Data Science Course

26 May 2023

Executive Summary

- Introduction
- Data collection and data wrangling methodology
- EDA and interactive visual analytics methodology
- Predictive analysis methodology
- EDA with visualization results
- EDA with SQL results
- Interactive map with Folium
- Plotly Dash dashboard
- Predictive analysis (classification)
- Conclusion

Introduction

The advent of powerful computers has made the use of data science relevant

Data processing is data reading (mainly from CSV file or SQL table), data wrangling (can be processed on Python or other programming language), data visualization and data analyses

Data collection and data wrangling methodology

For **data collection** we mainly use CSV files or SQL. Pandas library let us both way and lot of others (for example, JSON)

Data wrangling is data processing for made a good data. Firstly we need to eliminate empty data cells. We have some ways for it. For example we can remove lines with absent data cells or we can set to absent cells mean values.
Also we need to pay attention to data outliers. Here we have some ways (leave as is, put mean…). Also need to transform categorial variables to numeric (by get_dummies function)

EDA and interactive visual analytics methodology

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

There are four primary types of EDA:

- Univariate non-graphical
- Univariate graphical
- Multivariate non-graphical
- Multivariate graphical

For non-graphical analytics mainly we use Python or R
For graphical analytics we use graph libraries such as Matplotlib and Seaborn

Predictive analysis methodology

It’s prediction in some different algorithms. For example, Linear and Polynomial regression

In Linear regression we let what result variable Y depends from variables X linearity. It means what we can find a1, a2… So what Y = a0 + a1*X1 + a@*Y2 + … + an*Xn

In Polynomial regression we let what Y has polynomial dependance (Y = a0 + a1*X1 + a2*X1^2…). Rank is an one hyperparameter

When coefficients found we can predict Y from values X

After we can estimate accuracy by scope and / or RMSE

EDA with visualization results

Firstly we need to load libraries and load data (by pd.read_csv)

Now we can know rows and columns number (df.shape) and to see all categorial variables values (df[<field name>].value_counts()).

We can (if need) do data preparation. For example, we can very simple transform target numerical variable to categorial.
For example, if field ‘survival_status’ originally has values 1 or 0 we can transform it to ‘yes’ or ‘no’:
df['survival_status'] = df['survival_status'].map({1:"yes", 2:"no"})

Now we can build some visualizations for one and pairs variables.

General statistical analysis. We have very useful function df.describe(). This function shows us some statistical parameters for every numeric columns (count, mean, std, min, max, 25%, 50%, 75%)

EDA with SQL results

Firstly we need to load libraries and connect to SQL database

Now we can explore any table by select operator select * from <table name> order by <field> limit N

We can know rows count by count(*) SQL function

We can easy convert SQL select result to pandas DataFrame by function pd.read_sql

Also we can look SQL select result in Jupyter – just need to install library + put %sql + SQL operator or %%sql + multiline SQL operator

Interactive map with Folium

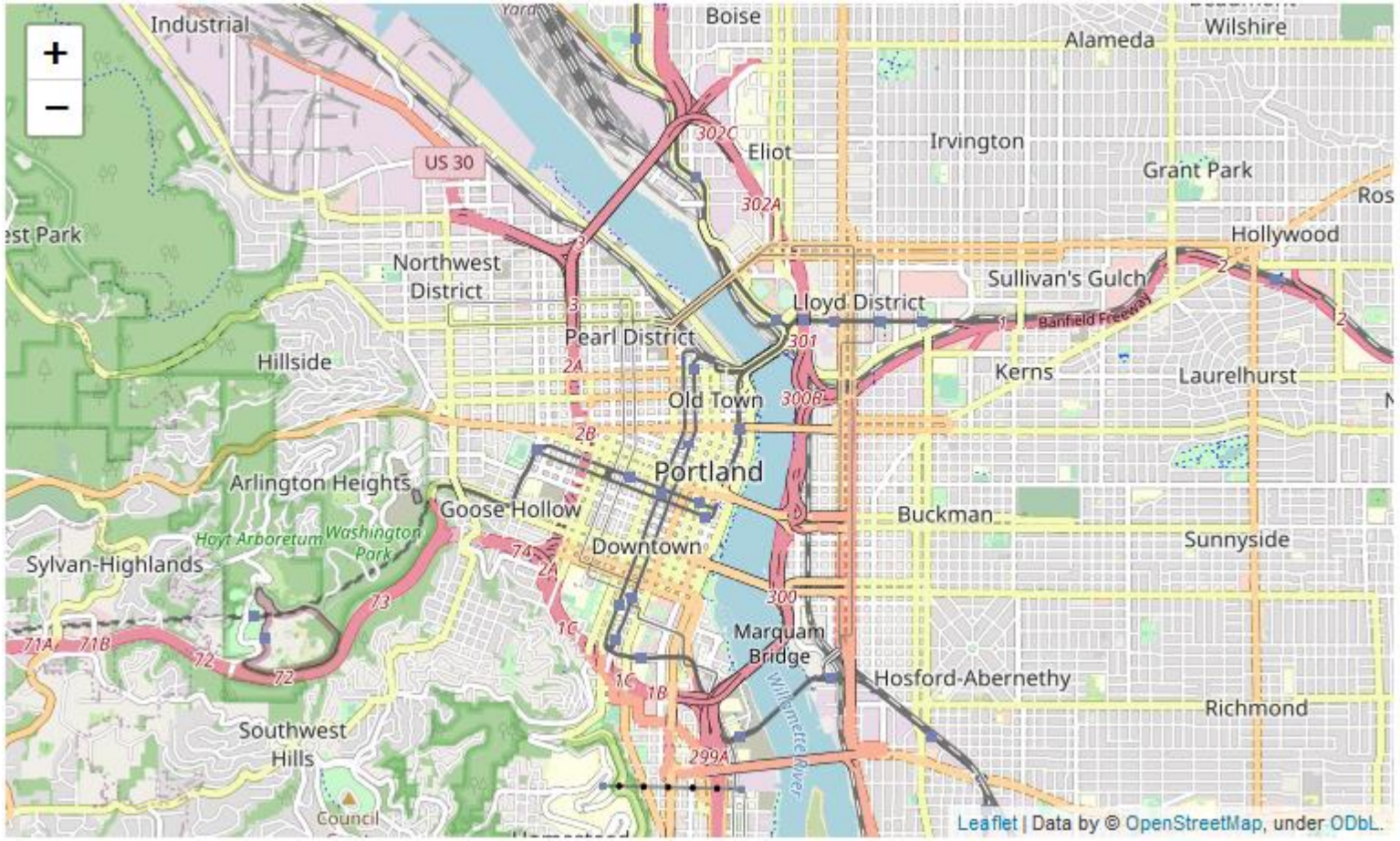
https://python-visualization.github.io/folium/

Python library for drawing interactive maps. Zoom level given on start. Possible to mark places points near places. Possible to draw lines and measure distances

Example:
import folium
folium.Map(location=[45.5236, -122.6750], zoom_start=13)
See picture on next slide

Markers example (picture does not provides)
m = folium.Map(location=[45.372, -121.6972], zoom_start=12, tiles="Stamen Terrain")
tooltip = "Click me!"
folium.Marker([45.3288, -121.6625], popup="Mt. Hood Meadows", tooltip=tooltip).add_to(m)
folium.Marker([45.3311, -121.7113], popup="Timberline Lodge", tooltip=tooltip).add_to(m)
m

Interactive map with Folium (1)



Plotly Dash dashboard

https://plotly.com/dash/

Python library for interactive graphic. Result seen in browser as HTML page. Provide report graph (pie, hist, scatter…) and controls (edit box, combo box, RangeSlider…)

Need to describe layout (app.layout = html.Div(children=[…])) and callbacks for control value changes

Predictive analysis (classification)

For classification we use model classes LogisticRegression, KNeighborsClassifier, DecisionTreeClassifier…

For quality control we use criteria accuracy_score,confusion_matrix,classification_report

If we have more from two classes we need to use special alrorithms: SoftMax Regression, One-vs-All or One-vs-One.
All these algorithms implement in model classes (parameter solver)

Conclusion

In course ‘Data Science’ we obtained these skills:

- To collect data and to wrangle it
- To analyze data from CSV and SQL by visualize and not
- To show interactive geographical map with Folium
- To create interactive dashboard with Plotly Dash library
- To predict data regression and classification (supervised machine learning)
- To clusterize data (unsupervised machine learning)