

# Laboratory Practical - 3 ML Mini Project Report

---

Group Members :-  
Tanay Zope 19CO074  
Saba Syed 19CO065  
Sagnik Roy 18CO061

# Introduction

- Daily traffic is a nuisance in our lives but it would be better if we knew how much traffic is present at a time.
- In this challenge we perform a complete analysis to understand the parameters that affect traffic of a city.
- In particular, we apply the tools of machine learning to predict the traffic of a city at a certain time.

# Problem Statement

- Develop a Machine Learning model to forecast twelve-hours of traffic flow in a U.S. metropolis using parameters like time,direction,coordinates etc.
- Dataset Link:  
<https://www.kaggle.com/competitions/tabular-playground-series-mar-2022/data>

# Objective

- To build a model for prediction.
- To analyze its performance on Chicago Traffic Tracker - Historical Congestion Estimates dataset. .
- To use different ML concepts to optimize the model's performance.

# Scope

- There may be certain unexpected conditions like car wrecks, road damages but for everyday purposes the traffic is based on certain parameters like time etc.
- Predicting the congestion of traffic in the city.

# System Requirements:

Operating System : 64 bit Linux or its derivatives / Windows.

Python Programming Language >=

3.6

Jupyter Notebook >=

4.1.11

Pip >=

3.0.2

Numpy >=

1.18.2

Pandas >= 1.0.3

Matplotlib >=

1.0.1

Seaborn >= 1.8.5

Scikit Learn >=

1.1.3

---

# Theory



## Scikit Learn

- Scikit-learn is a free machine learning library for Python.
- It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

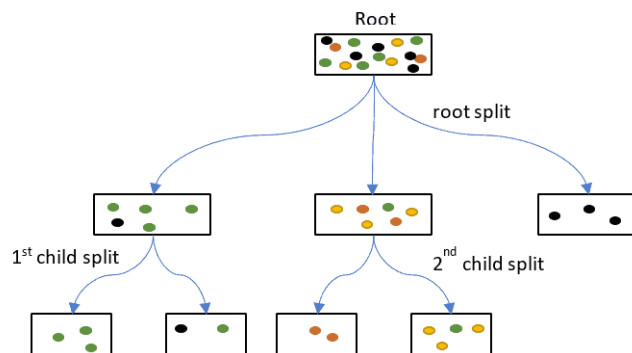
# CatBoost Regression

CatBoost builds upon the theory of decision trees and gradient boosting. The main idea of boosting is to sequentially combine many weak models (a model performing slightly better than random chance) and thus through greedy search create a strong competitive predictive model.

Gradient boosting fits the decision trees sequentially, the fitted trees will learn from the mistakes of former trees and hence reduce the errors. This process of adding a new function to existing ones is continued until the selected loss function is no longer minimized.

One of CatBoost's core edges is its ability to integrate a variety of different data types, such as images, audio, or text features into one framework. But CatBoost also offers an idiosyncratic way of handling categorical data, requiring a minimum of categorical feature transformation, opposed to the majority of other machine learning algorithms, that cannot handle non-numeric values.

It is not always as clear when to use random forests vs when to use gradient boosting. There are several sophisticated gradient boosting libraries out there (lightgbm, xgboost and catboost) that will probably outperform random forests for most types of problems.



## Benefits of the CatBoost Machine Learning Algorithm

1. One of the most frustrating parts of data science is, well, the data. That data can include a variety of forms, but perhaps the main one that causes all the problems is that of the categorical feature type. This type can also be described as a string, object, or categorical dtype more specifically.
2. It has effective usage with default parameters thereby reducing the time needed for parameter tuning.
3. CatBoost uses ordered target encoding, which essentially allows you to keep the feature/column in its original state, allowing you to collaborate with ml engineers and software engineers more easily.
4. The last benefit of CatBoost is a product of the first benefit. Because you do not have a sparse dataframe, the processing of your model that uses plenty amount of categorical features is much faster than if you used other algorithms like XGBoost or Random Forest.



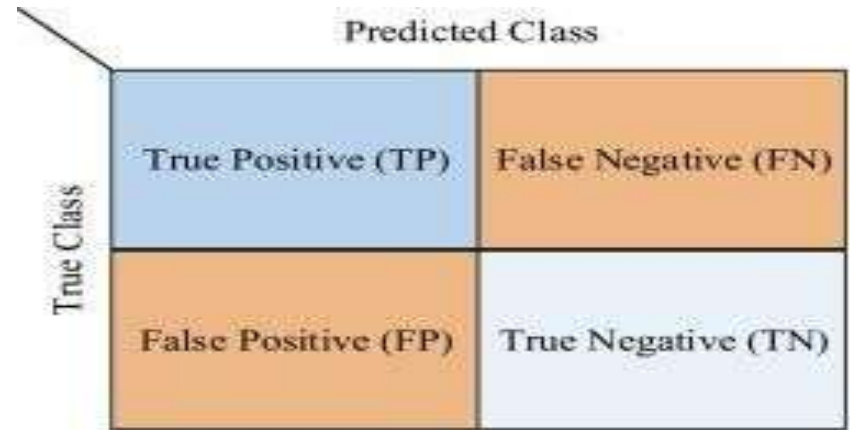
# Metrics for Accuracy Measurement

Evaluating the model accuracy regression model in Python.

- **MAE (Mean absolute error)** represents the difference by averaged the absolute difference over the data set.
- **MSE (Mean Squared Error)** represents the difference between the
  - original and predicted values extracted by squared the average difference over the data set.
  - **R-squared (Coefficient of determination)** represents the coefficient of how well the values fit compared to the original values. The value from 0 to 1 interpreted as percentages. The higher the value is, the better the model is.

The confusion matrix consists of four basic characteristics (numbers) that are used to define the measurement metrics of the classifier. These four numbers are:

1. TP (True Positive): TP represents the number of patients who have been properly classified to have malignant nodes, meaning they have the disease.
2. TN (True Negative): TN represents the number of correctly classified patients who are healthy.
3. FP (False Positive): FP represents the number of misclassified patients with the disease but actually they are healthy. FP is also known as a *Type I error*.
4. FN (False Negative): FN represents the number of patients misclassified as healthy but actually they are suffering from the disease. FN is also known as a *Type II error*.

A diagram of a confusion matrix. It is a 2x2 grid. The columns are labeled 'Predicted Class' at the top. The rows are labeled 'True Class' on the left. The four cells are: Top-Left (blue) is 'True Positive (TP)', Top-Right (orange) is 'False Negative (FN)', Bottom-Left (orange) is 'False Positive (FP)', and Bottom-Right (light blue) is 'True Negative (TN)'.

True Class	Predicted Class	
	True Positive (TP)	False Negative (FN)
False Positive (FP)		
True Negative (TN)		

# Modules Used

- **Numpy** : NumPy is a Python library used for working with arrays.
- **Pandas** : For Data Processing in the Notebook.
- **Matplotlib** :For Plotting charts and graphs for better visualization.
- **Seaborn** : For lightweight, powerful visualization of data.
- **Scikit learn**: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

# Algorithm

1. Import Necessary libraries and datasets.
2. Take a statistical look at the dataset.
3. Handle missing values in the dataset.
4. Plot various graphs for gaining insights from the data visualization process.
5. Do feature selection.
6. Split dataset into training and validation set and perform Standard Scaling.
7. Build the CatBoost Regressor Model.
8. Compile and train the model using the training dataset.
9. Get predictions on Validation dataset and display model evaluation results.
10. Use it to predict on the unseen Testing dataset.

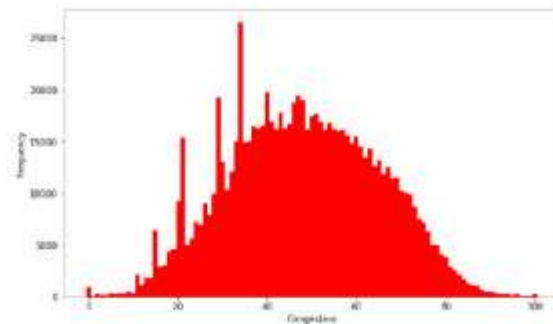
# Results

The model performed well with an accuracy of 77.34%.

Mean absolute error: 5.34

Mean squared error: 56.98

Coefficient of determination/Accuracy: 77.34%



# Conclusion

Hence, a machine learning model using CatBoost Regressor has been build, that predicts the traffic of a city at given point of time.