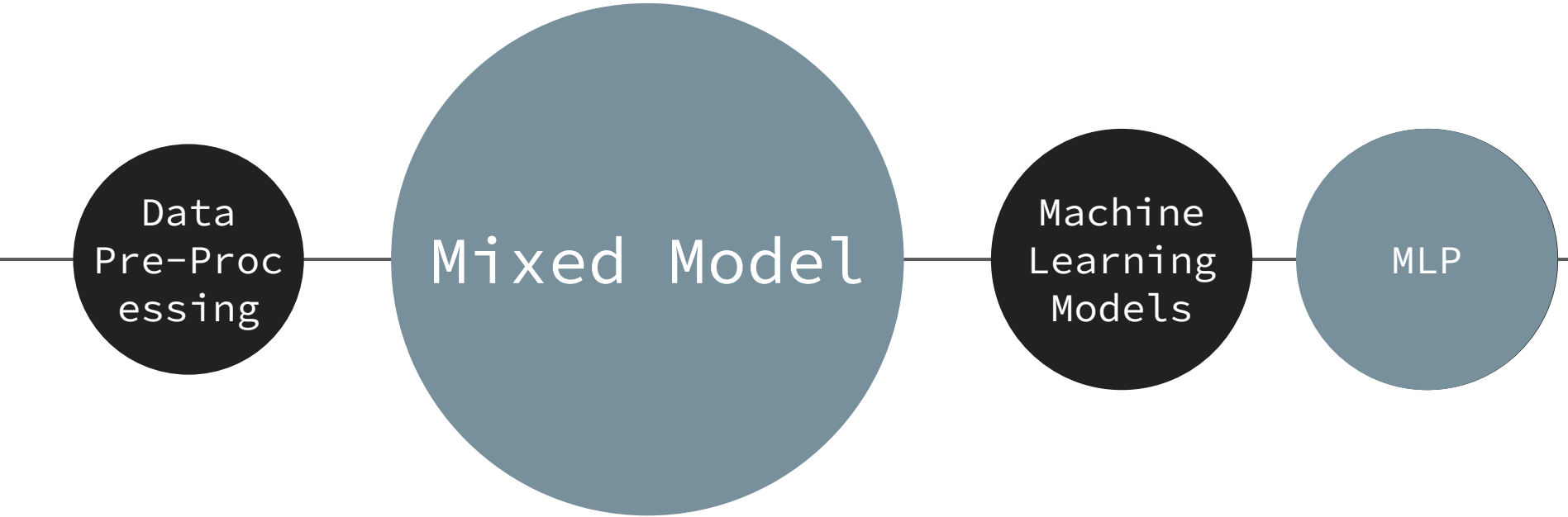
Predicting the Travel Length of Taxi in Portugal

Wenzhou Lyu
Yifei Wang
Yunlong Wang
Chenyang zhou

Summary

- We are Wenzhou Lyu, Yifei Wang, Yunlong Wang, Chenyang Zhou from team **Kaggle Bot**
- We found that **feature engineering** is more important than model selection.
- We found that a **combination** of machine learning and deep learning could produce amazing results.
- We learned how to use Pytorch on conducting a predicting task from scratch.
- Platforms: Deep Learning Models are trained on NVIDIA 3070, machine learning models are trained on CPU.

Key Words


Data Pre-Processing
Mixed Model
Machine Learning Models
MLP

Introduction

Team Introduction

- Wenzhou Lyu: 3rd-year Math-CS and DSC double majors.
- Yifei Wang: 3rd-year CS major
- YunLong Wang: 4th CS major
- Chenyang zhou: 4th CS major

Methodology

Data Processing and Feature Engineering

- Drop TRIP_ID, ORIGIN_CALL, DAY_TYPE, MISSING_DATA
- Convert POLYLINE to travel length (Length of the list -1) * 15
- Extract week, day, month, hour information from TIMESTAMP.
- One-hot encoding week, day, month, hour. (7, 31, 12, 24 unique values respectively)
- One-hot encoding TAXI_ID (448 unique Ids)
- One-hot encoding CALL_TYPE (3 unique values)
- One-hot encoding ORIGIN_STAND (64 unique values)
- Remove data with travel length <60 and data > mean + 3*std.
- Split the data into training and validation set (0.75:0.25)
- The result training dataset has 589 (448 +3 +12 +31 +24 +7 +64) columns of 0 or 1 after encoding.

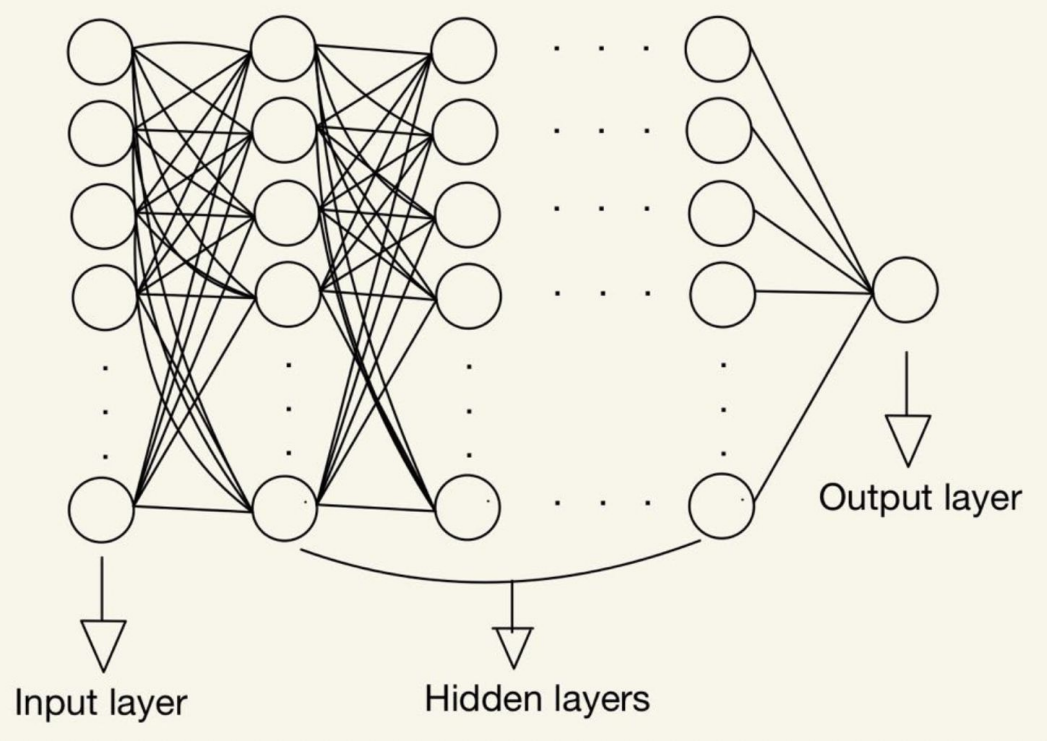
Models

- Machine Learning Models
- MLP
- Combined Models (Heuristic-oriented model)

Machine Learning Models

LightGBM Regressor	Random Forest Regressor	Bayesian Ridge Regression	Linear Regression
100 Leaf-Wise Growth Gradient Boosting Decision Trees	100 Decision Trees Combined	A Probabilistic Model Based on Gamma Distributions	A Hyperplane Fits the Pre-Processed Data

Initial MLP Model

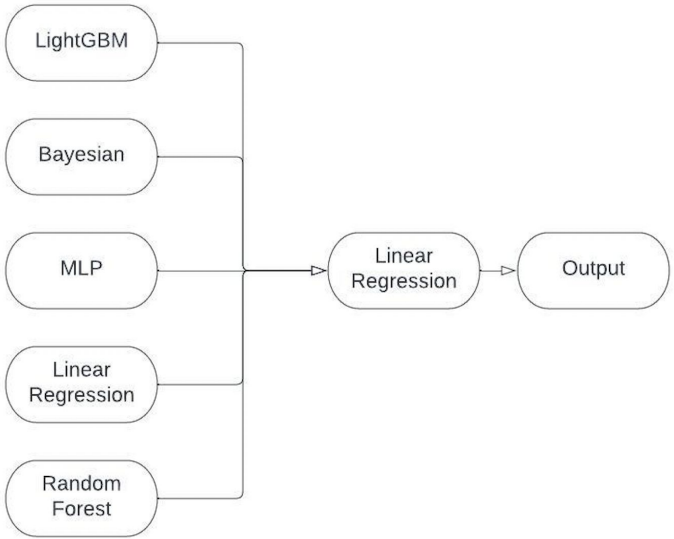


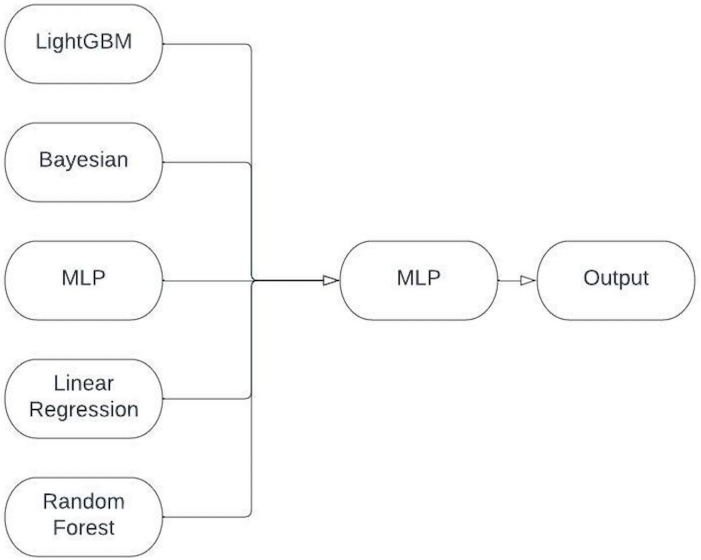
Architecture:

- 3 Hidden Layers
- Input Dimension:589
- First Hidden Layer: dimension 589 -> 128
- Second Hidden Layer: dimension 128 -> 64
- hird Hidden Layer: dimension 64->32
- Output Layer: dimension 32->1
- ReLu Activation Function for each fully connected layer
- Regularization: dropout after each fully connected layer with probability of 0.25
- MSE as Loss Function

Engineering Tricks and Combined Models

Inspired by the **boosting** in machine learning, we want to see if we can combine all the five models and feed the predictions using the five models into another model, expecting the new model could produce a better result.





Combined Model

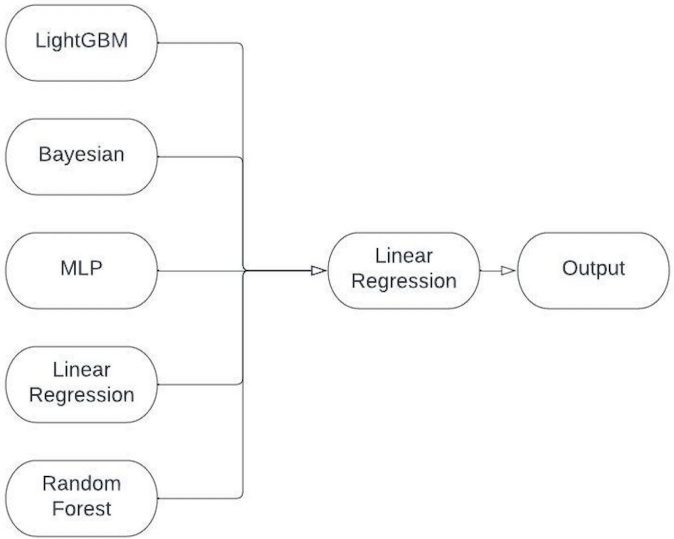
Step 1: We first use each model to predict the original training set. The output of each model is an array containing all the travel time predictions. It has the same length as the training set. We would have five of those arrays, and we save them into a csv file for future use. Also, we do the same operation on the validation set and test set, and save them into csv for future use

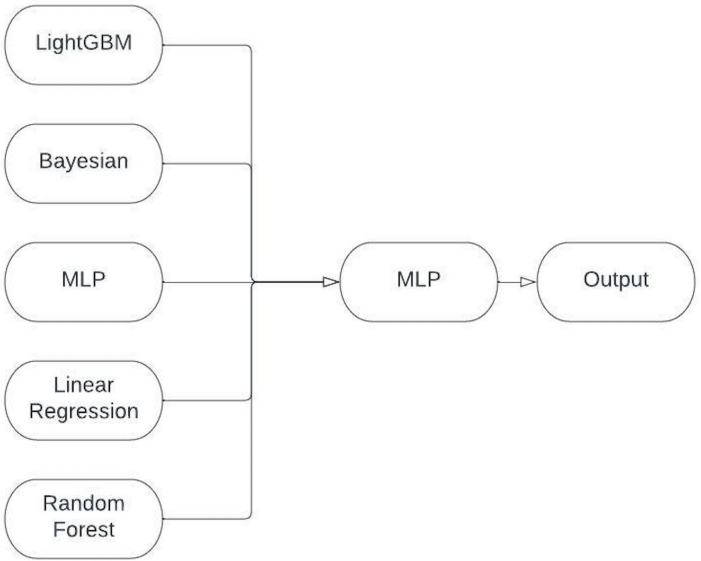
	A	B	C	D	E
1	lightGBM	linear regression	random forest	BayesianRidge	multi-layer perceptron
2	5.34E+02	4.66E+02	4.65E+02	4.66E+02	4.74E+02
3	6.34E+02	6.22E+02	4.65E+02	6.22E+02	6.07E+02
4	5.94E+02	5.67E+02	4.95E+02	5.69E+02	5.51E+02

Combined Model

Step 2: We then use this new data as the training set, and apply different models (Linear Regression, MLP) on this set.

Step 3: We test on different models and choose the best model to do prediction on the test set.





Other Tricks

- Round our final prediction result to multiples of 15 (later Abandoned).
- Save pre-processed data after each step into csv files, make them convenient for future use.

Experiments

Experiment 1.1 Different MLPs

- Architecture 1:
  - 6 hidden layers,
  - input dimension: 589
  - First Hidden Layer: dimension 589 -> 32
  - Second Hidden Layer: dimension 32 -> 32
  - Third Hidden Layer: dimension 32->32
  - Fourth Hidden Layer: dimension 32 -> 32
  - Fifth Hidden Layer: dimension 32 -> 32
  - Sixth Hidden Layer: dimension 32 -> 32
  - Output Layer: dimension 32->1
- Architecture 2:
  - 3 hidden layers,
  - input dimension: 589
  - First Hidden Layer: dimension 589 -> 128
  - Second Hidden Layer: dimension 128 -> 64
  - Third Hidden Layer: dimension 64->32
  - Output Layer: dimension 32->1
- Architecture 3:
  - 4 hidden layers,
  - input dimension: 589
  - First Hidden Layer: dimension 589 -> 1000
  - Second Hidden Layer: dimension 1000 -> 750
  - Third Hidden Layer: dimension 750 -> 500’
  - Fourth Hidden Layer: dimension 500 -> 250
  - Output Layer: dimension 250 -> 1

Experiment 1.1 Discussion

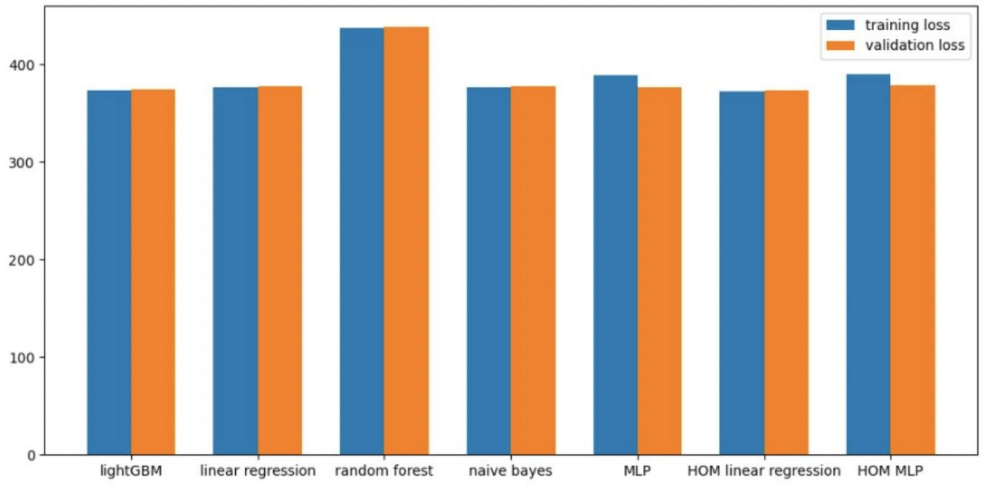
Architecture 2 has the lowest Kaggle Score. Therefore, we choose Architecture 2 as the best model.

Model	Kaggle Score
Architecture 1	786.50741
Architecture 2	770.33254
Architecture 3	775.92011

Experiment 1.2 All Models Comparison

model	training loss	validation loss	public test score
lightGBM	373.9727168608378	374.76647406151574	771.9719
Linear Regression	376.869386537811 2	377.6907599107631	778.9339
Random Forest Regressor	437.481232577257 97	438.3986068549048	789.78389
BayesianRidge	376.8711576634640 3	377.6935761656919	782.72438
MLP	389.241	376.871	769.66934
HOM Linear Regression	372.875127786182 2	373.6816272642171	763.3734
HOM MLP	391.137355158027 84	375.38357701356085	760.99909

Note: MLP in the table is Architecture 2 with Dropout (p=0.25)



Validation Loss >400 => Worse Kaggle Score
When Validation Losses are close, use Kaggle Score as the metric => HOM MLP is the best!

Discussion

What have you learned

- The most effective feature engineering strategy for this task is one-hot encoding.
- Regularization is important, applying dropout/normalization is necessary.
- Using a reliable device to train is necessary.
- Complicated Models tend to produce a best result.

Future Work

- Use CNN on route map of each taxi and find the spatial patterns of trips.
- Utilize the time series information (ordered by TIMESTAMP column), and apply Transformer or LSTM on the encoded information.
- Use Sklrean GridSearch on machine learning to find the best parameters combination (if our CPU is reliable enough).
- Use GPU to train LightGBM Regressor.