Milestone 3: Real-Time Traffic Analysis & Prediction

Team Name: Data Seekers
Sri Nihitha Mandadi
Siva Pavan Kumar Chava
Rakesh Kusa
Bhargava Reddy Erugula

Spring 2025
Course: Big Data
Dataset: Minnesota DOT Traffic Volume (2020–2024)

---

1 Introduction

Project Overview

The goal of this project is to develop a real-time traffic prediction system using historical traffic volume data from Minnesota's Department of Transportation (DOT). We explore and evaluate different machine learning and time-series models to forecast traffic flow and visualize volume intensity by day and hour.

2 Tools and Technologies

Component	Tools Used
Development	Google Colab (Python 3)
Libraries	Pandas, NumPy
ML Models	scikit-learn, XGBoost, CatBoost
Time Series	Prophet
Visualization	Matplotlib, Seaborn

3 Methodology

Workflow Diagram

```
Raw Dataset (CSV)
       ↓
Data Cleaning & Transformation (Pandas)
       ↓
Feature Engineering (hour, weekday, month)
       ↓
Train/Test Split
       ↓
Model Training
→ Linear Regression
```

```
→ Random Forest
→ XGBoost
→ CatBoost
→ Decision Tree
→ Prophet
       ↓
Prediction & RMSE Evaluation
       ↓
Classification (Low, Medium, High)
       ↓
Visualization (Forecast Plot + Heatmaps)
```

Implementation Steps

1. Restructured hourly columns using `pd.melt()`.
2. Created `datetime` column by combining date and hour.
3. Extracted hour, weekday, month as features.
4. Split data into training and testing sets.
5. Trained models: Linear, Tree-based, Prophet.
6. Evaluated RMSE and visualized results.
7. Classified volumes: Low (¡300), Medium (300–700), High (¿700).

4 Investigation Goals

- Predict hourly traffic volume using historical data.
- Compare model performance via RMSE.
- Visualize trends across weekdays and hours.
- Classify traffic volumes for better planning insights.

5 Results and Discussion

Model RMSE Comparison

Model	RMSE (Approx.)
Linear Regression	417.31918464462666
Decision Tree	363.7045059261617
Random Forest	363.7046963440941
XGBoost	363.6950130260243
CatBoost	363.472906937069
Prophet	80.57827892806492

Visualization and Classification

- Prophet produced trend, weekly, and seasonal plots.

- Seaborn heatmaps plotted predicted volumes by weekday and hour.
- Classification:
  - Low: #9be7ff
  - Medium: #ffc857
  - High: #ff5c5c

5Vs and Metrics

Metric	Value / Description
Volume	1.2 million+ rows
Variety	Hour, direction, station, timestamp
Velocity	Hourly readings
Veracity	Cleaned & normalized
Value	Useful for congestion planning
Latency	Low (except Prophet: ˜2min)
Compute Platform	Google Colab (Free)
File Size	˜500MB

6 Conclusion

This project successfully demonstrated real-time traffic volume prediction using various models. Prophet provided the most accurate forecasts. Tree-based models like CatBoost also performed well. Visual classification through heatmaps helped identify traffic intensity patterns across hours and weekdays.

7 Citations

- Minnesota DOT: `https://www.dot.state.mn.us/traffic/data.html`
- Prophet: `https://facebook.github.io/prophet/`
- XGBoost: `https://xgboost.readthedocs.io/`
- CatBoost: `https://catboost.ai/`
- scikit-learn: `https://scikit-learn.org/`
- Seaborn: `https://seaborn.pydata.org/`

8 GitHub Repository

`https://github.com/SriNihitha12/BigdataProject.git`

9 Appendix: Code Implementation Snapshots

```
# STEP 1: Mount Google Drive to access files
from google.colab import drive
drive.mount('/content/drive')
1. Linear Regression


# STEP 2: Install necessary Python libraries (only needed once in Colab)
!pip install pandas matplotlib scikit-learn

# STEP 3: Import required libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# STEP 4: Define the path to the dataset file in Google Drive
data_path = '/content/drive/MyDrive/TrafficData/Minnesota_TrafficData_2020-24.csv'

# STEP 5: Load the traffic dataset using pandas
df = pd.read_csv(data_path)

# STEP 6: Display first few rows (for checking structure)
df.head()

# STEP 7: Remove any fully empty rows
df.dropna(how='all', inplace=True)

# STEP 8: Optional - Print column names to verify hour columns exist
print(df.columns)
```

Figure 1: Linear Regression Code - Part 1

```
# 1. Import Random Forest Regressor from sklearn
from sklearn.ensemble import RandomForestRegressor

# 2. Create a Random Forest model with 100 trees
rf = RandomForestRegressor(n_estimators=100, random_state=42)

# 3. Train the model using training data
rf.fit(X_train, y_train)

# 4. Predict traffic volume using the test data
y_pred_rf = rf.predict(X_test)

# 5. Evaluate and print RMSE for Random Forest model
print("RMSE (Random Forest):", np.sqrt(mean_squared_error(y_test, y_pred_rf)))
RMSE (Random Forest):363.7046963440941

# 6. Install additional libraries (if needed)
!pip install pandas openpyxl
# 7. Create a pivot table to calculate average actual traffic volume by hour and weekday
pivot = hourly_df.pivot_table(values='volume', index='hour', columns='weekday', aggfunc='mean')

# 8. Set up the heatmap figure
plt.figure(figsize=(10, 6))
plt.title("Average Traffic Volume by Hour & Weekday")

# 9. Import seaborn and create the heatmap
import seaborn as sns
sns.heatmap(pivot, cmap='YlGnBu', annot=True, fmt='.0f')  # shows average volume per hour & weekday

# 10. Label the axes
plt.xlabel("Weekday (0 = Monday)")
plt.ylabel("Hour")

# 11. Display the plot
plt.show()
```

Figure 2: Random Forest Code - Part 2

```
# 1. Install XGBoost
!pip install xgboost

# 2. Import necessary libraries
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error
import numpy as np

# 3. Create and train the XGBoost model
xgb_model = XGBRegressor(objective='reg:squarederror', n_estimators=100, random_state=42)
xgb_model.fit(X_train, y_train)

# 4. Make predictions
y_pred_xgb = xgb_model.predict(X_test)

# 5. Evaluate the model using RMSE
xgb_mse = mean_squared_error(y_test, y_pred_xgb)
xgb_rmse = np.sqrt(xgb_mse)

# Print RMSE
print(" XGBoost RMSE:", xgb_rmse)
XGBoost RMSE: 363.6950130260243

# 6. Create a pivot table for heatmap visualization (Predicted volume)
predicted_pivot = pd.DataFrame(y_pred, columns=['Predicted'], index=X_test.index)
predicted_pivot['hour'] = X_test['hour']
predicted_pivot['weekday'] = X_test['weekday']
predicted_pivot = predicted_pivot.pivot_table(values='Predicted', index='hour', columns='weekday', aggfunc='mean')

#7 Plotting the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(predicted_pivot, cmap='coolwarm', annot=True, fmt='.0f', cbar_kws={'label': 'Predicted Volume'})
plt.title("Predicted Traffic Volume by Hour & Weekday")
plt.xlabel("Weekday (0 = Monday)")
plt.ylabel("Hour")
plt.tight_layout()
plt.show()
```

Figure 3: XGBoost Code - Part 3