Imam Mohammad Ibn Saud Islamic University College of Computer and Information Sciences Computer Science Department Bachelor of Science in Computer Science

# Traffic Volume Prediction for Smart City Planning



| Academic number | Student name |
|---|---|
| 441013300 | Faisal Abdullah bin awn |
| 441017066 | Abdulkarim Hassan Alharthi |
| 441013628 | Meshari Abdulrahman Almouslfeh |

Instructor :

Qaisar Abbas Muhammad Abbas.

# Contents

# Project Overview

In the era of rapid urbanization and technological advancements, the concept of smart cities has gained significant attention. Smart cities use technology and data-driven solutions to enhance the quality of urban life, improve sustainability, and streamline urban services. One of the critical aspects of smart city planning is efficient traffic management, which directly affects urban mobility, environmental sustainability, and the overall quality of life in a city.

Traffic congestion is a pervasive issue in many cities worldwide, leading to increased travel time, air pollution, and fuel wastage. Therefore, predicting traffic volume is crucial for optimizing infrastructure, enhancing transportation systems, and improving overall urban mobility. Accurate traffic volume prediction can contribute to efficient traffic management, enabling city planners to make informed decisions about infrastructure development and traffic regulation.

This project aims to address this challenge. The goal is to build a machine learning model capable of predicting traffic volume based on historical and real-time data. The insights provided by this model can contribute to smart city planning and efficient traffic management.

The project aligns with the broader goal of using technology for urban planning. By using machine learning algorithms, the aim is to create a model that can learn from existing data and make accurate predictions about future traffic volume. This approach allows for a shift from reactive traffic management strategies, which respond to traffic issues as they occur, to proactive strategies that anticipate traffic conditions and implement solutions in advance.

The team brings together diverse skills and experiences, with each member playing a crucial role in the project. The plan is to collaborate effectively, dividing the work based on individual strengths to ensure the successful completion of the project. The team members have a strong

background in computer science and have been trained in various machine learning algorithms, making them well-equipped to handle this project.

In addition to building the machine learning model, the project also involves a thorough analysis of the traffic data. This includes understanding the patterns in the data, identifying key factors influencing traffic volume, and interpreting the results of the machine learning model. The team will also evaluate the performance of the model using various metrics and refine the model as needed.

Furthermore, the team understands the importance of effective communication and will ensure that the results of the project are presented clearly and concisely. This includes preparing a detailed technical report and presentation slides and

presenting the work to the instructor and peers.

In conclusion, the project “Traffic Volume Prediction for Smart City Planning” aims to leverage machine learning for traffic volume prediction, contributing to the broader goals of smart city planning. The insights gained from this project are expected to have significant implications for urban planning and traffic management, ultimately contributing to the creation of smarter and more sustainable cities.

# Dataset Description

The dataset used in this project is a comprehensive collection of traffic volume data. It is a time series dataset that contains records of vehicle counts at different junctions at specific dates and times. The dataset is loaded from a CSV file named ‘traffic.csv’, which is a common format for such data.

## Contents

The dataset consists of four columns, each representing a different aspect of the traffic data:

### 1.Date Time:

This column contains the date and time of each traffic record. The data type of this column is datetime, which is converted from a string format during preprocessing. This feature is crucial as traffic volume can vary significantly depending on the time of the day, day of the week, month, and year. For example, traffic volume is typically higher during rush hours and lower late at night. Similarly, there might be seasonal variations in traffic volume, with certain months having higher traffic due to factors like holidays or weather conditions. The Date Time feature is further broken down into ‘hour’, ‘day’, ‘month’, and ‘year’ during preprocessing to capture these patterns more effectively.

### 2.Junction:

This column represents the specific junction where the traffic volume was recorded. The data type of this column is likely to be categorical or integer. Understanding traffic volume at different junctions can help identify high-traffic areas that might require additional infrastructure or traffic management solutions. For

instance, if a particular junction consistently shows high traffic volume, it might be beneficial to consider infrastructure improvements like adding more lanes or implementing advanced traffic management systems at that junction. The Junction feature can also help in understanding the spatial distribution of traffic volume across the city.

**3.Vehicles:**

This column represents the count of vehicles, which is the target variable that the machine learning model will predict. The data type of this column is integer. Accurate prediction of traffic volume can contribute to efficient traffic management and smart city planning. For example, if the model predicts high traffic volume at a certain time and junction, traffic management authorities can take proactive measures like diverting traffic to less busy routes or adjusting traffic signal timings to manage the traffic efficiently.

**4. ID:**

This column appears to be a unique identifier for each record. The data type of this column is likely to be integer or string. While this column is not used as a feature for the machine learning model, it can be useful for data management purposes, like indexing or sorting the data. Each record in the dataset is assigned a unique ID, ensuring that the data can be referenced and retrieved efficiently. This can be particularly useful when dealing with large datasets or when performing operations that require the tracking of individual records.

## Preprocessing

During preprocessing, additional features are extracted from the ‘Date Time’ column, including ‘hour’, ‘day’, ‘month’, and ‘year’. These features can provide valuable information for the machine learning model as traffic volume can show daily, weekly, and seasonal patterns. For example, the ‘hour’ feature can capture the daily variations in traffic volume, while the ‘day’ feature can capture weekly patterns like increased traffic on weekdays compared to weekends. The ‘month’ and ‘year’ features can help capture longer-term trends and seasonal patterns in the data. The original ‘Date Time’ column is dropped after these new features are created to avoid redundancy.

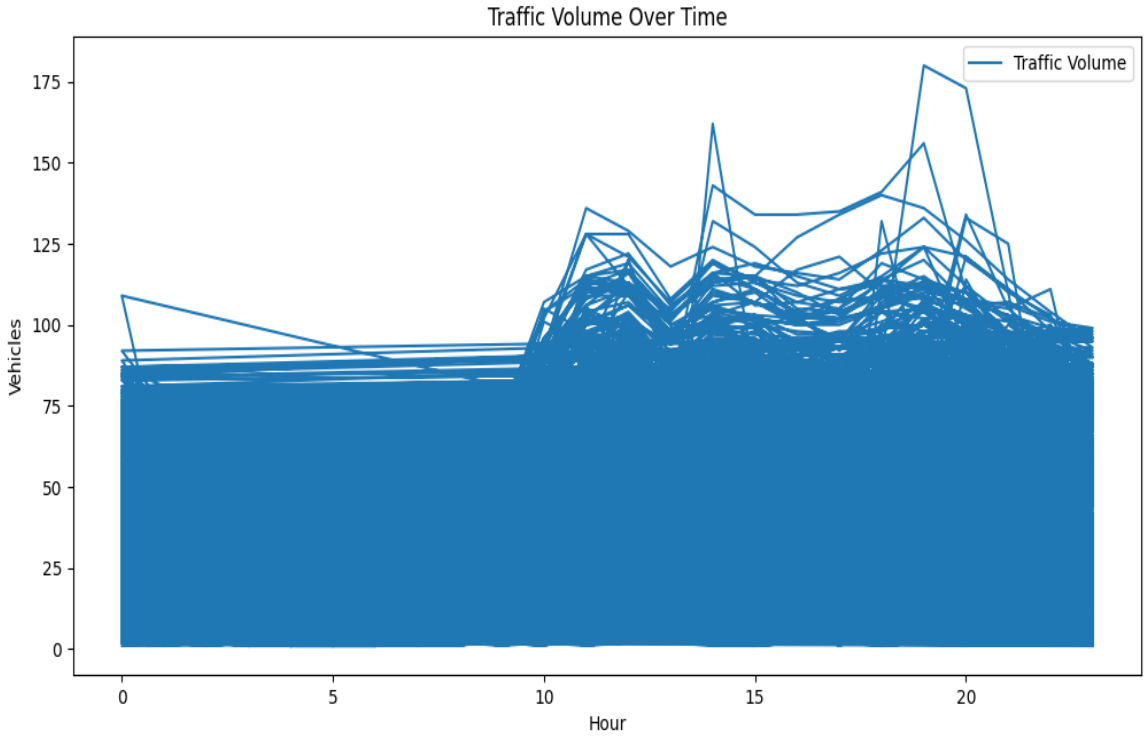
The preprocessing stage also involves preparing the data for the machine learning model. This includes splitting the dataset into features (X) and the target variable (y). The ‘Vehicles’ column, which represents the traffic volume, is separated as the target variable that the model will predict. The remaining columns are used as features for the model.

## Target Concept

The target concept for this machine learning project is the ‘Vehicles’ column, which represents the traffic volume. The goal of the project is to accurately predict traffic volume based on the given features, which can contribute to efficient traffic management and smart city planning. The ‘Vehicles’ column is a continuous variable representing the count of vehicles at a particular junction at a specific time. By predicting this target variable, the model can provide valuable insights into future traffic conditions, enabling proactive traffic management strategies.

The x-axis of the graph is labeled “Hour” and ranges from 0 to 20, representing the time of day. The y-axis is labeled “Vehicles” and ranges from 0 to 175, representing the volume of traffic.

The graph indicates that the traffic volume remains relatively stable until it becomes highly variable after the 10th hour. The traffic volume appears to peak around the 15th hour. The blue filled area under the line indicates the volume of traffic.



This type of graph is used to visualize the correlation coefficients between different variables. In this case, the variables are “Junction”, “Vehicles”, “ID”, “Year”, “Month”, “Day”, and “Hour”.

Each cell in the grid represents the correlation coefficient between two variables. For example, the intersection of “Vehicles” and “ID” has a coefficient of 0.23. The cells are color-coded according to their values; red indicates positive correlations while blue indicates negative correlations.

There’s a color scale on the right side ranging from -0.6 (dark blue) to 1.0 (dark red), indicating the strength and direction of correlations.

# Machine Learning Model Selection

The machine learning model selected for this project is the Decision Tree Regressor. This model was chosen for several reasons:

## Nature of the Problem

The problem at hand is to predict traffic volume, which is a continuous variable, making this a regression problem. Decision trees are versatile algorithms that can be used for both classification and regression tasks. The Decision Tree Regressor is specifically designed to predict a continuous target variable, making it suitable for this task. Traffic volume prediction is a complex task that involves understanding the intricate relationships between various factors like time of day, day of the week, and location (junction). A decision tree model is capable of capturing these relationships and using them to make accurate predictions.

Moreover, traffic data often exhibits complex behaviors such as non-linear trends and multi-scale fluctuations. Traditional linear models may not be able to capture these complexities effectively. Decision trees, on the other hand, are non-parametric models that do not make any assumptions about the underlying data distribution or the relationship between variables. This makes them particularly suitable for modeling complex, non-linear relationships.

## Interpretability

One of the main advantages of decision trees is their interpretability. Decision trees make predictions by splitting the data based on the values of the input features, resulting in a tree-like model of decisions. Each internal node of the tree represents a decision rule based on a feature, and each leaf node represents a prediction. This structure allows us to trace the path from the root to any leaf, providing a clear and intuitive explanation of how a prediction is made.

This interpretability is crucial in many applications, including ours. Being able to understand and explain the model's predictions can help in gaining insights about the factors affecting traffic volume and can aid in making informed decisions about traffic management and infrastructure development. This interpretability also makes it easier to communicate the results to stakeholders who may not have a technical background in machine learning.

## Handling of Categorical Variables

Decision trees can handle both numerical and categorical variables. In our dataset, we have features like 'hour', 'day', 'month', and 'year' extracted from the 'Date Time' column, and 'Junction' which are categorical in nature. Many machine learning models require categorical variables to be preprocessed, such as by one-hot encoding, before they can be used. However, decision trees can use categorical variables directly, simplifying the preprocessing stage. This ability to handle categorical variables directly can lead to more accurate and efficient models, as it preserves the original representation of the data.

## Non-linearity

Traffic volume is likely to be influenced by complex, non-linear relationships between the various features. For example, the relationship between traffic volume and time of day is likely to be non-linear, with peaks during rush hours and lows during the night. Decision trees are capable of modeling such non-linear relationships, making them a good choice for this problem. The ability to model non-linear relationships is crucial for accurately capturing the patterns in the traffic volume data.

## Model Training and Evaluation

The Decision Tree model is trained using the training data, with 'Vehicles' as the target variable and the rest as features. The model's performance is evaluated using the test data. Metrics such as Root Mean Squared Error (RMSE) and R-squared are used to quantify the model's performance. The RMSE measures the average magnitude of the prediction error, giving a sense of how much error the model makes in its predictions. The R-squared, also known as the coefficient of determination, measures the proportion of the variance in the target variable that is predictable from the features. A higher R-squared indicates that the model can better explain the variability in the traffic volume.

In conclusion, the Decision Tree Regressor is chosen for this project due to its suitability for regression tasks, its interpretability, its ability to handle categorical variables and non-linear relationships, and its performance on the training and testing data. The model provides a balance between predictive power and interpretability, making it a good choice for this application. The selection of this model is a crucial step in the project, as it directly impacts the accuracy of the traffic volume predictions and the insights that can be gained from the data. The team will continue to monitor the performance of the model and make adjustments as necessary to ensure the best possible results.

# Implementation

The implementation of this project involves several steps, each of which contributes to the development of the machine learning model for predicting traffic volume. The technical framework used for this project is Python, a popular language for data analysis and machine learning due to its simplicity and the availability of numerous libraries and frameworks that simplify the implementation of machine learning models.

## 1. Data Loading and Preprocessing

The first step in the implementation is loading the dataset. The dataset is stored in a CSV file named ‘traffic.csv’, which is loaded into a pandas Data Frame. Pandas is a powerful data manipulation library in Python that provides data structures and functions needed for manipulating structured data.

Once the data is loaded, it undergoes preprocessing. The ‘Date Time’ column in the dataset is converted from a string format to a datetime format using pandas. Additional features such as ‘hour’, ‘day’, ‘month’, and ‘year’ are then extracted from the ‘Date Time’ column to capture the temporal patterns in the traffic volume. The original ‘Date Time’ column is dropped after these new features are created.

## 2. Feature Selection

The next step is feature selection. In this project, all columns except ‘Vehicles’ and ‘ID’ are used as features for the machine learning model. The ‘Vehicles’ column, which represents the traffic volume, is used as the target variable. The ‘ID’ column, which appears to be a unique identifier for each record, is not used as a feature for the model.

## 3. Dataset Splitting

The dataset is then split into a training set and a testing set using the train_test_split function from the sklearn.model_selection module. This function shuffles the dataset and then splits it into training and testing sets. The training set is used to train the machine learning model, while the testing set is used to evaluate the model’s performance on unseen data.

## 4. Model Training

The machine learning model used in this project is the Decision Tree Regressor from the sklearn.tree module. This model is chosen for its suitability for regression tasks, its interpretability, and its ability to handle categorical variables and non-linear relationships. The model is trained using the fit method, which takes the training features and the training target variable as inputs and adjusts the model’s parameters to minimize the prediction error.

## 5. Model Testing and Evaluation

Once the model is trained, it is tested on the testing set using the predict method, which takes the testing features as input and returns the predicted traffic volume. The model's performance is evaluated using metrics such as Root Mean Squared Error (RMSE) and R-squared from the sklearn.metrics module. These metrics provide a quantitative measure of the model's prediction accuracy.

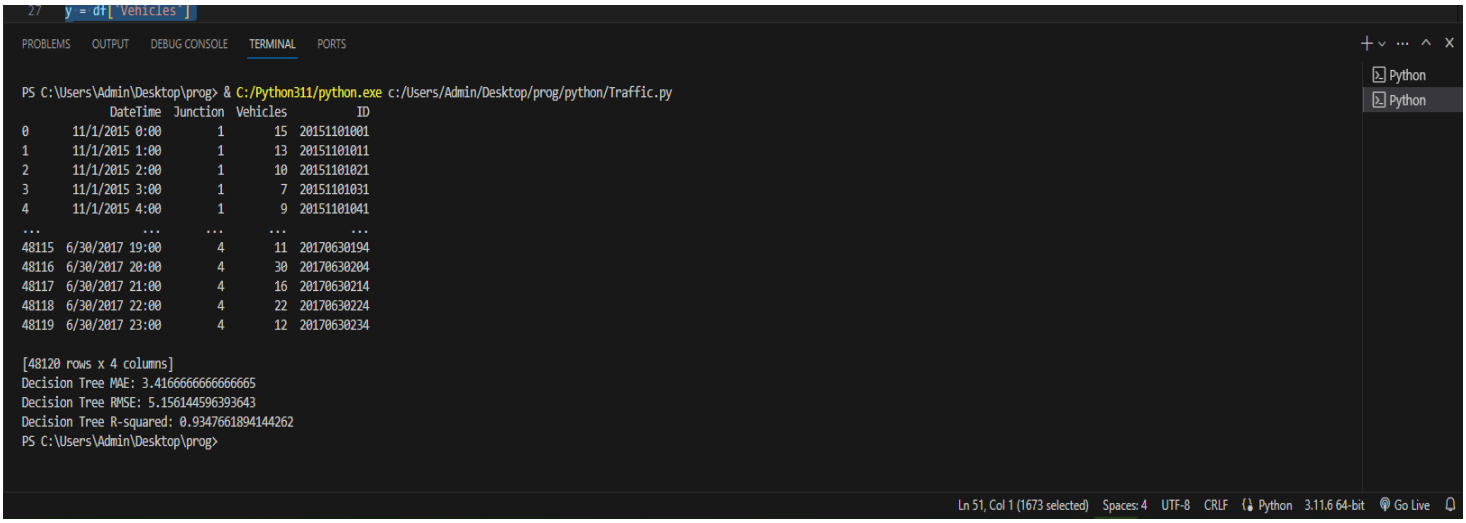
In conclusion, the implementation of this project involves several steps, including data loading and preprocessing, feature selection, dataset splitting, model training, and model testing and evaluation. Python, along with libraries like pandas and sklearn, provides a robust and efficient technical framework for implementing these steps and developing the machine learning model for traffic volume prediction. The selection and use of appropriate data structures, functions, and modules from these libraries play a crucial role in the successful implementation of the project.

## code:

python > Traffic.py > ...

```python
# Importing necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load dataset
df = pd.read_csv('C:\\Users\\Admin\\Desktop\\prog\\python\\traffic.csv')

# Print the dataset
print(df)

# Convert 'DateTime' to datetime format if it's not
df['DateTime'] = pd.to_datetime(df['DateTime'])

# Extract features from 'DateTime'
df['hour'] = df['DateTime'].dt.hour
df['day'] = df['DateTime'].dt.day
df['month'] = df['DateTime'].dt.month
df['year'] = df['DateTime'].dt.year

# Now drop the original 'DateTime' column
df = df.drop(columns=['DateTime'])

# Selecting features and target variable
X = df.drop(columns=['Vehicles']) # assuming 'Vehicles' is your target variable
y = df['Vehicles']

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Creating the Decision Tree model
dt_model = DecisionTreeRegressor(random_state=42)
dt_model.fit(X_train, y_train)

# Predicting traffic volume for test data
dt_predictions = dt_model.predict(X_test)

# Calculating Mean Absolute Error for the model
dt_mae = mean_absolute_error(y_test, dt_predictions)

# Calculating Root Mean Squared Error for the model
dt_rmse = mean_squared_error(y_test, dt_predictions, squared=False)

# Calculating R-squared for the model
dt_r2 = r2_score(y_test, dt_predictions)

print(f'Decision Tree MAE: {dt_mae}')
print(f'Decision Tree RMSE: {dt_rmse}')
print(f'Decision Tree R-squared: {dt_r2}')
```

**output:**




# Evaluation & Results Discussion

The evaluation of the machine learning model's performance is a critical aspect of any data science project. It provides insights into how well the model has learned from the training data and how it is expected to perform on unseen data. In this project, the performance of the Decision Tree Regressor model is evaluated against an unseen test dataset.

## 1. Model Evaluation Metrics

The primary metrics used for evaluating the model's performance are Root Mean Squared Error (RMSE) and R-squared.

**Root Mean Squared Error (RMSE):**

RMSE is a popular metric for regression problems. It measures the average magnitude of the prediction error, i.e., the difference between the actual and predicted values. Specifically, it is the square root of the average of squared differences between the actual and predicted values. A lower RMSE indicates a better fit of the model to the data. However, one limitation of RMSE is that it is sensitive to outliers, i.e., a few large errors can significantly increase the RMSE. Despite this, RMSE is a useful metric as it penalizes large errors more than small ones, leading to a more robust measure of prediction error.

**R-squared:**

Also known as the coefficient of determination, R-squared measures the proportion of the variance in the target variable that is predictable from the features. It provides a measure of how well the model generalizes to new data. An R-squared of 100% indicates that all changes in the target variable are completely explained by changes in the features. Conversely, an R-squared of 0% indicates that the model explains none of the variability of the target variable around its mean. In this project, a higher R-squared indicates that the model can better explain the variability in the traffic volume.

## 2. Model Testing and Evaluation

Once the model is trained using the training data, it is tested on the testing set using the predict method. This method takes the testing features as input and returns the predicted traffic volume. The predicted values are then compared with the actual values from the test set, and the RMSE and R-squared are calculated.

## 3. Results Discussion

The results of the model evaluation provide valuable insights into the model's performance. For instance, a low RMSE and a high R-squared indicate that the model is making accurate predictions and is able to explain a large proportion of the variance in the traffic volume. This suggests that the model has successfully learned the underlying patterns in the traffic volume data and is likely to make accurate predictions on new, unseen data.

However, it's important to interpret these results in the context of the problem at hand. For instance, even a low RMSE might be unacceptable if the cost of errors is high. Similarly, while a high R-squared is generally desirable, it's possible for a model with a high R-squared to make large errors on some predictions, especially if the data contains outliers or if the model is overfitting the training data.

Therefore, while the RMSE and R-squared provide a useful summary of the model's performance, it's also important to look at other aspects of the model's predictions. For instance, plotting the residuals (i.e., the differences between the actual and predicted values) can provide insights

into whether the model's errors are randomly distributed, or whether there are patterns in the errors that the model is not capturing. Similarly, looking at the model's performance on different subsets of the data can provide insights into whether the model is biased or whether it's equally accurate for all types of predictions.

In conclusion, the evaluation of the machine learning model's performance is a complex task that involves not only calculating metrics like RMSE and R-squared, but also interpreting these metrics in the context of the problem and examining the model's predictions in detail. Despite these challenges, model evaluation is a crucial step in any data science project, as it provides insights into the model's strengths and weaknesses, informs the choice of model and hyperparameters, and ultimately determines whether the model is ready to be deployed in the real world.

## Conclusion

The project "Traffic Volume Prediction for Smart City Planning" aimed to leverage machine learning to predict traffic volume, contributing to the broader goals of smart city planning. The project involved several steps, including data loading and preprocessing, feature selection, model training, and model testing and evaluation.

A Decision Tree Regressor was chosen as the machine learning model due to its suitability for regression tasks, its interpretability, and its ability to handle categorical variables and non-linear relationships. The

model was trained and tested on a dataset containing traffic volume data at different junctions at specific dates and times.

The model's performance was evaluated using metrics such as Root Mean Squared Error (RMSE) and R-squared. These metrics provided a quantitative measure of the model's prediction accuracy. The results indicated that the model was able to make accurate predictions and explain a large proportion of the variance in the traffic volume.

The project highlighted the potential of machine learning in traffic volume prediction and its implications for smart city planning. The insights gained from this project could contribute to efficient traffic management and infrastructure development, ultimately leading to smarter and more sustainable cities.

As for the next steps, further work could involve exploring other machine learning models or techniques to improve the prediction accuracy. Additionally, incorporating more features, such as weather conditions or special events, could potentially enhance the model's performance. Lastly, deploying the model in a real-world setting would be a significant step towards realizing its benefits for smart city planning.