ARRIVAL-TIME PREDICTION USING MACHINE LEARNING ALGORITHMS

INTRODUCTION:

In the ever-evolving landscape of transportation and logistics, one of the most critical aspects is the accurate prediction of arrival times. Whether it's for commuters relying on public transportation, delivery companies optimizing routes, or ride-sharing services providing estimated drop-off times, precision in arrival time predictions can greatly enhance user experience and operational efficiency. Historically, these predictions were based on static schedules and basic heuristics. However, with the advent of machine learning algorithms and the availability of extensive historical data and real-time traffic information, there exists a remarkable opportunity to significantly improve arrival time prediction accuracy. This integration of machine learning techniques allows us to harness the power of data-driven insights, enabling more reliable and timely arrivals, thus reshaping how we plan, navigate, and optimize transportation services.

MACHINE LEARNING MODEL:

Machine learning is a technique for computers to learn how to do certain tasks without being explicitly programmed. It allows the machine to learn and make use of complex patterns of the processed data that cannot or difficult to be programmed by the human in a short period of time. There are more advanced machine learning methods such as ANN and SVM being proposed in recent years as machine learning has gained popularity

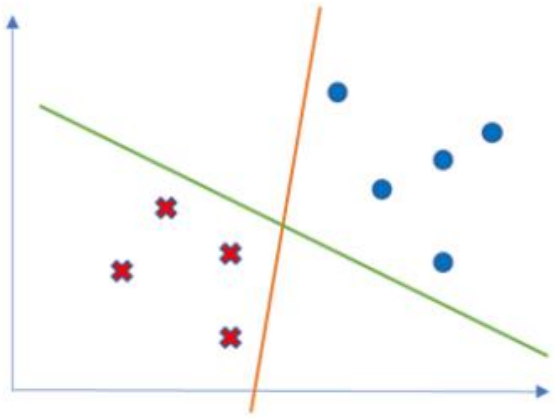
METHODS:

- **Support Vector Machine (SVM)**
- **Support Vector Regression (SVR)**

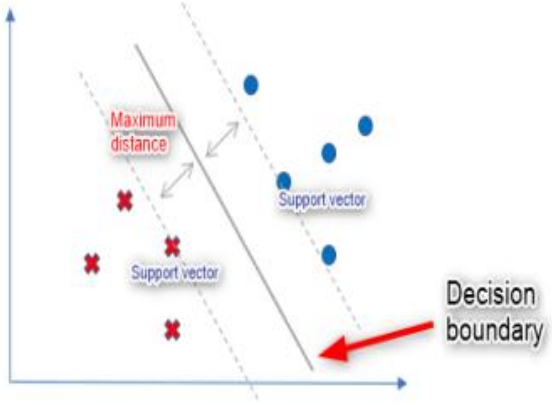
Support Vector Machine (SVM):

SVM constructs a hyperplane that separates the input data or the training data into classes linearly. This hyperplane acts as the decision boundary, and this boundary has the maximum distance between the linearly separable classes.

The distance of a class to the boundary is often described as margin. If the classes are not linearly separable, the input data can be projected to a higher dimensional space using kernel function in order to obtain a hyperplane or decision boundary that separates the classes.



CLASS SEPERATION



DECISION BOUNDARY AND MARGIN

Among the strengths of SVM are:

1. Able to learn from a small sample size

2. Avoid local minimum and

3. Good generalization capability to unseen data.

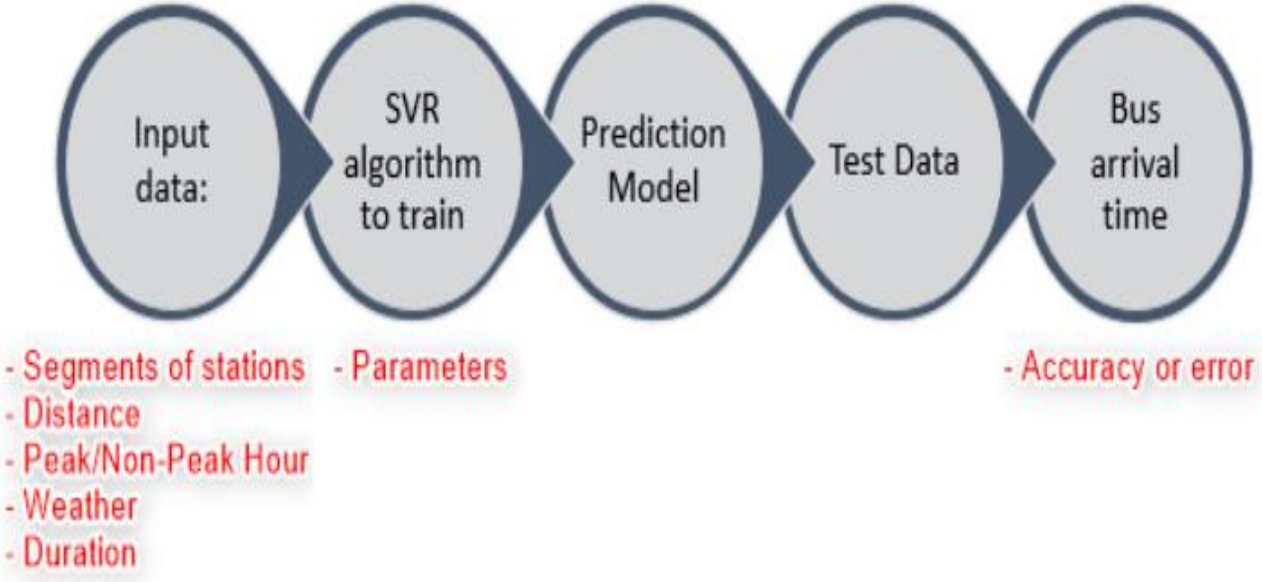
Support Vector Regression (SVR):

- SVR deals with the regression. It follows almost the same concept of SVM but a bit different from it. SVR is used as a regression algorithm while SVM as a classification algorithm. SVR is worked with continuous values instead of classification which is SVM.
- SVR defines the hyperplane as the line that would assist us to predict the continuous value or target value. In summary, it can be said that when SVM is used for the regression problems, it is called as SVR.

- SVM is not a probabilistic model and thus does not assume any randomness. It simply draws a simple line (hyperplane in higher dimensions) to separate the data points into two parts.
- However, the issue is that sometimes the classifier (the separating hyperplane) cannot be defined linearly. It is not always expected as a straight line, but it should rather be a wavy curve or surface. In this case, SVR is used.
- By applying a regression problem, linear regression could be described as an attempt to draw a hyperplane in higher dimensions that minimizes the error(or the loss function).
- Therefore, if we select different loss functions, the regression line changes. In this way, SVR gives the most appropriate results of estimation. It gives a wavy curved hyperplane rather than a straight line (that is given by SVM).

PROCEDURE:

There are 5 input features that consist of segments of bus stations, duration of the journey from a station to the next station, the distance, time segment of whether the bus operates during peak hour or non-peak hour, and lastly the weather data. All the input features are used in the training of the model. Generally, a part of the input dataset is used as a train set and trained by using the SVR algorithm to get the bus arrival time prediction model, while the other part of the dataset is used as a test set to check on the accuracy or error of the prediction.



STEPS:

- **DATA PREPROCESSING**
- **FEATURE SCALING**
- **TRAINING OF MODEL**
- **PERFORMANCE EVALUATION**

DATA PREPROCESSING:

In order to get the input features needed for model training, the data was preprocessed. First, the dataset was cleaned up in Python, whereby all non-ASCII text and all rows that contain missing data were removed.

FEATURE SCALING:

Before feeding the data into the model, it needs to be scaled as the range of every input feature can be very different. For example, a duration range can be tens to thousands of seconds while peak hour and non-peak hours are represented by 1 and 0, respectively. Huge discrepancies between the input features can cause a fitting issue as a feature with a huge number may get different weights than a feature with a small number. Min-max scaling is used to scale the value of the data used in this paper between 0 and 1 as a means to improves the prediction model.

TRAINING OF MODEL:

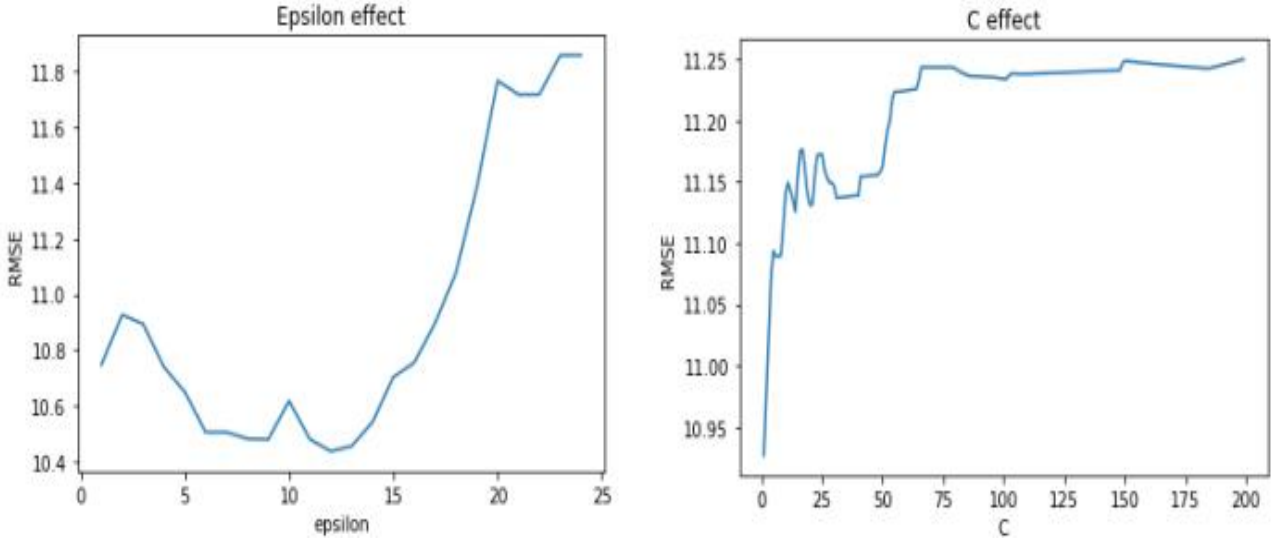
- ✓ Prior to the training of the model, the dataset has been split into a train set and test set by using the built-in function of the Scikit-learn library: sklearn.model_selection.train_test_split.
- ✓ 75% of the dataset has been categorized as a training dataset for the training of model while 25% of the dataset has been categorized as a test dataset for testing of the model.

✓ To train the model, the training dataset is fed into the SVR algorithm from the Scikit-learn library. There are a few parameters that need to be specified: C, Epsilon and Gamma value during the training of the model. They are set to default value during the first training, where C = 1, Epsilon = 0.1 and Gamma = 'Auto'. Gamma value is kept as 'Auto' for the rest of the project as 'Auto' indicates (1/number of features)

The best values for C and Epsilon of each segment are obtained through the reliable grid-search method. After the first model is trained by using the default parameters, the model is tested by using the test dataset. The accuracy of the model is evaluated via the root mean squared error (RMSE) where it shows the difference of the predicted duration and the actual duration in the test set.

The equation of RMSE can be represented as:

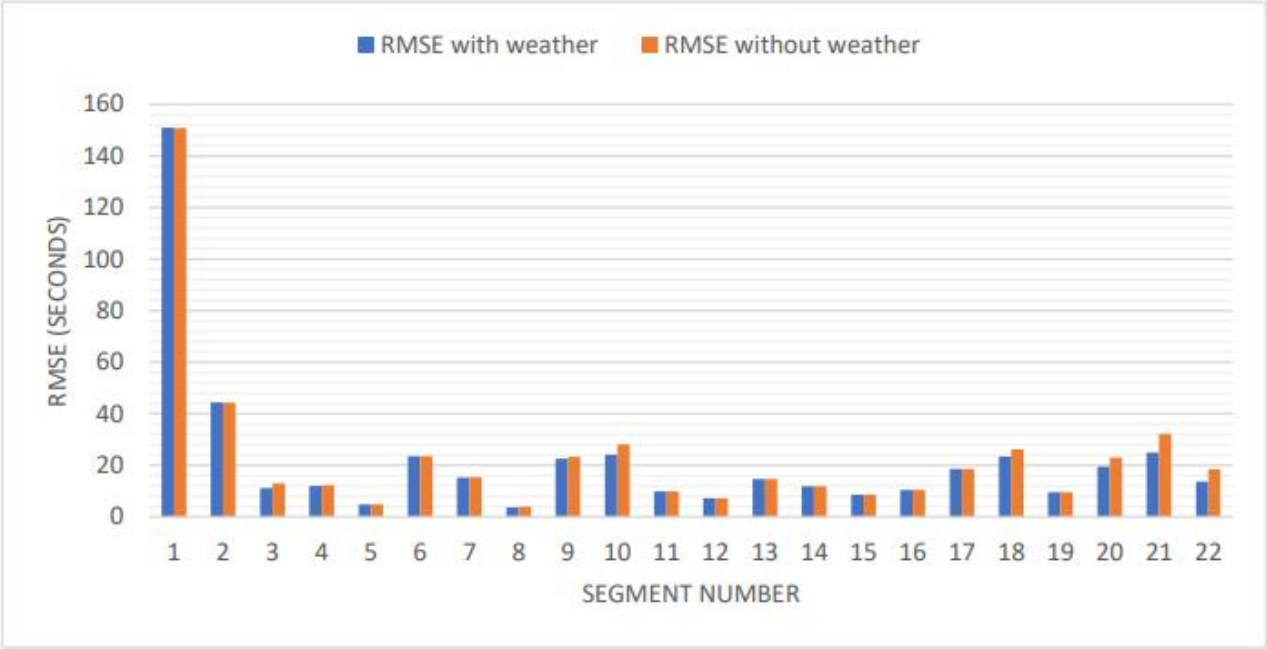
$$RMSE\ (Sec) = \sqrt{\frac{\sum_i^n (T_{A,i} - T_{P,i})^2}{n}}$$



LOWEST RMSE FOR CERTAIN EPSILON AND C VALUES

PERFORMANCE EVALUATION:

Lower RMSE represents better performance or accuracy in prediction.



Training of the model is always better with cleaner and well-structured data.There's the opportunity of increasing the performance of the model if more data samples are available.

CONCLUSION:

One of the main issues nowadays for bus operators is that the ETA is not accurate and it deviates from actual ETA by too much, and this discourages riders and so ridership is affected in the long run, hence the purpose of this paper is to develop a machine learning model that may provide more accurate ETA. SVR, which is based on the SVM classifier model, is chosen for this paper. The SVR model developed in this paper has displayed good prediction ability. One of the characteristics of SVR is that only a small amount of dataset is needed to train the model with the considerable performance and generalization capability. Input features for the training of the model include segments, distance, weather, peak or non-peak hour are used to predict the travel duration for the segments. The data has been categorized into peak and nonpeak hours based on the time the record is logged. Different types of hourly weather data have been assigned to each of the samples as well based on the log time. RBF kernel is chosen for this paper. By using the test dataset, the model is able to achieve 22 seconds of RMSE which is better than the result obtained in previous studies. Overall, SVR may be a feasible model for ETA prediction but it needs to be trained and tested vigorously with more data and features.