BE CSE VI (P BATCH)

CREATIVE AND INNOVATIVE PROJECT (CS6111)

OPTIMISED ROUTE PREDICTION ALGORITHM

MANUSCRIPT

TEAM MEMBERS:

- BALAJI S(2018103014)
- DHANANJEYAN AK(2018103523)
- ASHWATH NARAYAN KS(2018103517)

COLLEGE OF ENGINEERING,GUINDY-600025

ANNA UNIVERSITY

INTRODUCTION

Urban transportation is going through a rapid and significant evolution. In the recent past, the emergence of the Internet and of the smart-phone technologies has made us increasingly connected, able to plan and optimize our daily commute while large amounts of data are gathered and used to improve the efficiency of transportation systems. Today, real-time ridesharing companies like Uber or Lyft are using these technologies to revolutionize the taxi industry, laying the ground for a more connected and centrally controlled transportation structure, and building innovative systems like car-pooling.

A field that can make such important contributions is vehicle routing, i.e., the optimization of each vehicle actions to maximize the system efficiency and throughput. In this paper, we have proposed a working model that predicts optimal route for taxi travel with a lesser time complexity.

ABSTRACT

Travel optimization needs many criterias to look at. There are many attributes which contribute to travel path. The features should be looked at and look for optimal path considering the surroundings and physical and social conditions of the location.

The goal of the project is to optimize travel routes for a delivery vehicle by using **machine learning model predictions.** This is a two-component problem: first, we train a machine learning model on the data to predict how long it will take a delivery vehicle to go from point one point to another, and we feed these predictions into a **Genetic algorithm** which decides which is the most time efficient visit order for a given set of points.

"Genetic algorithm (GA) is a type of algorithm inspired by the process of natural selection to generate high-quality solutions to problems, which otherwise would be too difficult to solve."

For any given set of locations, these location are fed to the machine learning model, which predicts how long it will take to travel between each two given points. Then the algorithm "evolves" to find the visit order which minimizes time spent in transit. The genetic algorithm itself is fairly straightforward, but every genetic algorithm gives an optimal approximation.

RELATED WORKS

In this section, we briefly discuss about the published research related to vehicle optimization techniques. A first idea of **travelling salesman problem** first formulated in 1930 and is one of the most intensively studied problems in optimization. It is used as a benchmark for many optimization methods. But it was next to impossible to compute . So many heuristics and alternate algorithm were needed to implement the traditional TSP , so that some instances with tens of thousands of cities can be solved completely and even problems with millions of cities can be approximated within a small fraction of 1%.

In computer science and operations research, a **genetic algorithm** (**GA**) is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection. The first paper on this heuristic algorithm was published by Mitchell, Melanie (1996) as *"An Introduction to Genetic Algorithms"* **[1].**

An expansion of the Genetic Algorithm accessible problem domain can be obtained through more complex encoding of the solution pools by concatenating several types of heterogenously encoded genes into one chromosome. *(Cited by Patrascu M.; Stancu A.F.; Pop F. (2014). "HELGA: a heterogeneous encoding lifelike genetic algorithm for population evolution modeling and simulation")* **[2]**

This particular approach allows for solving optimization**[5]** problems that require vastly disparate definition domains for the problem parameters. For instance, in problems of cascaded controller tuning, the internal loop controller structure can belong to a conventional regulator of three parameters, whereas the external loop could implement a linguistic controller (such as a fuzzy system) which has an inherently different description. This particular form of encoding requires a specialized crossover mechanism that recombines the chromosome by section, and it is a useful tool for the modelling and simulation of complex adaptive systems, especially evolution processes. In our project, we combine two important Machine learning models XGBoost and Gradient Boost **[8]** to train the data and the trained data is passed into Genetic algorithm which is famous for its self optimization,to fulfill our optimization needs. The following is the detailed methodology.

SYSTEM DESIGN

As part of preprocessing, we split our dataset into test,train models and then preprocessing is done. ie. Any null values in data, unreal values and irrelevant data are removed. Then the process of feature engineering takes place to select features which should be responsible for path with more optimization. The model which we use to find distance based optimal model is **XGBoost** which is a Decision Tree based Ensembling method.

With a few random outliers in a huge data set, possibly extraneous features which came with it, and a number of possible categorical features, we need a **tree-based model**. Specifically, boosted trees will perform very well on this particular data set and be able to easily **capture non-linear relationships**, accommodate for **complexity**, and handle categorical features.

With our data and goals, a simple linear regression(existing system) won't do. Not only do we want to have a low variance model, we also know that the coordinates, while being numbers, do not carry numeric value for the given target variable. Additionally, we want to add direction of the route as a positive or negative numeric value and try supplementing the model with the **weather data set**, which is almost entirely categorical.

*"After doing a quick study with basic linear regression on the main dataset we realized that a **far more complex model** was needed. To this end, we selected a model to accommodate for **complex numeric and categorical features**. So we used XGBoost Ensembler."*

We also used another model when adding **weather data** because weather is also an important parameter in deciding travel path. In addition to the **standard XGBoost model**, we can try the LightGBM model because it is faster and has better encoding for categorical features. Once you encode these features as integers, one can simply specify the columns with categorical variables. This model combines distance and weather data using ensembling.

Then the Genetic algorithm "evolves" to find the visit order which minimizes time spent in transit. The genetic algorithm itself is fairly straightforward, but every genetic algorithm gives an optimal approximation.

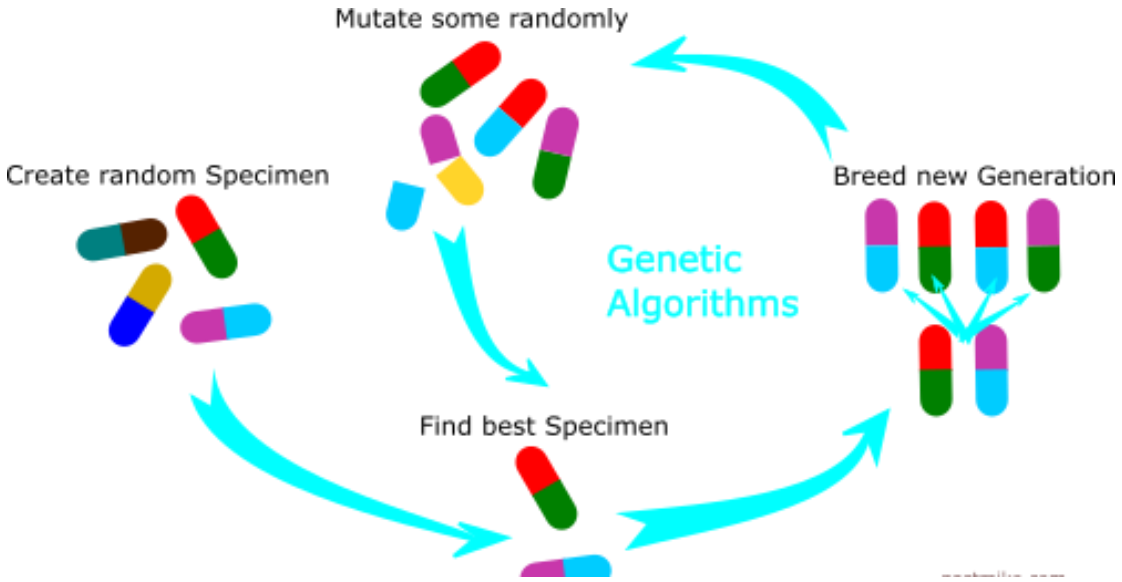

Fig 1 Genetic Algorithm Visualisation

This is a two component problem. Our initial step would be training a multi featured decision tree model on features from dataset and external features like distance, weather, traffic. Hence we train this using machine learning model of XGBoost, which is a decision tree model with ensembling features. It create multiple decision trees and combines them to yield better results. Then weather data is included by training a Light Gradient Boost Model and both results are combined. Then genetic algorithm, the evolutionary algorithm for solving np-hard problems is used to obtain predictions. Regular attributes of genetic algorithm like Crossover, Mutation are found in this algorithm too and finally predictions are made and visualised through UI features of python (Folium maps and GPX).

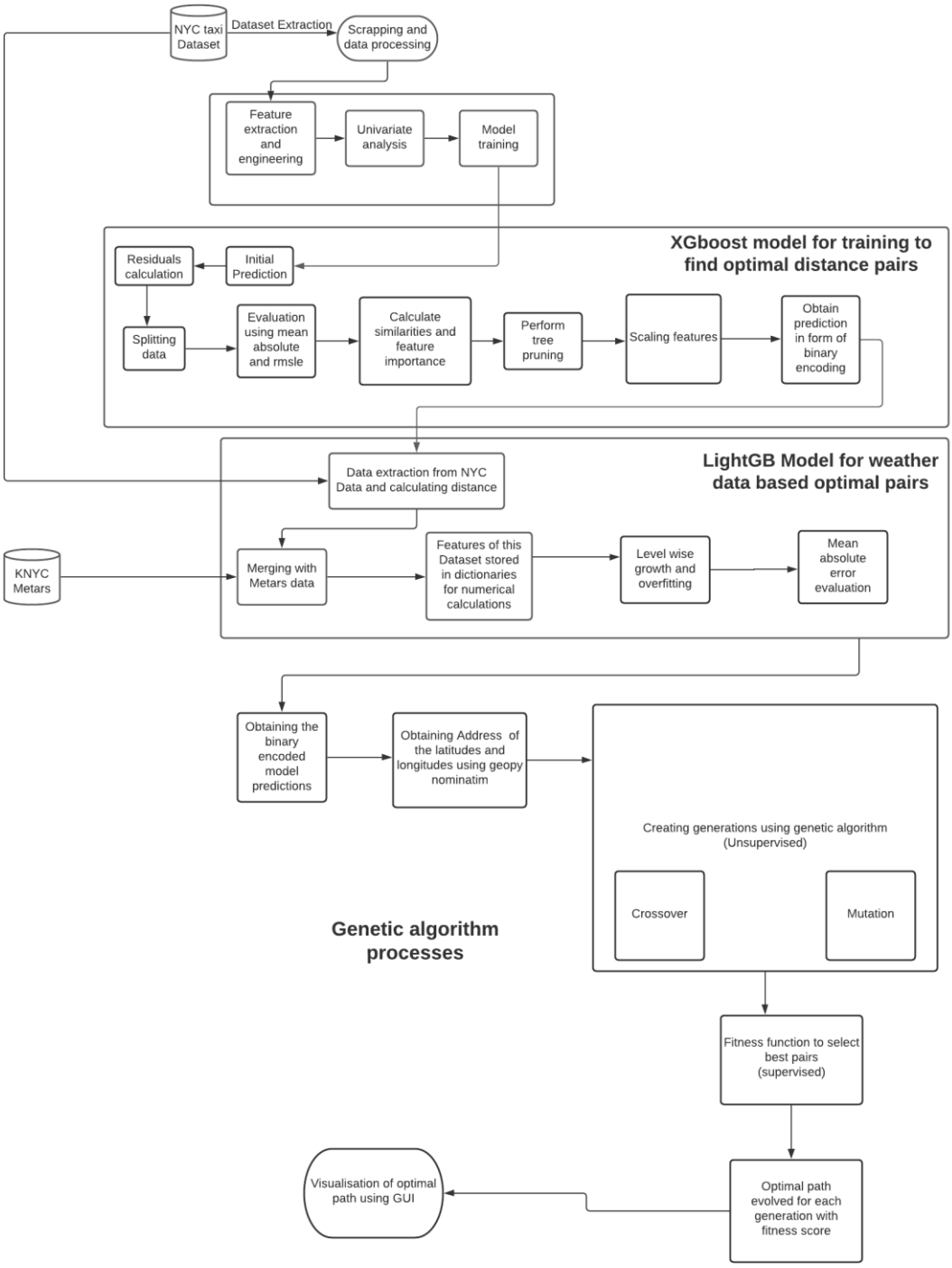

Fig 1 Architecture Diagram

MODULE 1

Data processing , Feature extraction ,univariate and bivariate analysis:

DESCRIPTION:

The first step is to obtain details of vehicle users and data of travel based on their destination routes. We will explore the data and modify dataset as per the our requirement for the further analysis of the problem. We alter date formats for convenience and use calculate function to find distance between pickup and dropoff coordinates.

- **Input:** Dataset of NYC taxi (2020) and KNYC_Metars (weather data).
- **Output:** Data (Table) of vehicle id ,pickup,drop location,latitude,etc…

MODULE 2

Training model creation and evaluation using weather and distance data:

DESCRIPTION:

When the travelling distance of two paths are different, the optimal path will be updated to the path with lower distance considering the traffic and weather input from a training model.

The data is split into training set ,test and validation sets. We extract details of class labels from pandas .Then the XGBoost model is trained and contents saved in a (.sav) file for quick access for upcoming processes.(As dataset elements are huge ,conversion to byte stream helps for easy access).

Input : Instances of modified dataset.
Output : Training model

Evaluation metric for model:
Square(log(predicted+1) – (log(true+1)).mean()**0.5

MODULE 3

Creating XGBOOST and LightGBM models and training of data:

DESCRIPTION:

We need a model to train on our dataset to serve our purpose of predicting the NYC taxi trip duration given the other features as training and test set. Since our dependent variable contains continuous values so we will use regression technique to predict our output.The aim is also to use a LightGBM model when adding weather data because this package can handle categorical data and runs faster..
In addition to the functionality of training set,this LightGBM model adds weather data to predict optimal routes based on weather also. This is a experimentary try to find optimal route across a network of roads.Then ,Model prediction and evaluation takes place.

Output: Optimal Distance between two points(node to node)

MODULE 4

GENETIC ALGORITHM:

The distance between two points is taken as **chromosomes.** They're sorted according to their random number. **Then crossover between two parents occur to form offsprings**. (Parent is cities and produced offsprings are helpful finding optimal path). In Crossover, parents fuse to form fittest offsprings.

Mutation: Original two point distance array is mutated into required style

Pseudocode for genetic algorithm:

```
For 50 points(nodes),
    For(iteration count < threshold){
    Gen( 50 chromosomes ie. distances);
    Gen(32 chromosomes through cross over);
    Gen( 14 chromosomes over mutation);
    Insert.chromosomes(4);
    Select(50 best fit chromosomes ie. 32+14+4);
    Iteration count++;
    }
    Present best solution obtained so far
```

Output: Optimal path

MODULE 5

PRESENT OPTIMAL ROUTE:

Then , optimal path is visualized into a path in route map using interface as it allows users to navigate between their paths easily and reach their destination in the quickest route with minimum amount of time.Visualisation of datapaths would be performed by folium maps and creations of gpx HTML files which can be loaded onto MyMaps to see the optimization.

FINAL OUTPUT: Visual optimal path between source to destination.

EXPERIMENTAL RESULTS:

Evaluation Parameters

• Learning rate (XGBOOST) of training model and epochs (num of rounds for training)

• Delay in travel (in mins)

• Mean Squared Error (using delay) – An evaluation metric for regression problems

• Mean Absolute error to get basic estimate of error

• Early stopping (prevents overfitting in genetic algorithm)

• Prediction Accuracy (XGBoost and LightGBM model accuracy)

CORRESPONDING FORMULAE:

- **Training model :**

Training -> xgb's matrix(X_train, log(y_train+1))

Valuate-> xgb's matrix(X_val, log(y_val+1))

- **To check error:**

Mean (absolute value(pred - y_test))

- **Evaluation metric:**

def rmsle(y_true, y_pred):

assert len(y_true) == len(y_pred)

return np.square(np.log(y_pred + 1) - np.log(y_true + 1)).mean() ** 0.5

- **Test prediction:**

exponent(predict(X test)) – 1

- **Genetic algorithm:**

Y=ax^5+bx^4+cx^3+dx^2+ex+f

Genes are a,b,c,d,e,f

Chromosome is array [a,b,c,d,e,f]

For every actual data point (x,y) Computing y'= x^5+bx^4+cx^3+dx^2+ex+f (Here genes undergo crossover,mutatio n ,values changes)

Find sum of (y-y')^2 over all x

Accuracy calculation:

Accuracy= Number of correct predictions/Total predictions

Generation creations in Genetic algorithm to obtain optimal path…

For say from 1 to 100 generations, best score (used to optimize) and path remains same… (accuracy can be verified)

Graphs:

Learning curves for XGboost regressor model and LightGBM model

Nyc Taxi Dataset

The yellow and green taxi trip records include fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts. The data used in the attached datasets were collected and provided to the NYC Taxi and Limousine Commission (TLC) by technology providers authorized under the Taxicab & Livery Passenger Enhancement Programs (TPEP/LPEP).

KNYC Metars (Weather Data)

METAR is a format for reporting weather information. A METAR weather report is predominantly used by pilots in fulfillment of a part of a pre-flight weather briefing, and by meteorologists, who use aggregated METAR information to assist in weather forecasting.

Experimental and Final Results

DIFFERENT INPUT TEST CASES

SYSTEM TEST CASES:

TEST CASE ID	NO.OF TEST LOCATIONS	PRODUCED OUTPUT
TC_01	5	['L4', 'L5', 'L1', 'L2', 'L3', 'L4']
TC_02	2	Unfortunately , two node routes are not predicted correctly , resulted in a lengthier path than expected.
TC_03	1	It produces a **travel time of 3.25 mins** by the model which is not possible.So this scenario **cannot be considered.**
TC_04	3	['L3','L1','L2','L3']
TC_05	11	['L9','L10','L11','L3','L5','L8','L2','L1', 'L7','L6','L4','L9']
TC_06	7	['L7','L5','L6','L1','L4','L3','L2','L7']
TC_07	9	['L9','L1','L3','L6','L8','L5','L4','L7', 'L2','L9']

FINAL OUTPUT OF 9 TEST LOCATIONS:

'L1': (40.763939,-73.979027),

'L2': (40.793209,-73.973053),

'L3': (40.757839,-73.969017),

'L4': (40.738400,-73.999481),

'L5': (40.727226,-73.992195),

'L6': (40.76040649,-73.97044373),

'L7': (40.76646805,-73.97818756),

'L8':(40.756680,-73.962982),

'L9':(40.767937,-73.982155),

```
addresses
```

```
['Avenue of the Americas Plaza, West 55th Street, Midtown, Manhattan Community Board 5, Manhattan, New York County, New York, 1
0019, United States',
 'Broadway & West 94th Street, Broadway, Upper West Side, Manhattan Community Board 7, Manhattan, New York County, New York, 10
025, United States',
 'Lipstick Building, 885, 3rd Avenue, Turtle Bay, Manhattan Community Board 6, Manhattan, New York County, New York, 10035, Uni
ted States',
 '154, West 14th Street, West Village, Manhattan Community Board 2, Manhattan, New York County, New York, 10011, United State
s',
 '38, East 4th Street, NoHo, NoHo Historic District, Manhattan, New York County, New York, 10012, United States',
 'Tower 56, 126, East 56th Street, Midtown East, Manhattan Community Board 6, Manhattan, New York County, New York, 10022, Unit
ed States',
 'JW Marriott Essex House, 160, Central Park South, Midtown, Manhattan Community Board 5, Manhattan, New York County, New York,
10019, United States',
 '435, East 55th Street, Midtown East, Manhattan Community Board 6, Manhattan, New York County, New York, 10022, United State
s',
 'Columbus Circle, Broadway, Theater District, Times Square, Manhattan, New York County, New York, 10107, United States']
```

Fig 2 Address of 9 test locations

UNOPTIMISED OUTPUT:

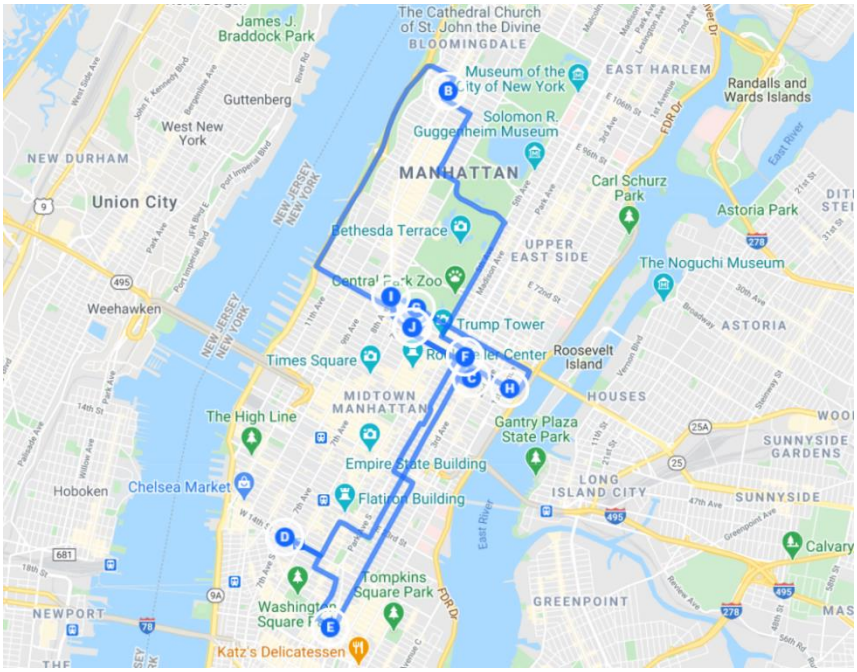

Fig 3 Unoptimised route for 9 test locations

OPTIMISED OUTPUT:

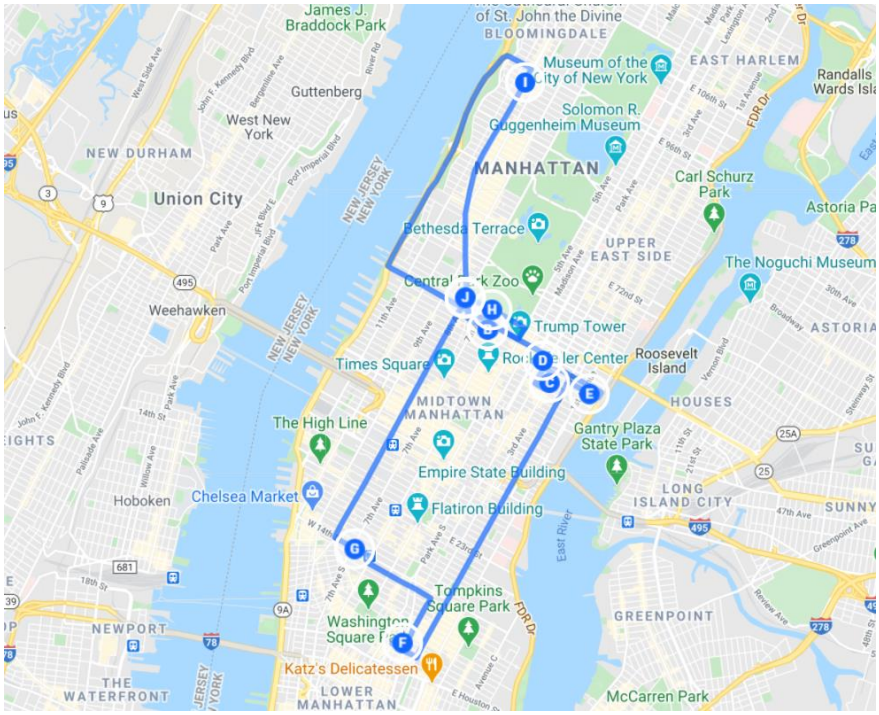

Fig 4 Optimised route for 9 test locations

RESULT ANALYSIS

Performance Metrics:

Distance data is processed , feature scaling is done and model used to predict is Random Forest Regressor which results in high error scores.

We use XGBoost Model to reduce RMS error to produce accurate results with high degree of ensembling . We combine distance and weather using ensembling technique of boosting the weaker features to produce strong learners. If we use Random Forest this specific focus will be missed.

At last Point we can see that our MSE is around 10000 and the MSE Error in Random Forest was two times greater.

We can observe that XGBoost model produces lesser RMSE error compared to Random Forest regressor and hence it is improved to produce predictions for travel.

Also ,the gap at end indicates the overfitting conditions.

Model : Regression Model with Ensembling

Hyper parameters: Learning Rate

Evaluation : Root mean square error

Epochs : Varying epochs

Testing: Mean absolute error

Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient. Furthermore, the learning rate affects how quickly our model can converge to a **local minima** (aka arrive at the best accuracy).

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

Inference Table:

Test	Validation	Epochs	Learning Rate	RMSE(Test)	RMSE (Eval)	Mean Error	Mean Error Final (All features considered)
30%	25%	500	0.2	0.07641	0.34768/0.34767 **(Underfitting)**	4.9699	4.906
40%	35%	700	0.1	0.07913	0.3417/0.3417	5.073	4.990
40%	35%	1000	0.1	0.0565	0.34188/0.34188	5.0449	4.991
40%	35%	1300	0.1	0.04191	0.34197/0.34197	5.027	4.9928
30%	25%	2000	0.05	0.0944	0.327426/0.327425/0.327426 **(Indicates overfitting. Hence we stop the epoch)**	4.9564	4.8273

Graphical Representation of results:

Learning Curve of Random Forest:

Fig 5 Learning curve of Random Forest

Learning curve of XGBoost (Used in our model)

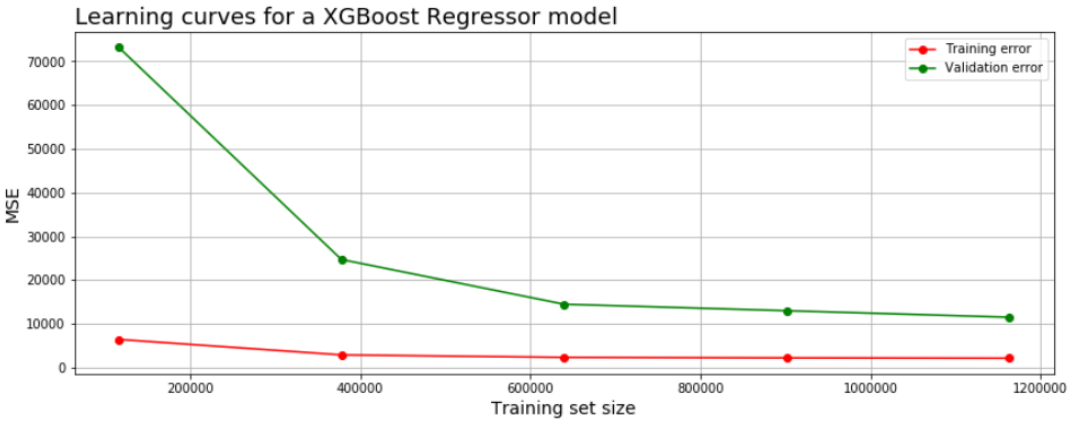

Fig 6 Learning curve of XGBoost Model

DISTANCE AND HOUR GRAPH

```
[9]: group5 = sample_df.groupby('pickup_hour').trip_distance.mean()
     sns.pointplot(group5.index, group5.values)
     plt.ylabel('Distance (km)')
     plt.show()
```

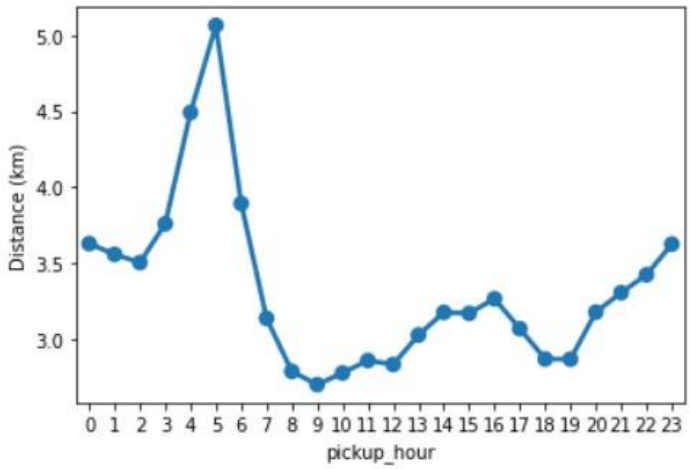

Fig 7 Distance and Hour graph

FEATURES VS INFORMATION GAIN:

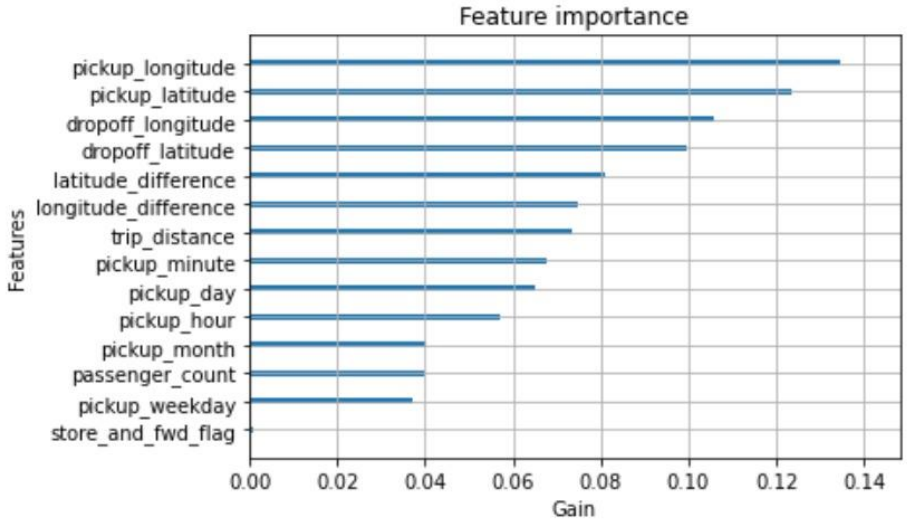

Fig 8 Features vs Information Gain

EPOCHS VS MEAN ERROR:

```
In [6]: import matplotlib.pyplot as plt

        #Line graph
        a = [500,700,1000,1300,2000]
        b = [4.906,4.990,4.991,4.9928,4.8273]
        plt.plot(a,b)
        plt.title("Epoch vs mean error")
        plt.xlabel("epoch")
        plt.xlabel("mean error")
        plt.show()
        #scatter graph
        x_values = [500,700,1000,1300,2000]
        squares = [4.906,4.990,4.991,4.9928,4.8273]
        plt.scatter(x_values,squares, s=10, color = "pink")
        plt.show()
```

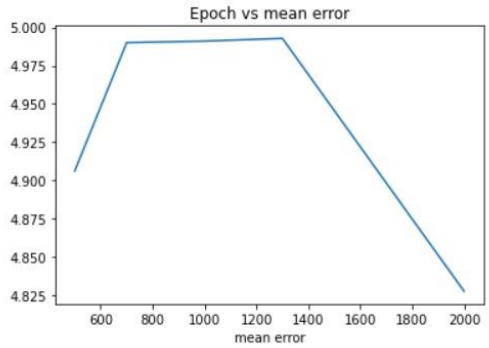

Fig 9 Epoch vs Mean error

We can see that in **Fig 9, mean error gets reduced gradually** with **increase in epochs.**

EPOCHS VS RMSE ERROR:

```
In [7]: import matplotlib.pyplot as plt

        #line graph
        a = [500,700,1000,1300,2000]
        b = [0.07641,0.07913,0.0565,0.04191,0.0944]
        plt.plot(a,b)
        plt.title("Epoch vs mean error")
        plt.xlabel("epoch")
        plt.xlabel("RMSE error")
        plt.show()
        #scatter graph
        x_values = [500,700,1000,1300,2000]
        squares = [4.906,4.990,4.991,4.9928,4.8273]
        plt.scatter(x_values,squares, s=10, color = "pink")
        plt.show()
```

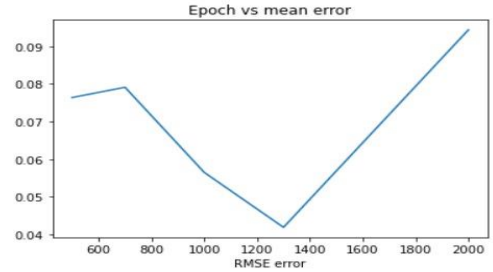

Fig 10 Epochs vs RMSE Error

We can see in **Fig 10** an **increase in RMSE error** due to **overfitting.**

CONCLUSION

The genetic algorithm proved to be efficient for almost all the test locations used and predicted the optimised path**.** The new XG Boost + Genetic algorithm combo has increased overall accuracy in finding optimal path. The XGB model had an *error of only 4.9 minutes* in estimating a single trip's duration for a taxi. While this may seem acceptable for one trip, the error may get bigger the more locations are visited. The genetic algorithm itself is fairly straightforward, but it must be noted that every genetic algorithm gives an optimal approximation, but not the single best solution there is since it is a regression problem.

Since the algorithm had difficulties In finding the optimal paths for few of the test locations(For ex: 1 test location),it did better in generating an optimal path for other cases.

REFERENCES

1. *"An Introduction to Genetic Algorithms"* by Mitchell, Melanie (1996)
2. *"HELGA:A heterogenous encoding lifelike genetic algorithm for population evolution modeling and simulation"* by Patrascu M,Stancu A.F
3. Kaas, K. Benefit of Traffic Incident Management, National Incident Management Coalition (NTIMC).
4. http://www.transportation.org/sites/ntimc/docs/Benefits11-07-06.pdf
5. Hagen, L.T. Best Practices for Traffic Incident Management in Florida; CUTR Report 2005, No. 21170543;U.S. Department of Transportation: Washington, DC, USA, 2000.
6. Yin, Y. A scenario-based model for fleet allocation of freeway service patrols. Netw. Spat. Econ. 2008, 8,407–417
7. https://www.analyticsvidhya.com/blog/2017/07/introduction-to-genetic-algorithm/
8. cs.princeton.edu (The Boosting Approach to Machine Learning An Overview)
9. https://www.hindawi.com/journals/mpe/2020/6348578/
10. Mitchell, M. An Introduction to Genetic Algorithms; MIT Press: Cambridge, MA, USA, 1996; ISBN 9780585030944