Project Intentions

This project aims to predict traffic flow in New York City (NYC) to assist in city planning and transportation systems. The predictions from the model will help in the following areas:

1. Traffic Management:

• Predicting traffic flow can aid in reducing congestion and improving route planning.

2. Resource Allocation:

• The model can help allocate vehicles and drivers more efficiently, optimizing resources.

3. Urban Planning:

• With accurate traffic forecasts, the city can make better decisions regarding infrastructure development.

Data Preprocessing

We performed several steps to clean and prepare the data:

• Replaced missing values (represented by “-”) with NaN (Not a Number).

• Dropped rows that were missing important values like Trips Per Day and Unique Vehicles.

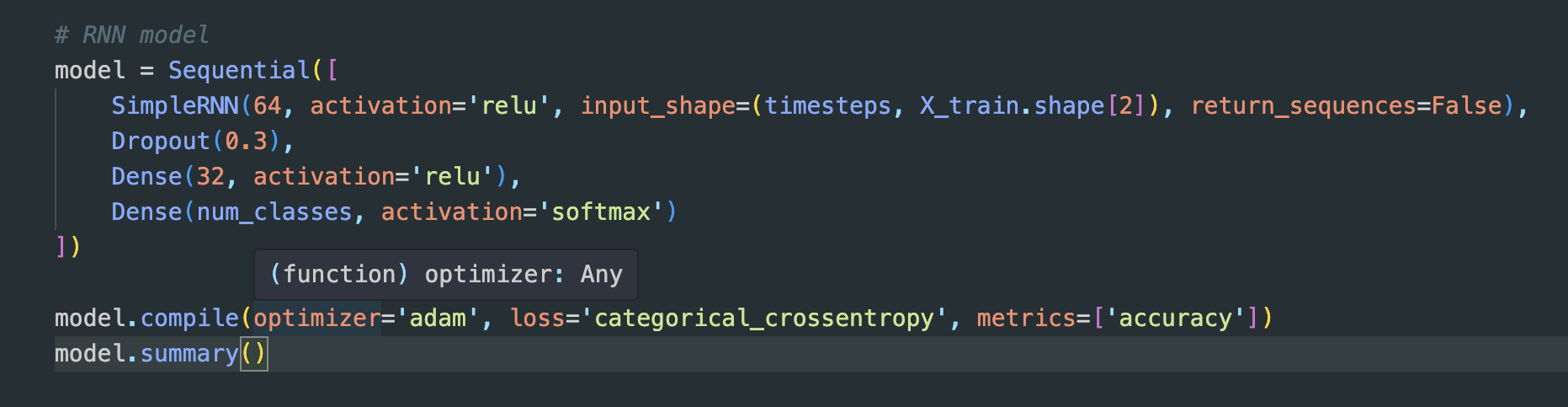
• Cleaned up numbers by removing commas and converting them to float format.

• We used LabelEncoder to change License Class and Month/Year into numerical values so that they could be used in the model.

**Recurrent Neural Network(RNNs)**

The project uses a SimpleRNN model to predict traffic flow. The key features used for prediction include Month/Year, Trips Per Day, Unique Vehicles, Unique Drivers, and License Class. The model utilizes historical data to identify patterns and make future predictions. The ultimate goal is to aid in better traffic management and urban planning.

RNNs model



This SimpleRNN model is designed for classification tasks involving sequential data. It starts with a SimpleRNN layer with 64 units, which processes the sequence and outputs a single result for the entire input. A dropout layer is added to reduce overfitting by turning off 30% of connections during training. Next, a dense layer with 32 units refines the data, followed by a final dense layer with softmax activation to output probabilities for each class. The model is compiled with the Adam optimizer for efficient training, categorical crossentropy as the loss function for multi-class classification, and accuracy as the evaluation metric. It works well for tasks like sequence classification or pattern recognition.

• Training Setup:

• The model was trained for 20 epochs with 20% of the data reserved for validation.

• Validation Performance:

• The model achieved similar results on the validation set, indicating good generalization and no overfitting.

• Test Performance:

• The model’s performance on unseen data showed:

• Test Loss (MSE): 0.0056

• Test MAE: 0.0110

This indicates that the model predicts traffic flow with minimal error, with the low MAE demonstrating that the model’s predictions are close to the real values.

Accuracy and Efficiency

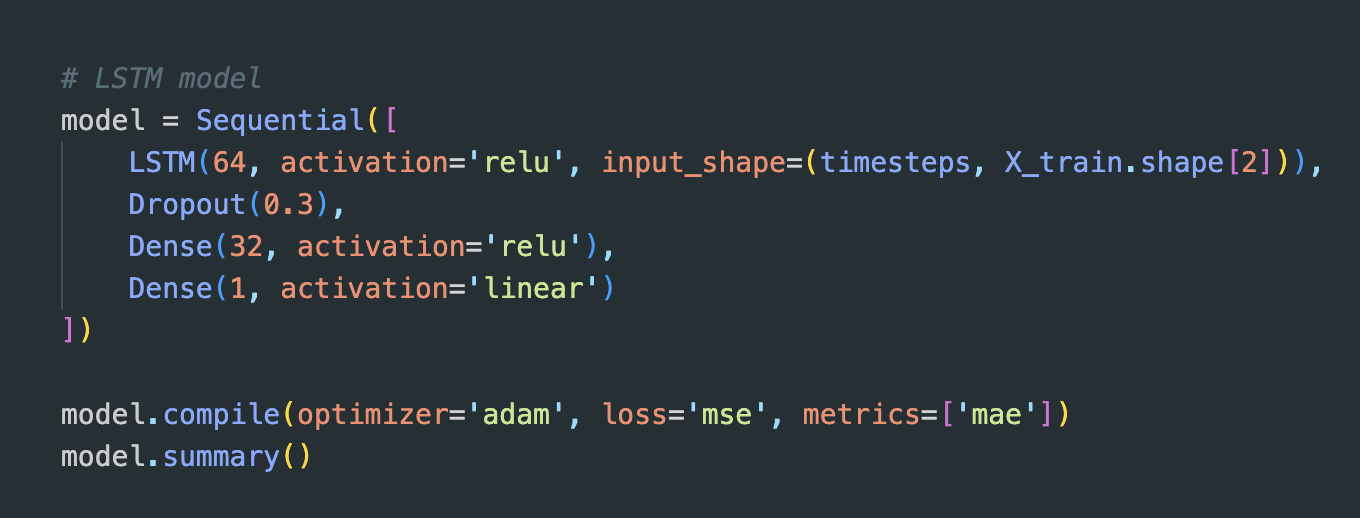
• The SimpleRNN model efficiently learned the sequential patterns in the data and stabilized quickly during training, indicating effective learning without overfitting.

• The validation results remained consistent with the training results, confirming the model’s ability to generalize well to new data.

**Long Short Term Memory(LSTMs)**

The LSTM model is a special kind of Recurrent Neural Network (RNN). It is very good at remembering information over time, which makes it perfect for predicting things like traffic flow, where past data affects future outcomes. Unlike regular RNNs, LSTMs can “remember” past data better, helping them make more accurate predictions.

LSTM model



This model predicts time series data, such as traffic flow. It starts with an LSTM layer with 64 units to process sequences, followed by a dropout layer that reduces overfitting by randomly deactivating 30% of connections during training. Two dense layers follow: one with 32 units using the ReLU activation for flexibility and another with 1 unit using a linear activation to produce the final prediction. The model is compiled with the Adam optimizer for efficient learning, Mean Squared Error (MSE) as the loss function to minimize prediction errors, and Mean Absolute Error (MAE) as a metric to evaluate performance.

LSTM Performance:

After testing the model on unseen data, it achieved the following results:

• Test Loss (MSE): 0.0056

• Test MAE: 0.0118

These results show that the LSTM model made very accurate predictions, with small errors.

The SimpleRNN model had a slightly higher error compared to the LSTM model, indicating it was not as accurate in predicting traffic flow.

Comparison of LSTM and SimpleRNN Models

| Model | Test Loss (MSE) | Test MAE |
| --- | --- | --- |
| LSTM | 0.0056 | 0.0118 |
| SimpleRNN | 0.0059 | 0.0123 |

Performance Comparison:

From the table above, the LSTM model performed better than the SimpleRNN model. It had lower values for both MSE and MAE, meaning it made more accurate predictions.

Efficiency:

• The LSTM model took more time to train because it has more complex layers. However, it is more accurate because it can remember long-term patterns.

• The SimpleRNN model was quicker to train, but it could not predict as well because it cannot capture long-term dependencies in the data.

Overfitting:

Both models showed good performance and did not overfit the training data. They both did well on both training and validation data, meaning they can generalize well to new data.

Conclusion

This study showed that the LSTM model is better than the SimpleRNN model for predicting traffic flow in NYC. The LSTM model produced lower error values (MSE and MAE) and was able to capture long-term dependencies in the traffic data, making it more accurate. While the SimpleRNN model is simpler and faster to train, it is not as good at predicting long-term patterns.

In the future, we can try using other models, such as GRU (Gated Recurrent Units), or add more features, like weather data, to improve the predictions even further. Hyperparameter tuning could also help to optimize both models.

Overall, the LSTM model is a great choice for predicting traffic flow because it is both accurate and reliable.

Let me know if you would like to add or change anything in this version!