INFO 7390 –MIDTERM VERSION 1.0

Instructions:

- You can use either Python or R to work on this case.
- No sharing of work. You can work in your teams only
- You are expected to submit a report that summarizes the key steps in your implementation as a flow chart and submit fully functional code.
- Deadline: 10/03/2017 11.59 PM. Late submissions lose 10% points per day.
- Each team will have 15 minutes to present the 4 parts + 5 min Q&A on 10/04/2017

Working with large datasets and machine learning:

You work for XYZ consulting and you have been tasked to prove your data science skills by working on a real-world data set. We are going to expand the housing problem we worked in the class and work with real Zillow data.

Review https://www.continuum.io/blog/developer-blog/productionizing-and-deploying-data-science-projects **for motivation on why we are working on this assignment**

Review the following and the links I posted before to understand the problem:

- https://www.kaggle.com/c/zillow-prize-1
- https://www.kaggle.com/philippsp/exploratory-analysis-zillow
- https://www.kaggle.com/sudalairajkumar/simple-exploration-notebook-zillow-prize
- https://www.kaggle.com/captcalculator/a-very-extensive-zillow-exploratory-analysis

1. Data ingestion, EDA, Wrangling:

- Download the data from Zillow. (https://www.kaggle.com/c/zillow-prize-1)
- Create an IPYB notebook and Conduct an in-depth EDA (See below for ideas; Note: Your code should be original. You are welcome to use ideas but with attribution).
- Put together a note on what data cleansing is required for automation
- Clean up the data and take care of missing data values using a Python/R script
- Combine the 2016 and 2017 properties by adding an additional column for year
- Programmatically write the data to a S3 bucket named "ZillowData". This should be downloadable by anyone who has the links.
- Write a report documenting your data ingestion, wrangling steps.

2. Build a prediction model

- You are now expected to try out different prediction models to predict the log errors. Use RMS and MAPE as your measures and try
  - Multiple linear regression
  - Random forests
  - Neural networks
- Which model works best? Write a report discussing the different models you considered and which one works best. You should consider interpretability, computational overhead, accuracy measures, etc. in your discussion.

3. Model deployment

You are now expected to choose an enterprise platform to deploy your model. See https://docs.google.com/spreadsheets/d/17NqDJHdJtqfvgVHAl2_YplG9O8t3PvWB233YaByl2w8/edit?ts=59c65fd3#gid=558508381 **for the platform you have been assigned.**

- You could export your trained model from step 2 or redo the “best” model in the assigned platform. Your choice.
- You should advertise the JSON API to use to invoke the model
- Deploy the model and provide examples on how to invoke the api and how to interpret the results. Create a Jupyter notebook to illustrate how to use your REST API

4. Enhancing your REST API: Geospatial search

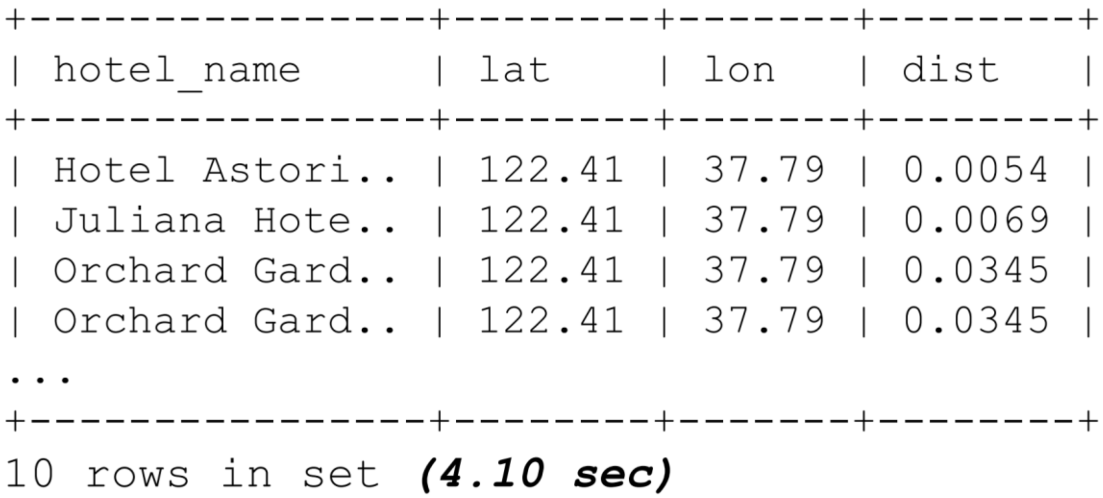
Note that each record has a Latitude and Longitude. Your goal is to create a REST API that given a Lat and Long, should return the top 10 closest homes.

Review these articles for the algorithm and how to use SQL to get these results:

- ✓ http://www.arubin.org/files/geo_search.pdf
- ✓ https://www.percona.com/blog/2014/06/19/using-udfs-for-geo-distance-search-in-mysql/
- ✓ https://www.percona.com/blog/2013/10/21/using-the-new-mysql-spatial-functions-5-6-for-geo-enabled-applications/

Tasks:

- Write a Jupyter notebook and illustrate using this REST API. For a given lat,long, you should present results something like this



```
+----------------+---------+-------+--------+
| hotel_name     | lat     | lon   | dist   |
+----------------+---------+-------+--------+
| Hotel Astori.. | 122.41  | 37.79 | 0.0054 |
| Juliana Hote.. | 122.41  | 37.79 | 0.0069 |
| Orchard Gard.. | 122.41  | 37.79 | 0.0345 |
| Orchard Gard.. | 122.41  | 37.79 | 0.0345 |
...
+----------------+---------+-------+--------+
10 rows in set (4.10 sec)
```

..

- Plot the results on a map. See https://www.kaggle.com/arjanso/kernel-density-estimation-for-predicting-logerror for ideas on plotting using scatter plots.