Overview of Traffic Flow Dynamics in Chicago

# Introduction

Traffic congestion is a major challenge faced by many urban areas, including the city of Chicago. Understanding the patterns and dynamics of traffic flow is crucial for improving transportation infrastructure, optimizing traffic management, and reducing the environmental impact of vehicles. In this mini project, we aim to leverage the power of Convolutional Neural Networks (CNNs) to analyze the traffic flow of Chicago city.

In this mini project, we aim to leverage the power of CNNs to address the challenge of text classification. By applying a CNN-based model to a text classification problem, we seek to demonstrate the effectiveness of this deep learning approach and explore its advantages over traditional machine learning techniques.

# Dataset

We used the Traffic Volume Dataset from Kaggle

<https://www.kaggle.com/datasets/bobaaayoung/trafficvolumedatacsv>

Data Preprocessing:

* One-Hot Encoding.
* Label Encoding.
* Feature Scaling.

# Model Selection and Architecture

We selected a Convolutional Neural Network **(CNN)**

ModelArchitecture**:**

* **Input Layer**: The input layer corresponds to the features after scaling.
* **Dense Layers**: Two dense layers with 64 and 32 units respectively, each followed by ReLU activation.
* **Output Layer**: A dense layer with 3 units and softmax activation, corresponding to the three traffic conditions (low, medium, high).

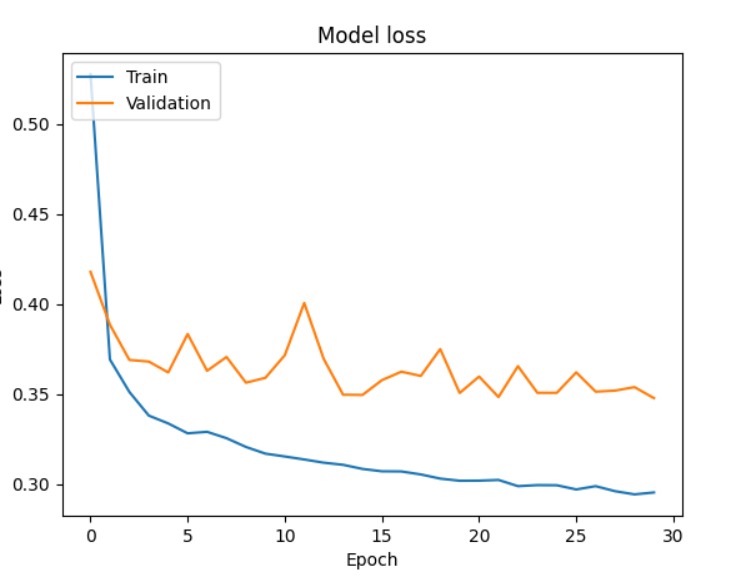
# Training and Validation

**Training Process**: The model was trained on the preprocessed dataset, with 70% for training, 30% for validation.

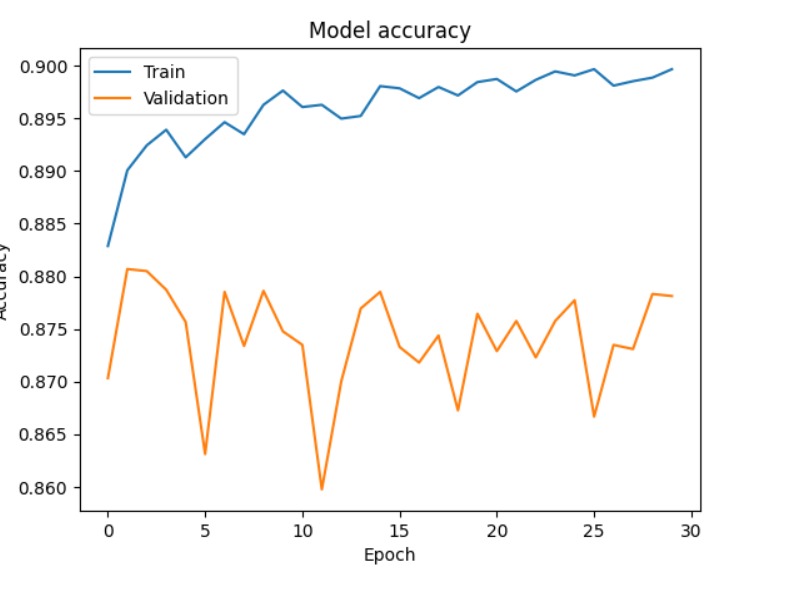
**Evaluation Metrics**: Accuracy: The final test accuracy for the CNN model was evaluated at the end of the training process.

Validation and Loss Curves: The accuracy and loss during training and validation were plotted to assess the model's performance over time.

# Model Loss and Accuracy

The following graph shows the model loss during training and validation phases:

The following graph shows the model accuracy during training and validation phases:



# Key Findings

* The CNN model is highly effective in predicting traffic conditions with a test accuracy of 87%.
* Data preprocessing, particularly feature scaling and encoding, played a crucial role in model performance.

|  |  |  |  |
| --- | --- | --- | --- |
|  | precision | recall | f1-score |
| 0 | 0.81 | 0.89 | 0.84 |
| 1 | 0.97 | 0.92 | 0.94 |
| 2 | 0.84 | 0.83 | 0.83 |

This output seems to performing reasonably well, but with some potential signs of a slight overfitting.

Precision: The model find a lot of the positives, but it can wrongly detect positives that aren’t actually positives.

Recall:The model shows a high recall means that there are positive cases in the data, though there can be negative cases identified as positive cases.

# Challenges

* Dataset Finds appropriate dataset based on our idea.
* Choosing the best model In the first, we struggle in many model until we find this model.
* Model Tuning Hyperparameter tuning involved selecting the optimal number of epochs, batch size, and learning rate to achieve the best validation performance.
* Data Preprocessing Managing categorical data and feature scaling were key challenges that were addressed through one-hot encoding, label encoding, and standardization.
* Linking MongoDB with our notebook.

# Future work

* Trying on more models.
* Try different methods to enhance the accuracy and prevent the overfitting.
* Build the model on different datasets to reach the perfect results.

# Contributions

|  |  |
| --- | --- |
| **Name** | **Task** |
| AlAnoud Basuliaman | Preprocessing and EDA |
| Sadeem AlMesned | Build the model |
| Nuha Aloqayli | Train and test the model |
| Rahaf AlShalahi | Ensemble (Random Forest) |