**Capstone Project 1 Report: Crunchbase dataset**

**Predicting the success of the startup.**

The crunchbase dataset contains information about several startups across the world. The startups are based out in several different industries like medicare, tech and finance. The problem statement is to determine whether the startups where successful or not taking several features into consideration.

The CrunchBase dataset, provides us with data that is pretty biased. In order to be in the CrunchBase database, a company must already have become newsworthy. Normally that means raising a significant round of funding or graduating from a prominent incubator. We will take this bias into account when performing our analysis. Simply raising one round of funding will not be considered a successful outcome. If startup raises one round of funding it cannot be considered to be successful.

We will consider a company to be a success if it IPOs or if it is acquired. We will also consider a startup a success if it has raised several rounds of funding. How many? Well, since all the startups in the CrunchBase database have already raised one round, we will pretend we are making a second round investment. We will also consider a startup a success if it has raised several rounds of funding. How many? Well, since all the startups in the CrunchBase database have already raised one round, we will pretend we are making a second round investment. Any companies that raise a third round will be considered successful as they have proven that other investors are interested in investing in this company after us.

**Data Collection:**

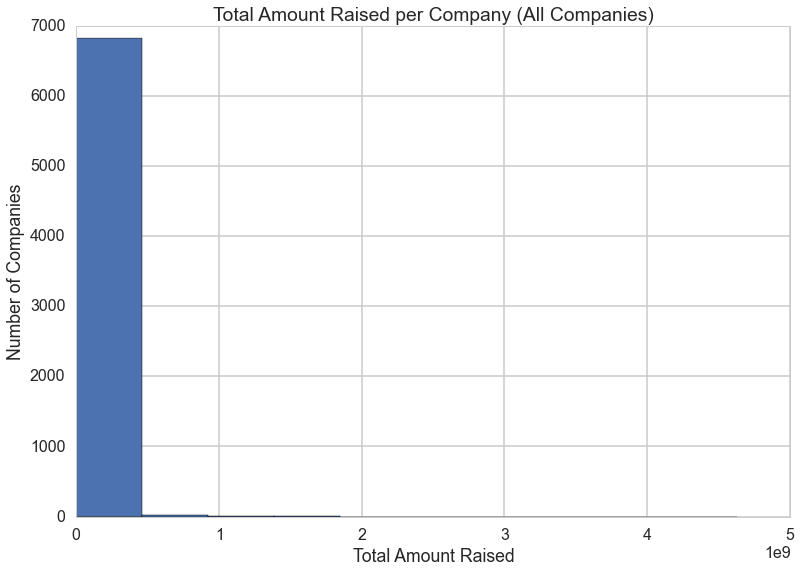
The data from the crunchbase dataset is obtained by crunchbase API which sends the request, collects data and is stored in the json format. Then the json file is imported to the Jupyter notebook and is stored in a dataframe in the form of a table.

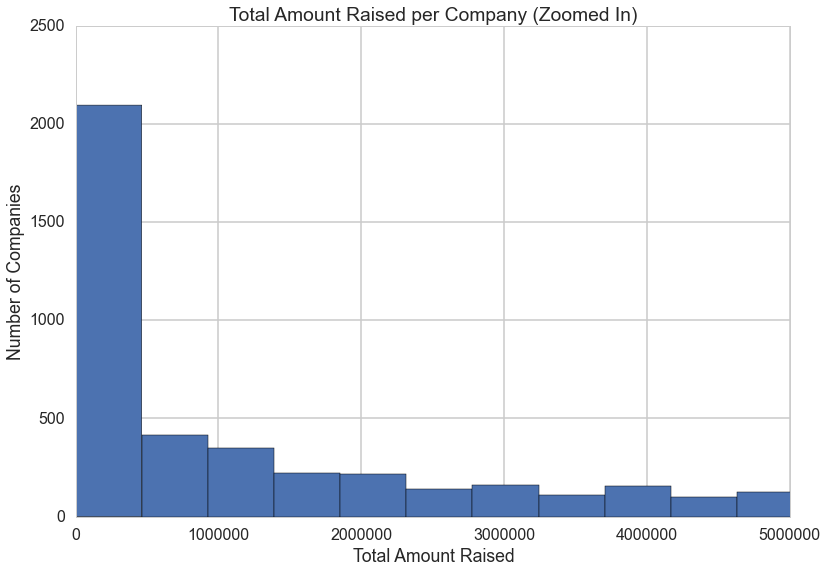
**Data Cleaning:**

The json format is converted into a dataframe and the description text with expressions, punctuations, adjectives and nouns are classified. Some do not have descriptions and they are given the value 0. Then indicator columns are attached to different categories like cities, investors and headquarters. We also gain information like funding round, funding amount mean, went for ipo, was acquired, founder names and number of founders, number of board members.

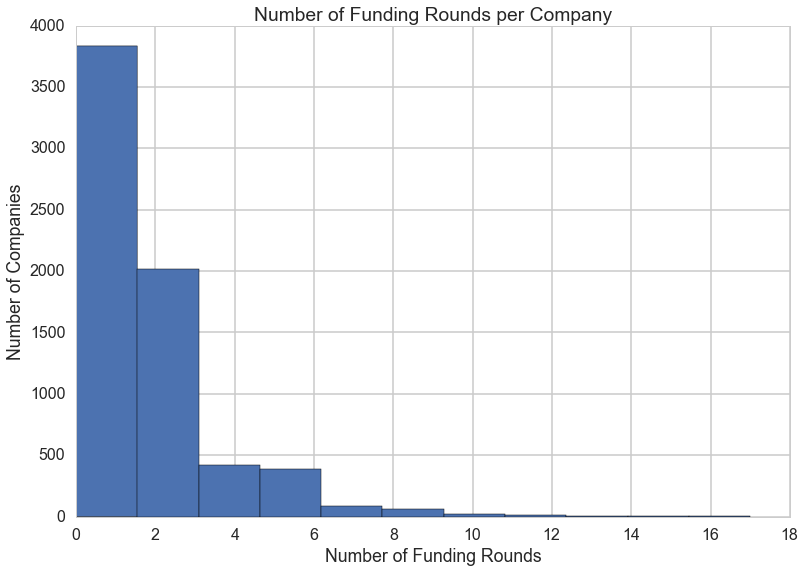
**Exploratory Data Analysis:**

Clearly Acquired and IPOed are good indicators, but funding rounds appears to be a prerequisite to making it into the CrunchBase system. Perhaps we can be more discerning in determining which companies that raised funds are considered successful, maybe a specific amount or number of rounds. So, funding rounds are taken into consideration and especially the amount of funds raised.





Total funds raised by the companies in the crunchbase dataset.



Number of rounds each company raised.

Again, we see that most companies are clustered near the bottom of our range with 0-3 rounds raised. 4 or more rounds raised is not very common. Perhaps we can use that as an indicator of success, seeing as how it appears to be less sparse than the total amount raised column. Perhaps this is due to the prevalence of companies and investors announcing that they raised a round, but not disclosing the exact amounts of that round. 1001 companies have raised at least 4 rounds.

This may seem like a high success rate when compared to other metrics about start-up success, but remember that the companies that make it into CrunchBase are already quite a selective bunch. As we saw above, most companies in CrunchBase have already raised at least 1 round, so this does not seem like such a high success rate considering this is already a pretty select crowd of companies.

After performing analysis on the success column, we can see that there is a wide range in the success percentage based on the category from 49% for Semiconductor companies to 10% for Consulting companies.

Then we will consider the cities that have a massive range in the success rate. Again, we see a massive range in the success rate, from 61% in Santa Clara down to 20% in Paris.

**Splitting the data:**

Let's split our dataset into training, validation and testing sets. We need to clean up the columns first and separate the success column from the rest of the data. We'll also remove any columns that the success column is dependent on (went\_ipo, was\_acquired, etc....).

Then we'll split out the testing data from the training data.

We'll then perform a second split on the training data to get the validation data.

The unnecessary columns are dropped and the success column is split from the training data.

**Baseline Model:**

Now we will attempt to find a baseline to improve upon. A basic baseline would be assuming that all companies fail or all companies succeed. Let's take a look at what those two assumptions would yield.

Accuracy of the model assuming all succeed: 0.33527696793  
Accuracy of the model assuming all fail: 0.66472303207

So the basic baseline to beat is 66.5%, which we get if we just assume all companies will fail. That's not so hot. Throughout the rest of this notebook we will be trying to create a predictive model to beat it

Also recall: we are looking to invest in companies that we think will be successful. Although we are trying to predict the success of a company, it's a little more nuanced - we are more so trying to find the next successful company. Remember that our models will consider a success companies with >50% probability threshold of being successful. We should also look at adjusted thresholds according to what is more costly to us - false positives or false negatives?

A false positive would potentially lead to us funding a startup that will eventually fail. A false negative would potentially lead to us missing out on a startup that's successful.

To see what is more costly, we must look at our baselines from the perspective of utility - we need to come up with some profit assumption based on some cost. This is hard to do as the majority of our "success" companies will either have been acquired or have gotten more than 4 rounds of funding - the true valuation and therefore return on an investment in this company isn't a known quantity.

The cost of a false negative is 6x more than a false positive. That means in our analysis, we should be less strict on the probability threshold of a success. When looking at it from a profit perspective, as expected given our utility matrix we are trying to beat the seed all startups baseline. We'd definitely want to beat this and be more selective. We will now start looking at predictive models in an attempt to beat the baseline. We will first focus on beating the all-fails 66% baseline.

**Machine Learning Model:**

**K-Nearest Neighbors**

We will perform a K-Nearest Neighbors classification to predict if a company will be successful or not based upon the training data. The two categories into which we will be classifying our data are successful or not successful.

Let's graph the n\_neighbors by accuracy, but with a balanced training set. This is a useful method to use in data analysis when you have a skewed data set.Here we will subsample our training set to build a balanced model where half of the data set is made of successful startups, and the other half is made of unsuccessful startups.

Since we have fewer successful companies than unsuccessful ones, we will downsample the unsuccessful ones to have the same number as successful. Now that we have the right number of unsuccessful companies, we will recombined them with our successful companies to make our balanced dataset. We'll start by iteratively running the balanced training set with a wide range of number of neighbors to find the best neighborhood of neighbors.

Best neighbors was 30 with max accuracy of 0.722852512156.