**Leeds** School of Business
UNIVERSITY OF COLORADO **BOULDER**

# MSBX5420 Team Project – Design Document

**Team :** Team Blanca Peak **Date:** 04/25/2020

**Team Members :**

| | |
|---|---|
| Chaerin Lee | Chaerin.Lee@colorado.edu |
| Dean Duke | Dean.Duke@colorado. edu |
| Ethan Goldbeck | Ethan.Goldbeck@colorado.edu |
| Matthew Kuchar | Matthew.Kuchar@colorado.edu |
| Soumya Panda | Soumya.Panda@colorado.edu |

## Document History

| Version | Date | Author | Comments |
|---|---|---|---|
| 1.0 | 4/19/2020 | Team Blanca Peak | Created initial version |

# Table of Contents

# 2. Introduction

## 2.1 Purpose

This document provides a comprehensive architectural overview of MSBX-5420 project of team blanca peak. This design document is prepared as per the requirement specification defined for our project.

# 3. Application / System Design Specification

### 3.1.1 Application / System Design

**Ingest Dataset in S3:**

Download csv files from https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page and ingest in Leeds S3 bucket using the following commands :

ssh to the EMR cluster and copy csv files.

```
sudo ssh -i ./Leed_HadoopKeypair.pem
hadoop@ec2-52-13-183-139.us-west-2.compute.amazonaws.com
```

Prerequisite : Download the .pem and .ppk files for Leeds AWS

**Read CSV and Save Dataset as Parquet :**

Create a group s3 bucket and copy the 2019 yellow cab files from amazon's open data repository

```
aws s3 mb team-blanca-peak
aws s3 cp s3://nyc-tlc s3://team-blanca-peak --recursive --exclude “*” --include
“yellow_tripdata_2019*”
```

Used a python script and pyarrow to read the csv and transform into parquet

```
import pandas as pd
import pyarrow
for i in range(1,13):
  if i<10:
      tmp_df=pd.read_csv(f'yellow_tripdata_2019-0{i}.csv')
      tmp_df.to_parquet(f'yellow_tripdata_2019-0{i}.parquet')
```

```
else:
    tmp_df=pd.read_csv(f'yellow_tripdata_2019-{i}.csv')
    tmp_df.to_parquet(f'yellow_tripdata_2019-{i}.parquet')
```

Pseudo Code :

```
1. import SparkSession and create SparkContext
2. SparkContext.textFile(“csv path”)
3. data frame.write.parquet("s3:team-blanca-peak”)
```

## Read parquet file and display records :

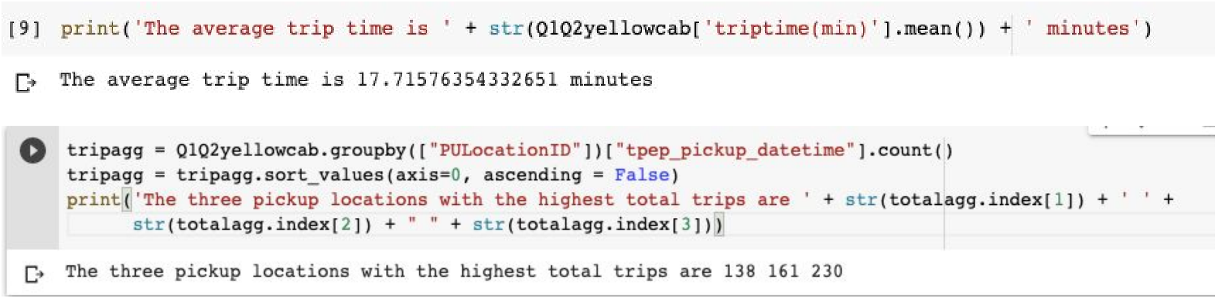
Pseudo Code :

```
data frame = sqlContext.read.parquet("s3:team-blanca-peak")
data frame.show(5)
```

**Data Analysis:**

Top pickup/dropoff locations
Average trip distance (total and by hour)
Converted dates into datetime
Average trip time by minutes
Most popular months for pickups
Most popular hours for pickups

Example:



```
[9] print('The average trip time is ' + str(Q1Q2yellowcab['triptime(min)'].mean()) + ' minutes')
```

```
The average trip time is 17.71576354332651 minutes
```

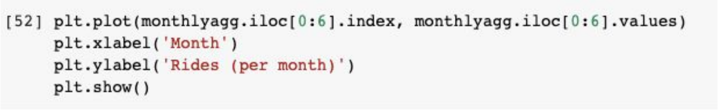
```
tripagg = Q1Q2yellowcab.groupby(["PULocationID"])["tpep_pickup_datetime"].count()
tripagg = tripagg.sort_values(axis=0, ascending = False)
print('The three pickup locations with the highest total trips are ' + str(totalagg.index[1]) + ' ' +
      str(totalagg.index[2]) + " " + str(totalagg.index[3]))
```

```
The three pickup locations with the highest total trips are 138 161 230
```

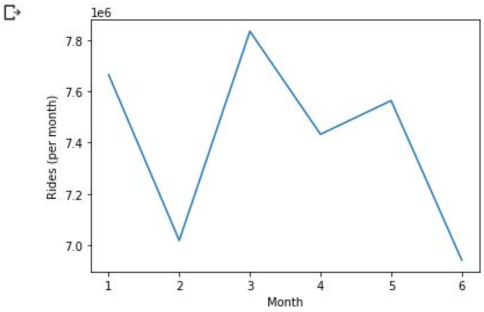
**Data Visualization:**

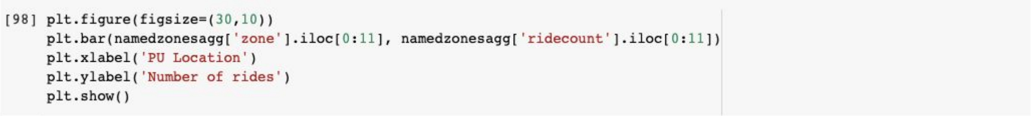
Rides per month
Rides per hour
Average Trip distance per month
Average Trip distance per hour
Top 10 Pickup locations

Examples:

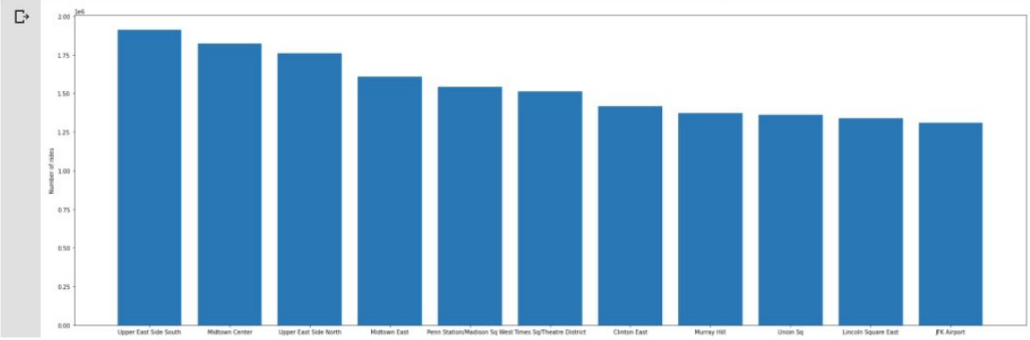


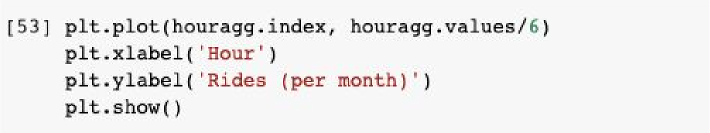
```
[52] plt.plot(monthlyagg.iloc[0:6].index, monthlyagg.iloc[0:6].values)
     plt.xlabel('Month')
     plt.ylabel('Rides (per month)')
     plt.show()
```



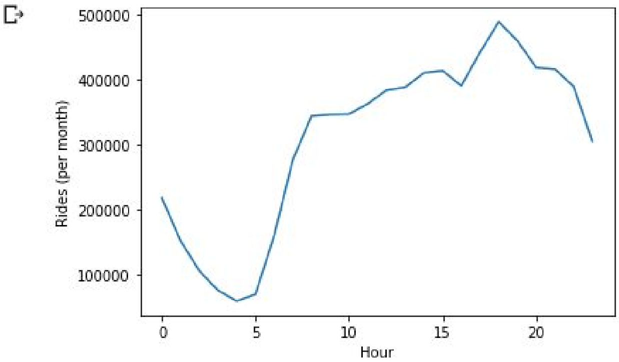


```
[98] plt.figure(figsize=(30,10))
     plt.bar(namedzonesagg['zone'].iloc[0:11], namedzonesagg['ridecount'].iloc[0:11])
     plt.xlabel('PU Location')
     plt.ylabel('Number of rides')
     plt.show()
```





```
[53] plt.plot(houragg.index, houragg.values/6)
     plt.xlabel('Hour')
     plt.ylabel('Rides (per month)')
     plt.show()
```



**Machine Learning Model:**

1. Classification model using pyspark.ml.classification RandomForest determine the tip amount based on various factors.

Steps :

1. Read data from parquet and load in the pyspark dataframe.
2. Data cleansing by filling the null values with 0 for float and Unknown for the string fields and drop unnecessary columns.
3. Create the boolean Y column based on the tip amount > 0 then 1 else 0
4. Converting the categorical columns to vectors using onehotencoder
5. Set the Y and X features and labels
6. Divide the dataset to train, test and validation based on 80:10:10
7. Fit the model to the train dataset
8. Test the model using test dataset
9. Tune the parameters and re-test
10. Fit the cross validation model to the validation dataset
11. Draw confusion matrix and evaluate the matrix

Pseudo Code :

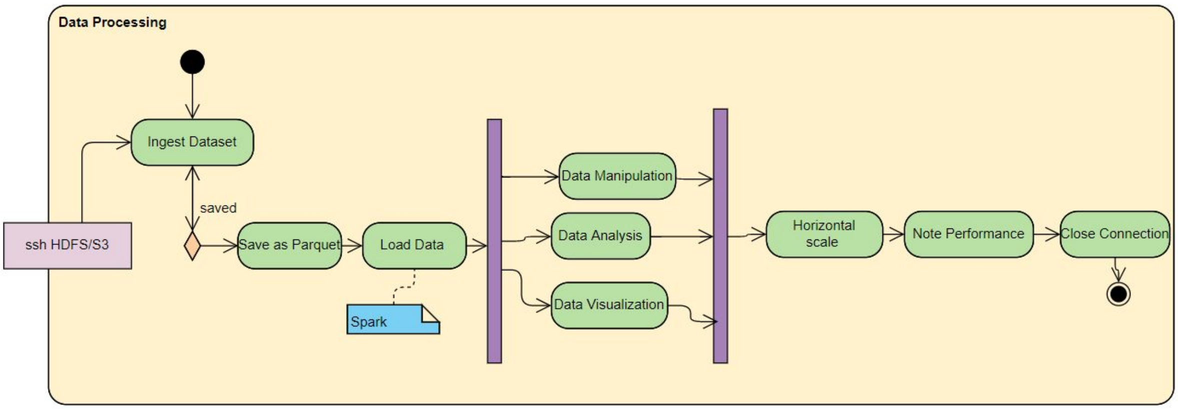
```
data frame = sqlContext.read.parquet("s3:team-blanca-peak")
data frame = df.na.fill(0)
data frame.withColumn 'tip' when 'tip_amount' > 0 then 1 otherwise 0
OneHotEncode[input columns, output columns]
pipeline.fit(model_dataset)
crossvalidation.fit(data set)
```

2. Any Other ML model to fill here :

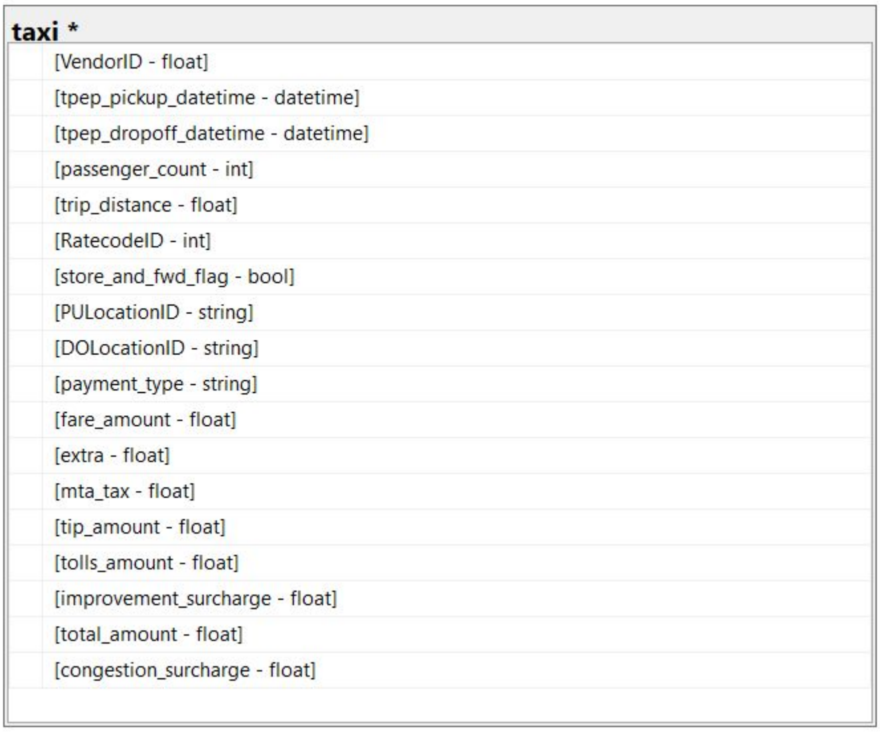
## 3.2 Application Components

Need to mention any Class/method names, third party pip install  etc here :

### 3.3 Activity



## 3.4 Schema Definition



| taxi * |
|---|
| [VendorID - float] |
| [tpep_pickup_datetime - datetime] |
| [tpep_dropoff_datetime - datetime] |
| [passenger_count - int] |
| [trip_distance - float] |
| [RatecodeID - int] |
| [store_and_fwd_flag - bool] |
| [PULocationID - string] |
| [DOLocationID - string] |
| [payment_type - string] |
| [fare_amount - float] |
| [extra - float] |
| [mta_tax - float] |
| [tip_amount - float] |
| [tolls_amount - float] |
| [improvement_surcharge - float] |
| [total_amount - float] |
| [congestion_surcharge - float] |

# 4. References, Abbreviations/Acronyms

https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
https://github.com/MingChen0919/learning-apache-spark/blob/master/notebooks/06-machine-learning/classification/random-forest-classification.ipynb