COMP309Sec006Group09 -Group Project Report

11/12/20

Group 09 Team members:

Liping Wu 300-958-061

# 1 Data exploration: a complete review and analysis of the dataset including:

## 1.1 Load and describe data elements (columns), provide descriptions & types, ranges and values of elements as appropriate.

import pandas as pd

import os

path = "D:/CentennialWu/2020Fall/COMP309Data/GroupProject2/"

filename = 'Bicycle\_Thefts.csv'

fullpath = os.path.join(path,filename)

print(fullpath)

data\_bicycle\_group09 = pd.read\_csv(fullpath)

data\_bicycle\_group09.columns.values

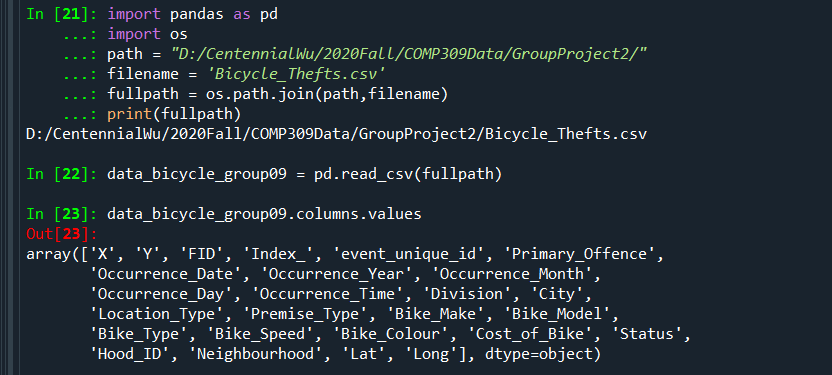
data\_bicycle\_group09.shape

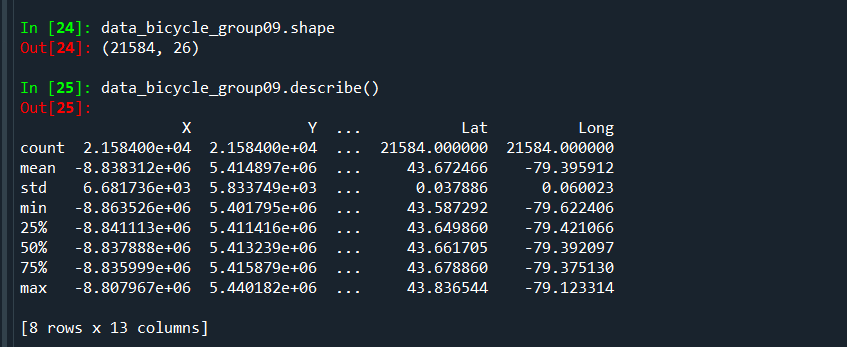
data\_bicycle\_group09.describe()

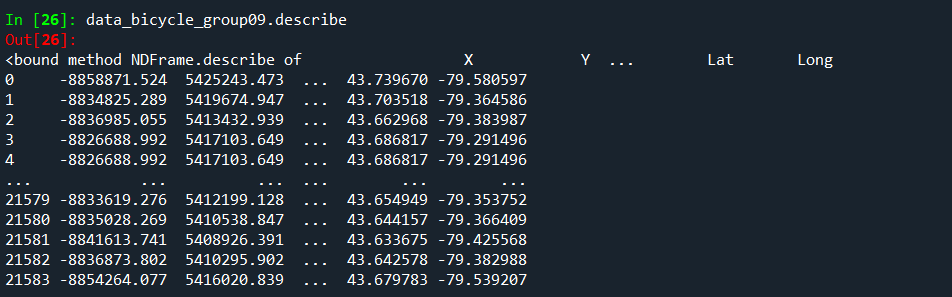
data\_bicycle\_group09.describe

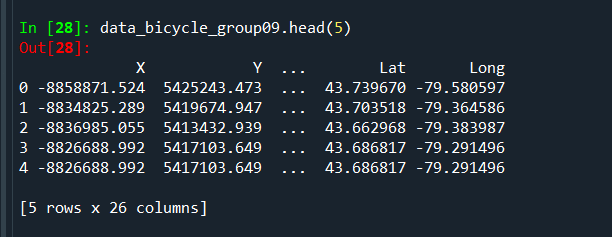
data\_bicycle\_group09.dtypes

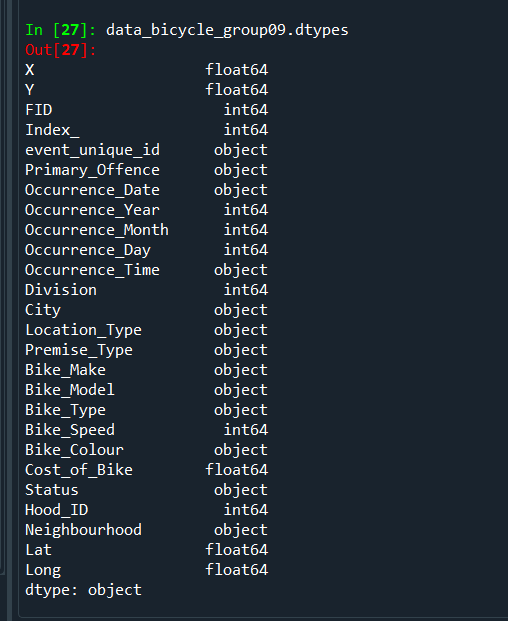
data\_bicycle\_group09.head(5)



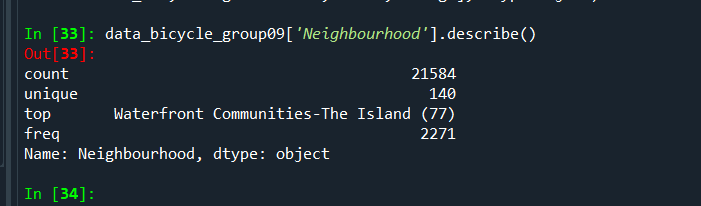


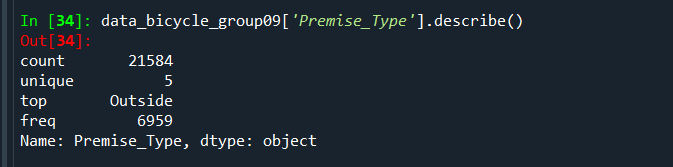


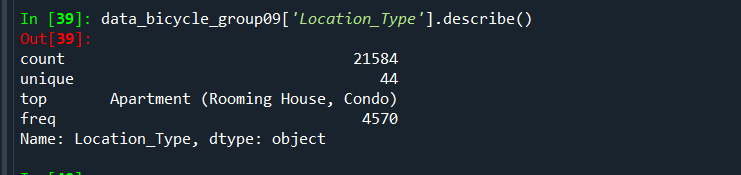


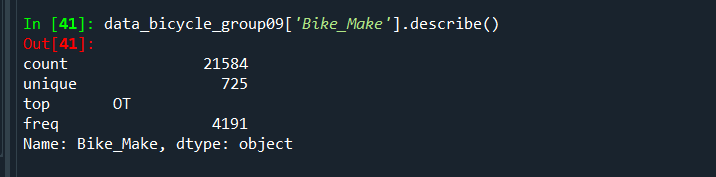


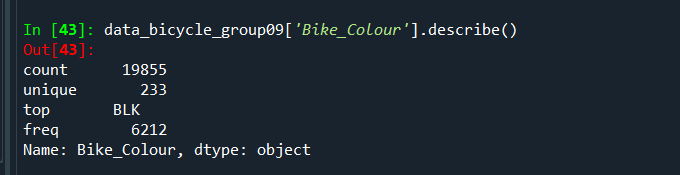
data\_bicycle\_group09['Neighbourhood'].describe()

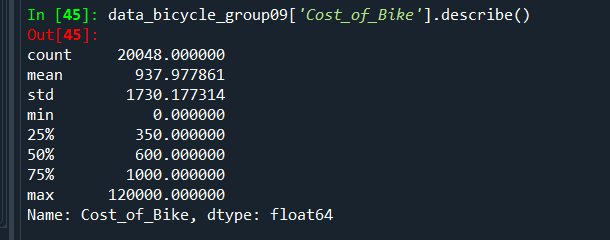


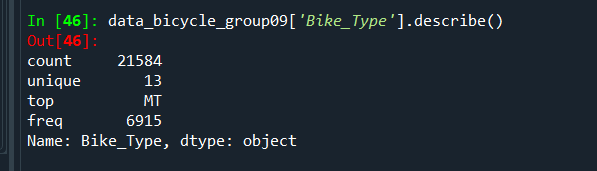


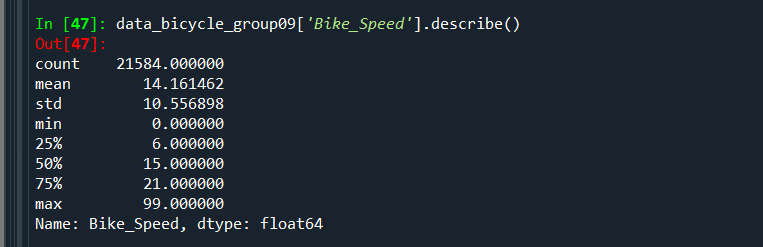












## 1.2 Statistical assessments including means, averages, correlations

## 1.3 Missing data evaluations – use pandas, NumPy and any other python packages

## 1.4 Graphs and visualizations – use pandas, matplotlib, seaborn, NumPy and any other python packages, you also can use power BI desktop.

# 2. Data modelling:

## 2.1 Data transformations – includes handling missing data, categorical data management, data normalization and standardizations as needed.

## 2.2 Feature selection – use pandas and sci-kit learn.

## 2.3 Train, Test data splitting – use NumPy, sci-kit learn.

## 2.4 Managing imbalanced classes if needed. Check here for info: https://elitedatascience.com/imbalanced-classes

# 3. Predictive model building

* 1. Use logistic regression and decision trees as a minimum– use scikit learn

# 4. Model scoring and evaluation

* 1. Present results as scores, confusion matrices and ROC - use sci-kit learn
  2. Select and recommend the best performing model

# 5. Deploying the model

* 1. Using flask framework arrange to turn your selected machine-learning model into an API.
  2. Using pickle module arrange for Serialization & Deserialization of your model.
  3. Build a client to test your model API service. Use the test data, which was not previously used to train the module. You can use simple Jinja HTML templates with or without Java script, REACT or any other technology but at minimum use POSTMAN Client API.

# 6. Prepare a report explaining your project and detailing all the assumptions, constraints you applied should have the following sections:

* + Executive summary (to be written once nearing the end of project work, should describe the problem/solution and key findings)
  + Overview of your solution(to be written once nearing the end of project work)
  + Data exploration and findings (dataset field descriptions, graphs, visualizations, tools and libraries used….etc.)
  + Feature selection (tools and techniques used, results of different combinations…etc.)
  + Data modeling (data cleaning strategy, results of data cleaning, data wrangling techniques, assumptions and constraints)
  + Model building (train/ test data, sampling, algorithms tested, results: confusion matrixes ...etc.)