PrecisionLender – Data Science Homework (Ryne Schultz)

1. **Where would you start? What information do you wish you had? In the absence of perfect knowledge and access to perfect data, what can you do to make a first attempt?**

   Immediately, upon reading the prompt, I thought of the different forms of classification methods available in data science, since the question "a bank like us" is really asking "for banks in our class/category". Since there was no classification metric available in the public data, I also knew that this was going to be an *unsupervised* classification problem, a subset of classification that draws inferences from datasets *without* labeled responses.

   Clustering algorithms form groupings or classes in such a way that data within a given class have a higher measure of similarity than data in the other classes. There are many clustering algorithms (hierarchical, gaussian mixture models, self-organizing maps utilizing neural networks, etc.), but for my first attempt at the model I decided to use the most common method, K-means clustering.

   For these types of clustering problems, it would be ideal to have data that contains features/dimensions that highly differentiate the banks. For example, if we had a hypothetical dataset containing information about cars, a feature such as "# of Tires" wouldn't be as helpful in classifying cars as, say, "Ground Clearance"; the number of tires, generally speaking, will not vary as much as how high off the ground the car is (think of the difference between a sports car – which is very low off the ground – versus a truck – which is very high off the ground). Thus, features containing bank characteristics that have high variance are of particular interest.

   However, we do not have the ability to pick our data's features, so the next best thing is to include *all* of the features, perhaps applying some Principal Component Analysis to pare down the feature count. In essence, our model should be agnostic as to which features are the most important – opting to err on the side of too much information versus too little. Using all of these various features, the model will then create a "fingerprint" for each bank, grouping similar banks together.

2. **Now do it. Build a model that classifies banks. Explain the steps you took and the decisions you made along the way, as if you were leaving notes for the next person who might tackle this problem.**

The first step I took in building the model was to make the information usable by converting the data from a tab-delimited text file to a tidy DataFrame. There were several data files available on the FFIEC's website but only one of them contained bank-level data – "Call Reports -- Balance Sheet, Income Statement, Past Due -- Four Periods". This dataset contained hundreds of bank related fields (Total Assets, Total Deposits, Interest Income, etc.) by bank, which would allow me to cluster the banks.

The data was actually split between two files, so both files had to be converted to usable DataFrames and then combined. Additionally, the naming conventions (unique identifier codes) weren't particularly useful, so the more helpful names contained in the first row were substituted for the original column names.

Once a combined DataFrame was created I then filtered on the most recent reporting period, imputed the null values first by pulling in any non-null values from the other data file and finally by setting any remaining null values to 0, and then separated out the bank balance sheet features from the ID columns.

Finally, there were several columns that had the same column name. These were columns that had slightly different original ID column headers, implying different types of the same category (for example there were two different IDs associated with Total Assets, implying two slightly different types of asset values, both rolling up under the same broader "Total Asset" category). To remove these duplicate columns, I combined them by summing their values using a groupby function. Thus, the broader category was retained (though the sub-categories were not). The end result was a set of unique bank features that I could input into the clustering model.

The dataset contained 338 columns, while only containing 5,456 observations, implying that there was a potential dimensionality problem. The Curse of Dimensionality is a phenomenon in mathematics such that when dimensionality increases (i.e. when there are a large number of features in a dataset), the volume of the space also increases, but at such a high rate that the amount of data needed to support a statistically reliable result grows *exponentially*. This is of particular concern in clustering models, since in high dimensional datasets all observations can appear to be sparse and dissimilar, dramatically decreasing the efficiency of the model.

Thus, I need to apply a method of dimensionality reduction. One popular technique is Principal Component Analysis (PCA). When using classification algorithms, not all features will contribute equally to the generalizability of the model. Irrelevant and correlated attributes often decrease performance. PCA would allow me to set a certain number of "best" features – "best" being defined as the minimum number of features that capture most of the variance in the data.

First, however, I needed to normalize the data by rescaling the data to a range between 0 and 1. Though the different features were largely of the same unit (US Dollars), the relative sizes were vastly different. For example, some features contained values on the order of tens of thousands of dollars while others, like total assets, contained values in the tens of billions. To perform a reasonable covariance analysis, we need to normalize this data, putting all attributes in the same order of measurement (i.e. a range between 0 and 1). This will improve the maximization of the variance for each component that the PCA needs in order to perform its matrix operations in the optimal way.

After standardizing the data, I was then able to calculate the explained variance for each feature and plotted the result. The plot indicated that by selecting the top 50 features I could preserve around 98% of the total variance of the data. Thus, I was able to reduce the dimensionality of the dataset by around 85%.

Now that I had my lower dimensional dataset, the next step was to determine the optimal number of clusters to utilize in my K-means clustering model. This was determined by creating a Scree plot. A Scree plot plots the reduction in the within cluster sum of squares (often referred to as the "distortion") as the number K (the number of clusters) increases. The within cluster sum of squares is the sum of the squared deviations from each observation and the cluster centroid, whereby a more compact cluster is one that has a smaller sum of squares. In order to determine the optimal K value in my K-means clustering model, I must choose the "elbow" of the Scree plot, the K value at which the distortion decreases at the greatest rate, before leveling out as K increases.

When I looked at the Scree plot, I saw that this K value is 2 – suggesting that the optimal number of clusters is 2. However, I selected a number slightly larger (4) largely to see how the clusters are distributed in terms of the number of banks in each cluster.

After recombining the ID columns, the unscaled data, and the new cluster values into a new DataFrame I then looked at how the model was clustering the banks across various metrics. For example, when I looked at the clustering across total assets, I saw that cluster 0 groups banks with smaller assets together, cluster 1 groups banks with larger asset values, while clusters 2 and 3 comprise banks with *extremely* large asset values. Indeed, when I looked at how skewed the clusters were, it seems to suggest that the model was having difficulty appropriately clustering the banks due to a group of outlier banks, comprised of Citibank, Wells Fargo, Bank of America, and JP Morgan Chase. This would suggest that removing these outliers could greatly improve the model.

Either way, I now had a model that could cluster bank clients and analyze key bank metrics by cluster. Therefore, when a current or prospective client asks a question about

"banks like them", I could answer those questions by looking at which cluster the bank belongs to and then looking at the within cluster average for the metric in question.

3. **Now that you're done, suppose a co-worker is eager to use your results & ideas in our business, starting immediately. What would you advise and why?**

   The first thing I would advise is that this model is still a work in progress, and that currently, the answers it provides are tentative at best. This is because of the outlier problem I noticed when I analyzed the clusters on certain metrics. However, after having more time to remove the outliers and rerun the model I would then say that we could use the model on a provisional basis for specific questions from clients. However, I wouldn't advise productionizing the model in any way just yet.

4. **When should your solution NOT be used?**

   The model would not be useful for clients that are not represented in the public data, unless they were able to provide their information for every single metric in the dataset – at which point we could append their data to the DataFrame and rerun the model with them in it. The model looks at a very specific set of metrics and if we had only a few metrics about a prospective client I would not feel comfortable using the model to say anything definitive. However, to the extent that we have complete or at least mostly complete data in the underlying dataset for a specific client I would feel comfortable using the model in a bespoke way to answer specific questions.

5. **If you had more time and resources, what would you do next, to improve or refine your work?**

   As mentioned, I would remove the outliers and rerun the model to see how the model improves its ability to cluster the non-outlier banks. I might also try several different clustering methods to see if there is any improvement in the reduction in the within cluster sum of squares. I would be particularly interested in utilizing a neural net on the data to see if such a technique would be better at clustering the banks.

   I would also like to determine which of the features are represented in the top 50 features that the PCA model selected, and which were inputted into the K-means clustering model. It would be very instructive to determine which features had the greatest effect on the model.

   Finally, I could refine my model by working with other team members who have greater domain knowledge and who understand the nuances of bank balance sheet metrics. For example, instead of taking an agnostic view to which features are most important I could work with a member of the team to determine the most relevant bank features

and to utilize those features exclusively. In this way, I could improve the relevancy of my model by utilizing the strengths of the team.

6. **Tell us what you think of this homework assignment. What would you do differently, if you were designing it?**

   I thought that this project was incredibly engaging and of great utility since the amount of time spent on data wrangling versus model results analysis is accurate to real world scenarios. In general, a lot more time is spent on trying to wrangle data into a usable format (upwards of 80% of the time on a given project) and that was certainly the case with this project. I've worked on several projects through the course I am taking with Springboard but most of the data sources are already cleaned up and ready to use. This homework assignment provided me with the opportunity to tackle a much more advanced data wrangling project, and I learned a lot from it!

   If I were to design the project I wouldn't change much, but I would probably provide greater clarity about which data files to utilize. I spent an inordinate amount of time discerning which data files were the relevant ones – though that is also a part of real-world data science. Other than that, the data was relevant to the core competencies of the role and it provided an extremely realistic example of something a new data scientist would do for the team.