

# TAXI DEMAND PREDICTION USING BIG DATA ANALYTICAL TECHNIQUES

ABSTRACT. This empirical paper compares the accuracy of six univariate methods for short-term taxi demand forecasting for lead times up to a week-ahead. The methods are compared using a time series of half hourly demand for New York.

Tentative plan:

- Introduction
- Exploratory Data Analysis
- Data Preparation
- Time Series Models
  - Holt-Winters seasonal method
- Supervised Learning Models
  - Linear Regression
  - Conditional Inference Decision Tree
  - Random Forest
  - XG Boost
- Deep Learning Model - LSTM
- Conclusions

## 1. INTRODUCTION

Taxi-supply planning requires efficient management of existing taxis and optimization of the decisions concerning additional capacity. Demand prediction is an important aspect in the development of any model for taxi planning. The form of the demand depends on the type of planning and accuracy that is required; hence it can be represented as an annual demand, a peak demand or demand like daily, weekly etc. Short-term demand forecasts are required for the control and scheduling of taxis. The focus varies from minutes to several hours ahead. The predictions can help in optimizing taxi supply at a given location and time.

In the short run, the taxi demand is mainly influenced by seasonal effects (daily and weekly cycles, calendar holidays) and special events. Weather related variation is certainly critical in predicting taxi demand for lead times beyond a day ahead. In this paper, we compare the accuracy of simple benchmarks and four more sophisticated methods. We evaluate the methods using 24 weeks data for the New York city. We consider lead times up to a week ahead.

Refer to [1] for a detailed about World of Warcraft game.

Keep your comments or questions like this

February 21, 2018

1.1. **Sample algo1.** Calculate the following probabilities:

**Data:** 5 seconds sampled order book data

**Result:** States Table with probabilities

Initialization ;

1. Get state  $\hat{S} = (\hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4, \hat{s}_5)$  at time time t;
2. Run K-means clustering and find relevant cluster for  $\hat{S}$ . Let us call this cluster as  $S(t)$  ;
3. Update counter for  $S(t)$
4. If  $P_t > P_{t+\delta} + 5$  basis points, update upCounter for  $S(t)$ ;
5. If  $P_t > P_{t+\delta} + 5$  basis points, update downCounter for  $S(t)$ ;
6. Calculate upProbability and downProbability for  $S(t)$  ;
7. Prepare state table with probabilities

**Algorithm 1:** State Calculation Train Algorithm

1.2. **Sample table.**

State	upProbability	downProbability
$S_1$		
$S_2$		
$\vdots$		
$S_{24}$		

1.3. **Sample algo2.** Sample algorithm

**Data:** 5 seconds sampled order book data

**Result:** Buy, sell and hold signal

Initialization ;

1. Get state  $\hat{S} = (\hat{s}_1, \hat{s}_2, \hat{s}_3, \hat{s}_4, \hat{s}_5)$  at time  $t$ ;
2. Run K-means clustering and find relevant cluster for  $\hat{S}$ . Let us call this cluster as  $S(t)$  ;
3. Look at the state table for respective upProbability and downProbability

```

while presentTime > 9.25 and presentTime < 15.20 do
  read current;
  if (upProbability > 0.7)&(inventory == 0) then
    tradingSignal = Buy ;
    inventory = 1;
  else if (downProbability > 0.7)&(inventory == 0) then
    tradingSignal = Sell ;
    inventory = -1;
  else if (downProbability > 0.4)&(inventory == 1) then
    tradingSignal = Sell ;
    inventory = -1;
  else if (upProbability > 0.4)&(inventory == -1) then
    tradingSignal = Sell ;
    inventory = -1;
  else
    tradingSignal = Hold ;
    Move to the next 5 seconds time step
end

```

**Algorithm 2:** State Calculation Train Algorithm

#### 1.4. Conclusions.

#### REFERENCES

- [1] Marcelo Souza Nery, Victor do Nascimento Silva, Roque Anderson S. Teixeira and Adriano Alonso Veloso *Setting Players Behaviors in World of Warcraft through Semi-Supervised Learning*.