MCIS6273 Data Mining (Prof. Maull) / Fall 2018 / HW2

This assignment is worth up to 40 POINTS to your grade total if you complete it on time.

Points Possible	Due Date	Time Commitment (estimated)
40	Wednesday, December 05 @ Midnight	*up to* 6 hours

- **GRADING:** Grading will be aligned with the completeness of the objectives.
- **INDEPENDENT WORK:** Copying, cheating, plagiarism and academic dishonesty *are not tolerated* by Univerisity or course policy. Please see the syllabus for the full departmental and University statement on the academic code of honor.

OBJECTIVES

- improve on your homework 1 assignment, if necessary
- perform a clustering analysis using k-means
- understand the issues of data science at scale by listening to a current podcast about data mining and machine learning

WHAT TO TURN IN

You are being encouraged to turn the assignment in using the provided Jupyter Notebook. To do so, make a directory in your Lab environment called `homework/hw2`. Put all of your files in that directory. Then zip that directory, rename it with your name as the first part of the filename (e.g. `maull_hw2_files.zip`), then download it to your local machine, then upload the `.zip` to Blackboard.

If you do not know how to do this, please ask, or visit one of the many tutorials out there on the basics of using zip in Linux.

ASSIGNMENT TASKS

(50%) perform a clustering analysis using k-means

We talked about the simplicity and power of k-means algorithm and now we are going to use it to explore an interesting dataset.

The State of Delaware maintains a dataset of the causes of death from many different diseases. We going to explore this dataset in some interesting ways using unsupervised learning, namely clustering, to do some exploratory data analysis.

DATA PREPARATION

- You will want to filter the original data down to just those data points where you have just the data for `STATE OF RESIDENCE == DELAWARE`.

MAKE SURE TO SHOW ALL YOUR WORK IN THE NOTEBOOK SO YOU CAN RECEIVE PARTIAL CREDIT WHERE APPROPRIATE!

§ Load the CSV of the data from the URL:

- https://data.delaware.gov/api/views/nck5-dhqv/rows.csv?accessType=DOWNLOAD

Please provide a **bar plot** of all the diseases and their frequencies (over all years). You will need to use the `CAUSE OF DEATH` column once the data is loaded. Recall, you can simply do the loading by `pandas.read_csv=("DATA_URL")`. You are also free to download the CSV file to a local filesystem, just remember to include it in your ZIP so I can run your code correctly.

You may want to read up on `Series.value_counts()` to aggregate the data and ‘Series.plot(kind=‘bar’.

Turn in the bar plot showing all the diseases and answer:

- **Excluding “all other diseases”, what are the top 5 diseases?** (you can use `value_counts()[:6]` to quickly answer this or you can look at the plot)
- **What percentage of the top 10 causes of death were from cancers (i.e. *neoplasms*)?** Use `value_counts[:11].index.str.contains()` (see `Series.str.contains()`) to do this.

§ We want to filter the data further to just cancers (over all data, not just the top 10). Use `df['CAUSE OF DEATH'].str.contains('neoplas')` to do this.

Use `value_counts()` over `YEAR` to get total deaths per year.

- **Which year had the highest number of deaths?** Do not worrry about the rate of death, just the raw total.
- **Which year had the *highest rate* of cancer deaths? HINT**: If you have two Series objects of the same size `S1['column'].value_counts() / S2['column'].value_counts()` will return a Series with all the computations representing the rate we’re looking for. If S1 is the number of cancer deaths and S2 the total number of deaths filtered by year.

§ To prepare our data for clustering, we will need to turn all of our *categorical* variables into *numeric* one’s. The easiest way to do this is to use `Pandas.get_dummies(your_dataframe)`.

- **How many features are now in your dataframe?**

§ In class we talked about the fact that the k number of clusters needs to be determined *a priori* – that is you will need to know how many clusters beforehand to run the algorithm. To find the optimal k, we will use a method called the *silhouette score*.

Adapt the following code to compute the silhouette scores on *only* the dataset filtered by cancer deaths. For this part, we are not interested any of the other causes of death except cancer.

```python
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

Sum_of_squared_distances = []
K = range(2,15)
for k in K:
    km = KMeans(n_clusters=k, n_init=20)
    km = km.fit(YOUR_NEOPLASM_DATAFRAME_WITH_DUMMY_VARS)
    Sum_of_squared_distances.append(km.inertia_)

    silh_score = silhouette_score(YOUR_NEOPLASM_DATAFRAME_WITH_DUMMY_VARS, km.labels_)
    print("k = {} | silhouette_score = {}".format(k, silh_score))
```

The largest score is typically the k you go with. If $k = 2$ is your largest score, we will ignore that since 2 clusters is not usually an interesting number of clusters when dealing with a large set of data points.

NOTE: You may drop the columns `YEAR`, `COUNT`, `STATE OF RESIDENCE_DELAWARE` from your data since these do not have any bearing on our analysis.

- **What is the optimal k according the silhouette score?**
- **Do you see anything else interesting about the scores?**

§ Now that you have clusters, let’s find out which features dominate them.

- **Print a report of the clusters and their top dominant features adapting the code below.** Note that the k-means algorithm returns the cluster centers for each cluster, hence in that center, we will use the dominant features as the *representative features* for that cluster. For the sake of this exercise, we will use 0.395 as the threshold of an interesting feature. Recall also that all of the feature values are 0 or 1 and hence we have cluster centroid values that range between 0 and 1.0.

```
optimal_k = THE_OPTIMAL_SILH_K

km = KMeans(n_clusters=optimal_k, n_init=150)
km = km.fit(YOUR_NEOPLASM_DATAFRAME_WITH_DUMMY_VARS)

for i in range(0, optimal_k):
    l = list(zip(YOUR_NEOPLASM_DATAFRAME_WITH_DUMMY_VARS.columns, \
                km.cluster_centers_[i]))
    l.sort(key=lambda x: x[1], reverse=True)

    print('CLUSTER : {}\n'.format(i))
    for attr, val in l[:15]:
        if val > 0.395: # we are going to stop printing values < 0.395
            print('\t{} : {}\n'.format(attr, val))
```

(50%) understand the issues of data science at scale by listening to a current podcast about data mining and machine learning

Throughout the course, we have talk about various algorithms, methods and tools for doing data mining. We've also talked about "Data Mining" being a part of the over all field of "Data Science"

You already know that there is a lot of talk about "Data Science" being one of the hottest (if not *the hottest*) tech jobs out there. It is certainly one of the fastest growing jobs categories in tech, spanning many different domains, but the breadth of the discipline is large and has many use cases and impacts in many different areas.

We are going to listen to a real-word account of Data Science "in the wild" and how one company is dealing with the demands of their customers to bring data science (i.e. data mining, analytics, visualization and machine learning) into their enterprise product offerings.

§ Listen to the O'Reilley Data Show podcast episode "Building tools for enterprise data science" (Nov. 21, 2018) (you can also find it on Soundcloud here https://soundcloud.com/oreilly-radar/building-tools-for-enterprise-data-science), which is an interview with the VP of Data Science and Engineering, Vitaly Gordon of Salesforce.com.

- Why did Salesforce decide to build their own internal platform, Einstein, for delivering data science at scale to their customers?
- What estimate does Vitaly give for the amount of data that sits in "custom objects" or "custom tables" in their customer data?
- What problem does TransmogrifAI try to solve?
- What kind of data does the platform work with (e.g. structured or unstructured)?
- Give an example of the kind of metadata Vitaly mentions is being used to perform feature engineering?
- Why is domain expertise so important when dealing with data pipelines (according to Vitaly)?
- What three things did Vitaly identify as having a higher impact on model accuracy than the models themselves?
- What does the "Prediction Builder" tool do for customers?
- Why was it important to introduce tools used to help customers monitor models?
- What does Vitaly predict will be the future role of data scientists as tool automation continues in AI and machine learning (e.g. use his comparison with software engineering)?