.6.0 or later
- Installation of Python packages listed in <*requirements.txt*>
- Command prompt application to run the source code from the current working directory

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

3. Usage

Building the Mobility Traffic Counts Model

The ml folder contains the modules and classes to read in the requisite training data for building the mobility counts model. The following modules are contained therein:

- `main.py`: Produces a Streamlit GUI that reads the training data files, normalizes all columns to numerical types, and runs a training loop on the normalized data to produce a neural network to predict the user-chosen target column. The Streamlit application can be opened in any web browser at "localhost:8501"
- `use_model.py`: Uses a cached or pickled model file (.pt/.pkl format) to produce traffic count estimates. Also provides an easier, script based methodology to train a new model version without using the Streamlit application or interface. This is useful for more rapid model iteration
- `setup_funcs.py`: Set up the various data sources for training the model
- `module_data.py`: Reads, formats, and joins the various data sources into a single training dataset. This includes the Traffic Monitoring and Analysis System traffic volume data and the National Performance Measurement Research Data Set speed data
- `module_census.py`: Connects the training data to census information to improve model performance
- `module_ai.py`: Defines the ML training loop, the model architecture, and saves the resulting model for later use. Also provides methods to use a saved or cached model file

Testing the Model

The ROADII team is currently building testing functions for this code and will update the repository when those testing functions are available.

Executing the Model

The steps to run the model training algorithm are as follows:

1. Download the TMAS data for the analysis year of interest from the following website: DANA Tool Input Data Installers
2. Download the NPMRDS data for the analysis region of interest from the following website: RITIS NPMRDS Analytics Site
3. Download the included "TMC_Matches_2021.csv" file
4. In a command prompt application (e.g., Anaconda Prompt), execute the line → `pip install -r requirements.txt` to ensure all necessary Python packages are up to date
5. Create a new folder at the same level as the "ml" folder named "data" and place "NPMRDS_TMC_TMAS_NE_C.csv" in "data"
6. Create a new folder at the same level as the "ml" folder" named "models"; the models trained by the source code are saved in this folder
7. Update the data file paths and directory file paths in `main.py` according to your working directory

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

8. Run `main.py` to produce the Streamlit GUI. For instance, execute the line in a command prompt application → `streamlit run main.py`
9. After the GUI has loaded successfully, click the "choose source data file" button to select input data
10. After the GUI populates the "Raw Data" and "Normalized Training Data" viewing panes, use the "choose input columns" drop-down menu to select input data
11. After selecting the desired input columns for training, use the "choose a target column" drop-down menu to select the data field to predict, the results of which will be saved in a trained AI model
12. After a model has been trained successfully and saved in "models"; click "use model"

The following figures show in-development screenshots from the Streamlit GUI. Figure 4 shows the user's ability to choose a source data file as the input dataset. If the user clicks on the "Choose source data file" button, then a dialog box opens, and the user may explore files and select a source data file.

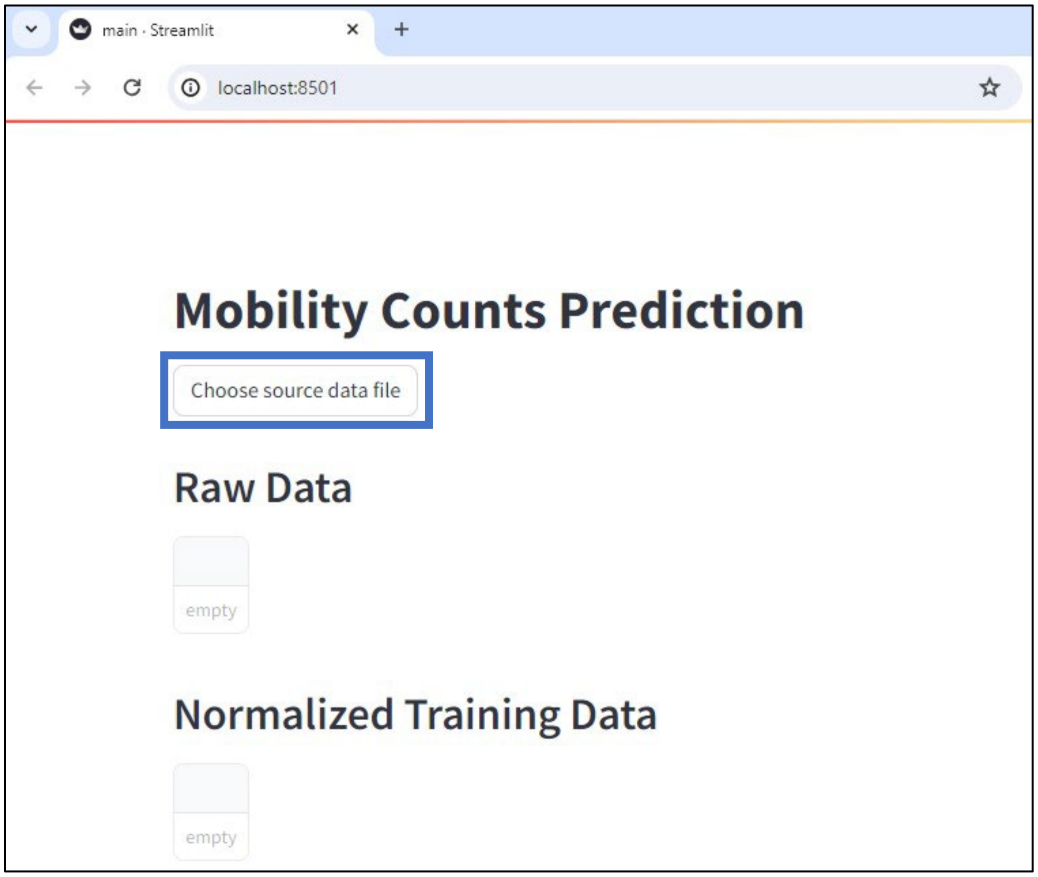
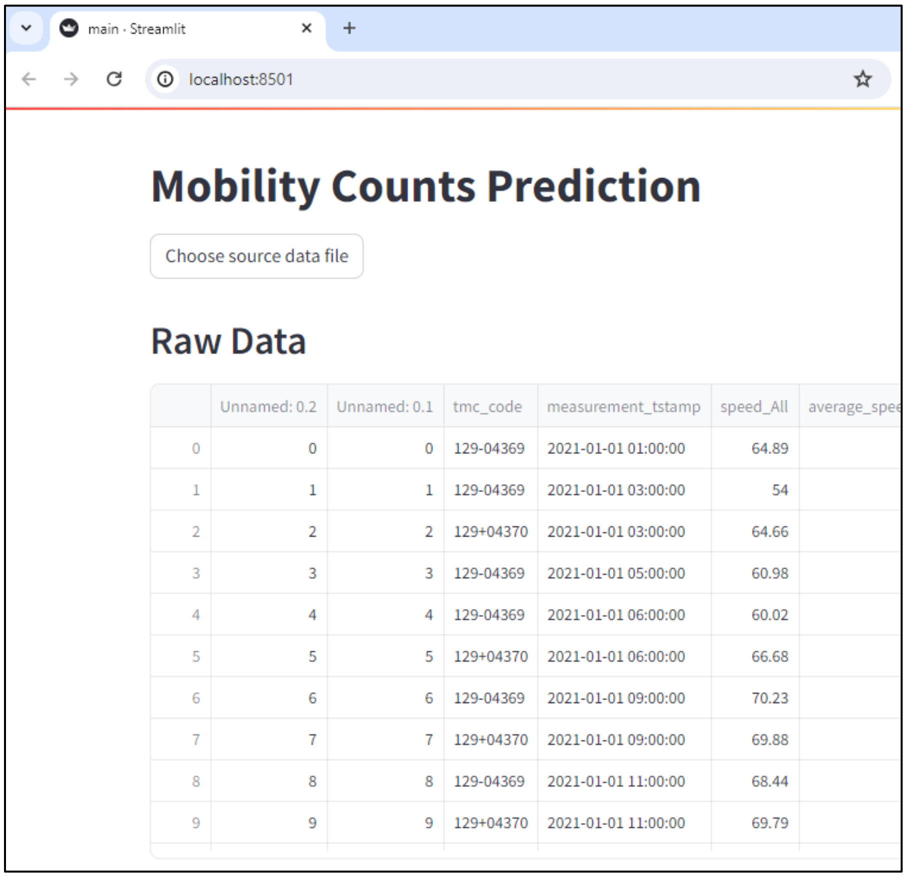


Figure 4. Streamlit GUI – User Chooses Source Data File

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

Once the user chooses a source data file, Figure 5 shows the user's ability to view an abridged sample of the input data. In addition, the code base normalizes non-numerical data columns and the Streamlit GUI displays this "normalized" data in a separate pane in the same format as Figure 5, this "normalized" data pane is not shown.



Mobility Counts Prediction

Raw Data

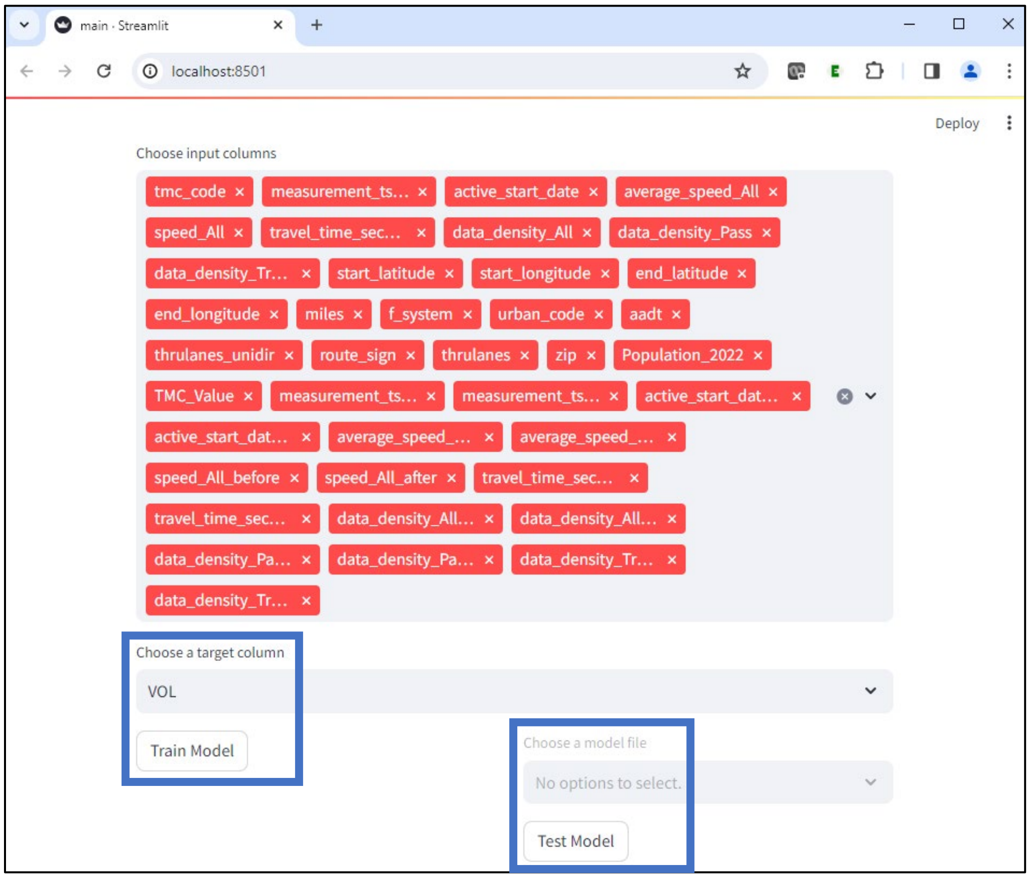
	Unnamed: 0.2	Unnamed: 0.1	tmc_code	measurement_tstamp	speed_All	average_spee
0	0	0	129-04369	2021-01-01 01:00:00	64.89	
1	1	1	129-04369	2021-01-01 03:00:00	54	
2	2	2	129+04370	2021-01-01 03:00:00	64.66	
3	3	3	129-04369	2021-01-01 05:00:00	60.98	
4	4	4	129-04369	2021-01-01 06:00:00	60.02	
5	5	5	129+04370	2021-01-01 06:00:00	66.68	
6	6	6	129-04369	2021-01-01 09:00:00	70.23	
7	7	7	129+04370	2021-01-01 09:00:00	69.88	
8	8	8	129-04369	2021-01-01 11:00:00	68.44	
9	9	9	129+04370	2021-01-01 11:00:00	69.79	

Figure 5. Streamlit GUI – User Views Input Data

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

Figure 6 shows the user's ability to choose input data columns and the target data column in AI model training. The input data columns should not include the target column. After the user chooses input data columns and the target data column and clicks "Train Model" – then AI model training is initiated and the user will start to see in-progress results in their command prompt application. After AI model training is complete, the code base saves an AI model file to the sub-directory "..\models."

In addition, the user may test a previously-generated AI model without training an AI model. If the user clicks "Test Model" – then a dialog box opens, and the user may explore files and select an AI model file. Typically, AI model files are saved in the sub-directory "..\models."



Choose input columns: tmc_code, measurement_ts..., active_start_date, average_speed_All, speed_All, travel_time_sec..., data_density_All, data_density_Pass, data_density_Tr..., start_latitude, start_longitude, end_latitude, end_longitude, miles, f_system, urban_code, aadt, thrulanes_unidir, route_sign, thrulanes, zip, Population_2022, TMC_Value, measurement_ts..., measurement_ts..., active_start_dat..., active_start_dat..., average_speed_..., average_speed_..., speed_All_before, speed_All_after, travel_time_sec..., travel_time_sec..., data_density_All..., data_density_All..., data_density_Pa..., data_density_Pa..., data_density_Tr..., data_density_Tr...

Choose a target column: VOL

Choose a model file: No options to select.

Figure 6. Streamlit GUI – User Selects Input Data for AI Model Training on a Targeted Metric. Alternatively, User May Test Previously Generated AI Models

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

If the user chooses to train an AI model, Figure 7 shows the AI model training progress with a real-time updating graph on the Streamlit GUI. The x-axis is the number of AI training epochs; the user may set the number of training epochs in the source code, and the AI model training process ends once the number of epochs is reached. The y-axis is the logarithmic loss of the AI model training.

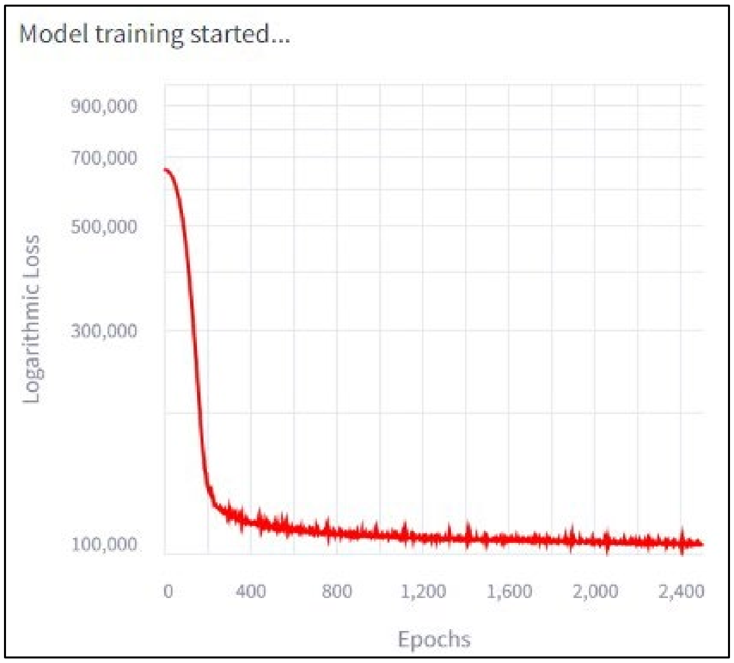


Figure 7. Streamlit GUI – Example of AI Model Training Progress

In addition to Figure 7, Figure 8 shows the AI model training process with a periodically updating graph in the command prompt application. In Figure 8, the x-axis is the percent difference (absolute value) between AI Model Training (i.e., Predicted Value) and Input Data (i.e., Expected Value), and the y-axis is the number of occurrences in a percent difference histogram bin. The bin size in Figure 8 is two (2) percent.

Figure 9 is another periodically updating graph in the command prompt application. The x-axis is the expected value while the y-axis is the predicted value.

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

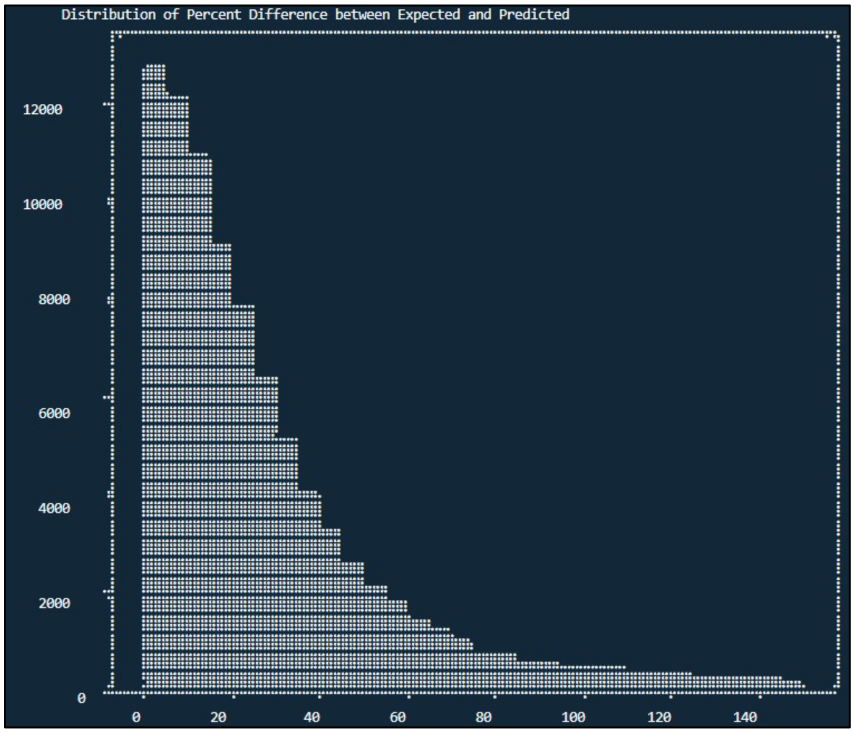
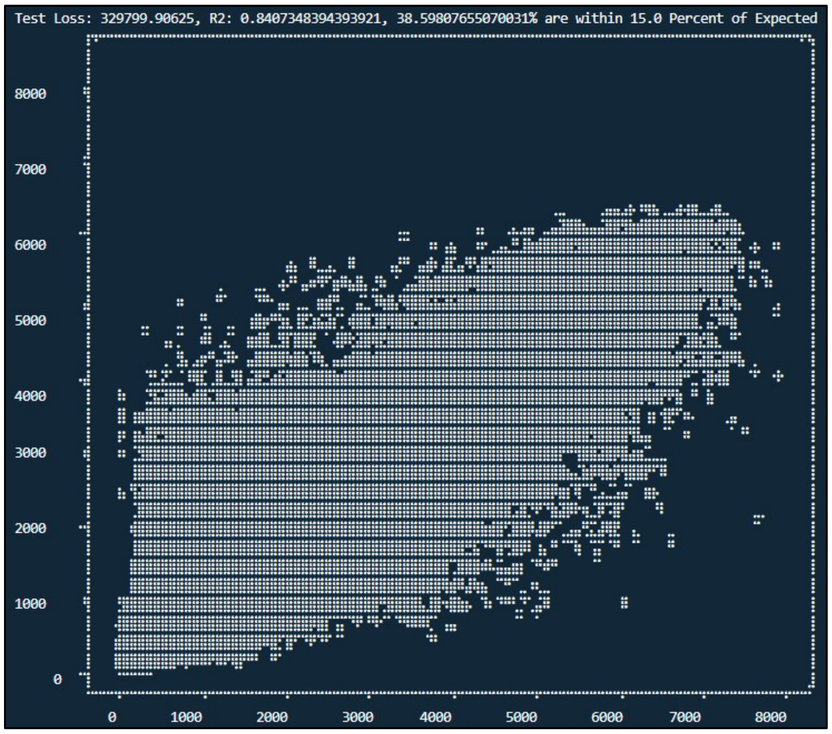


Figure 8. AI Model Training – Histogram of Percent Difference (Absolute Value) between AI Model Training (i.e., Predicted Value) and Input Data (i.e., Expected Value)



```
Test Loss: 329799.90625, R2: 0.8407348394393921, 38.59807655070031% are within 15.0 Percent of Expected
```

Figure 9. AI Model Training – AI Model Training (i.e., Predicted Value) versus Input Data (i.e., Expected Value)

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

After the AI model training completes, the following outputs are seen on the command prompt application. The logarithmic loss, tensor, test loss, and R-squared values provide a high-level summary of the AI model training. The AI model is saved to "..\models."

```
…

Epoch [2500/2500],

Logarithmic Loss: 104156.6484375,

tensor([[2079.9739],
          [ 271.9318],
          [4203.4741],
          ...,
          [ 647.0178],
          [3022.9729],
          [ 242.9163]])

tensor([[2846.],
          [ 394.],
          [5372.],
          ...,
          [2676.],
          [2528.],
          [ 103.]])

Test Loss: 382541.6875,

R2: 0.8152651190757751,

36.359431140945375% are within 15.0 Percent of Expected,

Model weights saved to ../models/model__20240329_194737

Model file saved to ../models/model__20240329_194737
```

{to add: explanation of loss, tensor, and R2 values if helpful}

{to add: explanation of AI training outputs}

{to add: outputs of "Use Model"}

4. Additional Notes

The geographic region that the algorithms use to train the model is determined by the NPMRDS data input into the code. Additional updates and improvements are planned in future releases and iterations.

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

Known Issues: None identified, this use case is still in development and future updates will be tested sufficiently before being released.

Associated Datasets: This use case incorporates NPMRDS, TMAS, U.S. Census, and other data sources to train the model discussed herein.

5. Version History and Retention

Status: This project is in active development phase.

Release Frequency: This project will be updated when there are stable developments. This will be approximately every month.

Retention: This project will likely remain publicly accessible indefinitely.

6. License

This project is licensed under the Creative Commons 1.0 Universal (CC0 1.0) License - see the License.md file for more details.

7. Contributing to the Code

Please read Contributing.md for details on our Code of Conduct, the process for submitting pull requests to us, and how contributions will be released.

8. Contact Information

Contact Name: Billy Chupp, William.Chupp@dot.gov

Contact Name: Eric Englin, Eric.Englin@dot.gov

Citing this code

Users may cite our code base and/or associated publications. Below is a sample citation for the code base:

ROADII Team. (2024). *ROADII README Template* (0.1) [Source code]. Provided by ITS JPO through GitHub.com. Accessed yyyy-mm-dd from https://doi.org/xxx.xxx/xxxx.

https://github.com/ITSJPO-TRIMS/R29-MobilityTrafficCounts

When you copy or adapt from this code, please include the original URL you copied the source code from and date of retrieval as a comment in your code. Additional information on how to cite can be found in the ITS CodeHub FAQ.

9. Acknowledgements

- Billy Chupp (Volpe), William.Chupp@dot.gov
  wchupp
- Eric Englin (Volpe), Eric.Englin@dot.gov
  eric-englin-volpe
- RJ Ritmuller (Volpe), Robert.Ritmuller@dot.gov
- Michael Barzach (Volpe), Michael.Barzach@dot.gov
- Jason Lu (Volpe), Jason.Lu@dot.gov

This project is funded by the U.S. DOT, ITS JPO under IAA HWE3A122. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the ITS JPO.

Languages

- Python 100.0%

About

This repository provides code for using ML methods to join national traffic datasets. One of these traffic data sets measure speed, and the other measures traffic volumes.