Traffic Prediction with ML and DL Algorithms

Group 14

Team Introduction

Jon Roof

MEng: Computer Security

VT Chemistry

Defense Industry

Kent Sullivan

MEng: Data Analytics & Artificial Intelligence

VT Mechanical Engineering

Operations Research

Aarya Shah

MEng: Software Development & Applications

VT Computer Science

Defense Industry

Seshadri Kakani

MEng: Data Analytics & Artificial Intelligence

VT Computer Science

Defense Industry

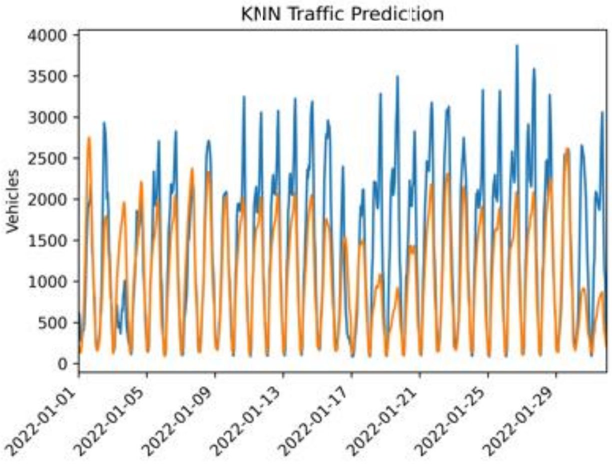
Problem Description

- Northern Virginia is next to the nation’s capital, Washington D.C. → massive amount of commuters
- Ability to predict traffic density and congestion is of vital importance
- Collected data from DOT FHWA from 2011-2022
- Analyzed traffic data at the off-ramp from I-66E onto Glebe Rd. due to its close proximity to the VT Northern Virginia Research Center

Approach 1: k-Nearest Neighbors (KNN)

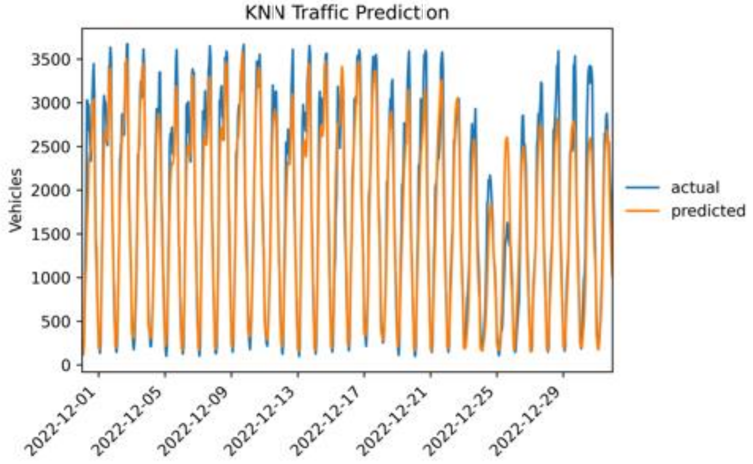
- Baseline approach - attempts to classify records based on the “n” data points closest to it
- RandomizedSearchCV with 10 fold cross validation, tuning the following hyperparameters (n_neighbors, leaf_size, power parameter, and weight)
- “Best” Parameters (6, 6, 1, distance) → Training RMSE of 411.92 / Validation RMSE of 459.33

Jan



...

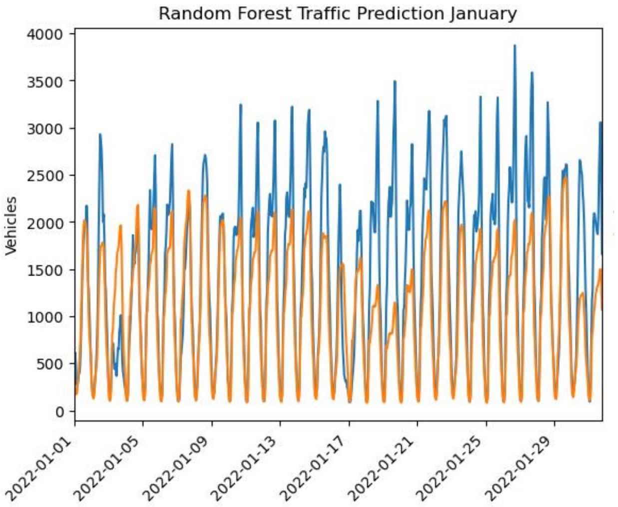
Dec



Approach 2: Random Forest

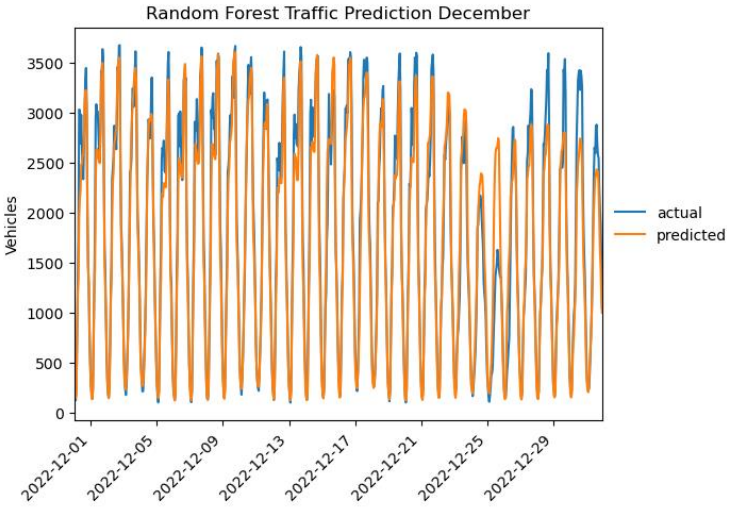
- Combines multiple decision trees to make a predictions
- Used RandomizedSearchCV(3 fold cross validation across 100 different combinations) to tune parameters. Tuned n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, and bootstrap

Jan



...

Dec

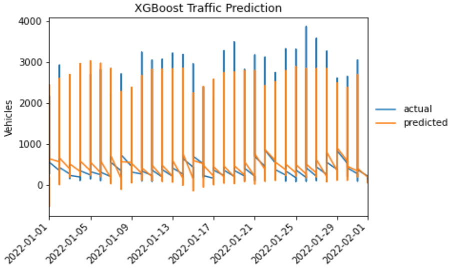


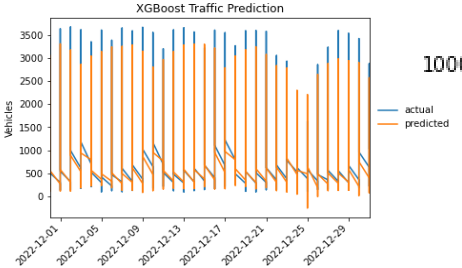
Approach 3: XGBoost

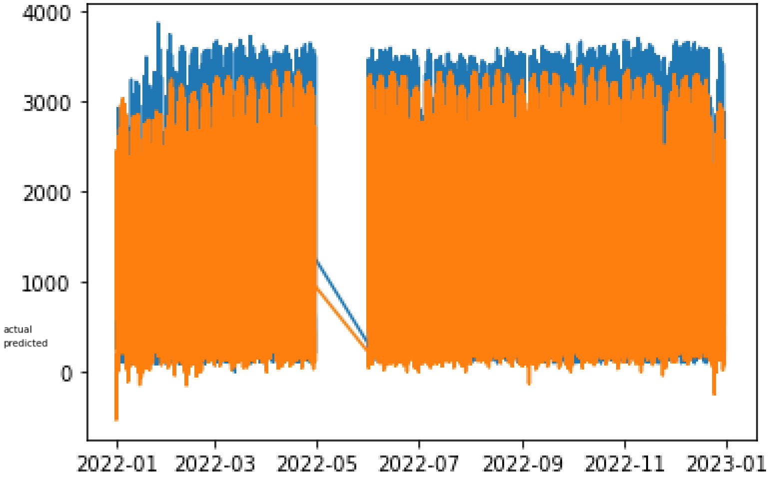
- Generate an ensemble of weak prediction models, and utilize boosting methods as well as optimizing a differentiable loss function. (Gradient Boosting)
- XGBoost uses parallel trees to perform Gradient Boosting
- Hyperparameters were tuned with Optuna

Validation RMSE: 353.000
Training RMSE: 94.465

Parameter	Value
Iterations	1820
Learning Rate	0.114
Max Depth	6
Lambda	0.014
Alpha	0.571

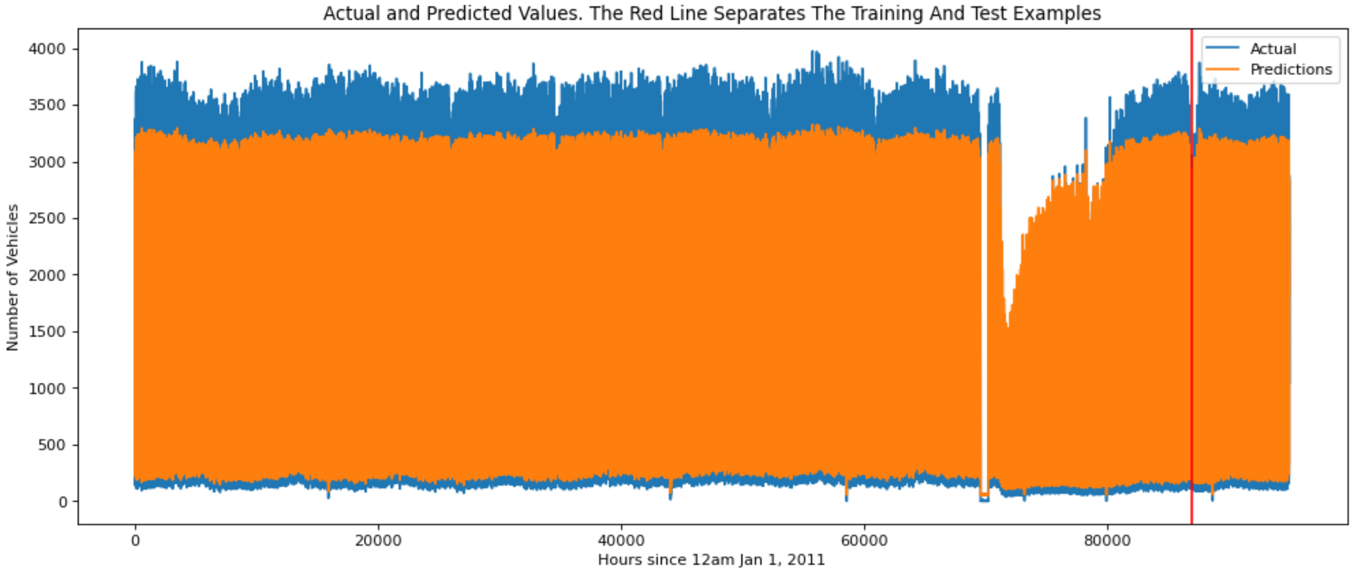






Approach 4: Recurrent Neural Network (RNN)

- Saves the output of the given data and feeds back to predict the output of the layer
- Train RMSE: 442.654 RMSE / Validation RMSE: 365.022 RMSE



Results Comparison

Model	Training Error (RMSE)	Validation Error (RMSE)
KNN	411.92	459.33
Random Forest	72.679	441.782
XGBoost	94.464	353.000
RNN	442.654	365.022

Lessons Learned

Objective

- Predict future congestion from past data
- Training with all previous data instead of randomly selected data from set
- As driving behavior & road patterns change, traffic congestion becomes less predictable
    - Stale Data
    - Covid

Hyperparameter Tuning

- Tuning hyperparameters should theoretically provide a lower error rate
- In our testing we had mixed results. After tuning the parameters for the Random Forest implementation, we got a higher error on the training data, but a lower error on the validation data

Location Variance

- Each model will be very context dependent, and will be most effective for a small set of locations
- I-66 E & Glebe Rd will not ever have the same patterns as Prices Fork & West Campus Drive
- Main Street North vs Main Street South

Future Work

- How much data is relevant? Is 2011 data useful for predicting 2023?
    - Could experiment with training set size
- Integrate weather data into feature set or other features
    - Adding the weather data could provide another feature to train on
- State of the Art
    - Combination of GNN(Graphical Neural Network) and RNN
        - GNN provides Spatial Dependency
        - RNN provides Temporal Dependency
    - Deep Traffic State Prediction(DeepTSP) Model
        - Uses Computer Vision Algorithms
        - Overlays maps of the city to find areas of traffic and predicts where high traffic areas will occur
        - Incorporates weather data