TIM 245 - Data Mining: Homework #1

Due: *May 6, 2017*

Instructor: Tyler Munger

Panos Karagiannis
ID: -

Contents

Problem 1: Exploratory Data Analysis in Excel

Open the csv dataset in Excel and identify 2-3 numerical attributes that you think would be potentially be good predictors if a movie will receive an academy award. Then answer the following questions:

1. Provide a brief (1-2 sentence) explanation of each selected attribute and the rationale of why you think it would be a good predictor.
2. Use a combination of visual and quantitative tools to answer the following questions for each selected attribute:
   (a) What is the typical value (central tendency)?
   (b) What is the uncertainty (spread) for a typical value?
   (c) What is a good distributional fit for the data (symmetric, skewed, long-tailed)?
   (d) Does the attribute affect other attributes (correlation)?
   (e) Does the attribute contain outliers (extreme values)?

It may be useful to format the answers as a table. Include any relevant plots and descriptive statistics in an appendix section.

Answer:

1. After looking at the *imdb* dataset the 3 numerical attributes that I think could potentially be good predictors are:
   (a) `movie_facebook_likes` (column AA):
   The number of "likes" that a particular movie has gathered is a good indication of whether the movie has a strong public appeal. Therefore, we could assume that a large number of "likes" is evidence of the good quality of a movie.
   (b) `imdb_score` (column Y):
   Similarly to the number of "likes", the `imdb_score` could function as a measure of the quality of a movie. Moreover, the `imdb_score` is calculated using various filters and a weighted vote average rather than raw data averages (source: imdb website), Therefore, this measure could be even more reliable than the number of Facebook "likes".
   (c) `budget` (column V):
   The budget of a movie could implicitly inform us about the quality of actors, directors and equipment used. These factors play an important role in the overall quality of a movie.

   For each of the attributed chosen in part (1) follow the answers to question 2:

   `movie_facebook_likes`:

   (a) Typical Value: Calculating the mean for this data using Excel we get that the average number of "likes" is 7527

(b) SPREAD: The standard deviation (sample) for this data is $19,362$. Since significant outliers exist in this attribute a better measure of dispersion is $mad = 163$

(c) DISTRIBUTIONAL FIT: Drawing a histogram of the data we see that the distribution looks skewed to the right (i.e. few movies have a lot of likes)

(d) CORRELATION: In calculating the Pearson coefficient between `movie_facebook_likes` and every other numerical attribute of this dataset we see that it is mostly positively correlated with `num_voted_users` with correlation coefficient $\sigma = 0.5397$, whereas, it is not negatively corelated with any other attribute.

(e) OUTLIERS: There exist many outliers which is evident from the magnitude of the measures of dispersion ($sd = 19362$, $mad = 163$).

`imdb_score`:

(a) TYPICAL VALUE: Calculating the mean for this data and ignoring incomplete entries we get that the average score is approximately 6.43

(b) SPREAD: The standard deviation (sample) for this data is 1.117

(c) DISTRIBUTIONAL FIT: Drawing a histogram of the data we see that the distribution looks to be symmetric centered around the mean.

(d) CORRELATION: In calculating the Pearson coefficient between `imdb_score` and every other numerical attribute of this dataset we see that this attribute is not strongly correlated with any other attribute. `imdb_score` exhibits the strongest positive correlation with `num_voted_users` ($\sigma = 0.42$) and the strongest negative correlation with `facenumber_in_poster` ($\sigma = -0.06$).

(e) OUTLIERS: Looking at the histogram as well as at the standard deviation ($sd = 1.117$) there does not seem to be significant outliers for this attribute. Here we don't need to look at the *mad* measure since our data are relatively symmetric around the mean.

`budget`:

(a) TYPICAL VALUE: Calculating the mean for this data and ignoring incomplete entries we get that the average budget is approximately $39,809,665.9916$

(b) SPREAD: The standard deviation (sample) for this data is $206,468,209.615$. Since significant outliers exist in this attribute a better measure of dispersion is $mad = 16,000,000$.

(c) DISTRIBUTIONAL FIT: Drawing a histogram of the data we see that the distribution looks to be similar to the number of Facebook likes, therefore it is very skewed to the right.

(d) CORRELATION: In calculating the Pearson coefficient between `imdb_score` and every other numerical attribute of this dataset we see that this attribute is not strongly correlated with any other attribute. `budget` exhibits the strongest positive correlation with `num_critic_review` ($\sigma = 0.119$) and the strongest negative correlation with `facenumber_in_poster` ($\sigma = -0.01$).

(e) Outliers: Looking at the histogram as well as at the standard deviation ($sd = 206468209$) and the median absolute deviation ($mad = 16000000$) there seem to be significant outliers for this attribute.

$\Longrightarrow$ All histograms are included in the Appendix.

Problem 2: Data Cleaning in Open-Refine

1. Does the dataset contain missing values? What approach do you recommend for handling missing values?
2. What attribute, or combination of attributes, can be used as a unique identifier? Does the dataset contain duplicates?
3. Does the dataset contain outliers? What approach do you recommend for handling outliers?
4. Compare and contrast the following clustering methods. For each method provide examples from the IMDB dataset of where the method correctly identifies inconsistencies and where the method does not work.
   (a) Key Collision / Fingerprint
   (b) Nearest Neighbor / Levenshtein Distance

Answer:

1. By looking at the entries for the selected attributes we realize that there exist many missing values. There are many different approaches that we could follow to handle such values, but in this case we can just ignore them because our data set is large, therefore we are discarding only a small fraction of data. For example, using OpenRefine we see that the number of entries whose budget is missing is only 434 which is approximately 8% of the total data. Similarly the fraction of missing imdb scores and facebook likes is 0.1% and 0.02%, respectively.
2. First we check to see if the combination of attributes `movie_facebook_likes` and `imdb_score` and `budget` can function as a unique identifier but unfortunately we see that there exist in total 462 entries having all 3 attributes common. Therefore we check to see whether the movie title is unique among all movies, nevertheless, it turns out that our dataset contains 117 movie title duplicates. Therefore, we first remove the duplicate values of our attributes and then use as a unique identifier the combination of attributes `movie_title` and `movie_facebook_likes`.

   The number of total duplicate rows in the dataset is 40. Since this is a small number of duplicates we can just remove them while only keeping the duplicates that we encountered first.
3. Based on our analysis in Question 1, we know that there exist outliers especially in the attributes `movie_facebook_likes` and `budget`. Therefore for all 3 of our attributes we

define $Q_0 = 5\%$ quantile and $Q_1 = 95\%$ quantile and then take the attributes which fall inside these quantiles. Following these process we only keep the following values:

$$0 \leq \texttt{movie\_facebook\_likes} \leq 47,350$$

$$4.6 \leq \texttt{imdb\_score} \leq 8$$

$$1,165,000 \leq \texttt{budget} \leq 140,000,000$$

4. First, after filtering (following steps 1-3) my new dataset consists of 2929 total entries. In comparing the two methods on the *filtered* IMDB dataset we notice that *Nearest Neighbor / Levenshtein Distance* performs a bit better than the simpler *Key Collisions/ Fingerprint* method. More precisely, by trying the *Key Collisions/Fingerprint* on `content_rating` we see that it clusters together categories like { `PG-13, PG_13, pg13`} nevertheless categories like { `13 pg, pg 13`} are separately clustered. In those cases we had to intervene and make both categories clustered together.

   Similarly, *Nearest Neighbor / Levenshtein Distance* produced useful results for `content_rating`. For example, this method created two seperate clusters for the ratings of {`Restricted, restricteed`} , {`restricted, Restricted`} where again we had to intervene and make both categories clustered together. Nevertheless, this method provided better results in other nominal values such as `movie_title` and `director_name`.

Problem 3: Data Integration and Transformation in Open-Refine

1. Which normalization method did you use for your selected numerical attributes: min-max or z-score? Explain the reason for your selection.

2. Examine one of the Wikidata film entries from the IMDB dataset (e.g. `https://www.wikidata.org/wiki/Q103474`) What other attributes in the Wikidata knowledge base could potentially be useful in predicting if a movie will win an academy award?

Extra credit:
Add the additional the attributes from your answer above to the IMDB dataset.

Answer:

1. To normalize my data I chose the z-score normalization. In computing the z-score normalization I found it useful to recompute the mean and standard deviation of reach attribute since the removal of the outliers *(Problem 2 Step 3)* reduced dramatically the standard deviation. The reason why I chose to use the z-score normalization is because it is more robust to outliers. Even if I have filtered the most significant outliers of my dataset for the selected attributes, there still exist outliers that could affect the linear normalization method of min-max.

2. In reviewing the Wikidata film entries from the IMDB dataset we see that the `based on` attribute could be useful in predicting if a movie will win an academy award. For example, if a movie is based on an award-winning novel then the chances of people and critics liking that movie are much higher. Moreover, another attribute that could allow us to make more accurate predictions could be the `depicts` attribute. For example, it could be the case that the majority of films that win academy awards depict "space exploration" or "artificial intelligence" therefore such information would be very insightful.

Problem 4: K-Nearest Neighbors Classification in Weka

1. What is the predictive accuracy of the kNN model? Do you think this model is good or bad? (We will cover model evaluation formally later in the quarter but I want to see how you think about model evaluation).
2. What is the interpretation of the model, i.e. given the predictive accuracy what can we say about the relationship between the attributes and the target? For example, does the model support the hypothesis that the Academy Awards is a popularity contest?
3. What are some of the applications of the predictive model? For example, who would be interested in this predictive model, e.g. directors, studios, actors, producers? How could they potentially use the model?
4. What impact did the data cleaning from Problem 2 have on the predictive model? For example, what effect do missing values, duplicates, outliers, and inconsistencies have in the kNN model?
5. What impact did the data transformation from Problem 3 have on the predictive model? For example, how does normalization affect the kNN model?
6. What are some of the potential issues with the classification model? What are some possible improvements that could be made to address these issues?

   Extra credit:
   Implement the improvements your suggested improvements above and re-evaluate the model. Do the improvements make a difference in the model?s prediction accuracy? Explain why or why not?

Answer:

1. In our KNN classifier, we trained for parameter values $K = 1$, using the Euclidean distance and testing using 10-fold cross validation. The accuracy of the classifiers is:

$$Accuracy = 90.4051\% \pm 5.1992\%$$

   ,where 90% reflects the mean value among all 10 training samples and 5% reflects their standard deviation. I believe that whether the model is good or bad depends on the specific application. For example this model is good if we want to utilize it as evidence that the Academy Awards are strongly dependent on these 3 attributes. Nevertheless, if we were to use this model to bet a lot of money on a specific movie winning the Academy Award, then the precision is very low.

2. Based on our predictive accuracy it seems that our model is able to predict fairly accurately whether a movie will win the Academy Award based on these 3 attributes. Since the attributes we used are `movie_facebook_likes`, `imdb_score` and `budget`, the model indeed verifies the hypothesis that the Academy Awards is a popularity contest.

3. A potential use of the model would be for producers to be able to sign contracts in advance with the actors/directors participating in the movie. After a movie has won the Academy Award then the popularity of the actors as well as the directors increases and therefore it would be harder and more expensive to sign contracts with them.

4. The cleaning of the data in Problem 2 was very significant. For example *duplicate* values can alter the final results by introducing biases in our model. Moreover, since *outliers* are extreme values that do not occur often, they make our classifier more inaccurate. Finally, we chose to discard the *blank* values instead of, for example, assigning them the typical value of the attribute. Nevertheless, I believe that this choice is safer since we have a lot of data and also we don't know what the exact actual entries for each attributed were.

5. Normalizing the values in Problem 3 had a very significant impact on the model. For instance, the imdb scores are in the range of $[0, 10]$ whereas the budget range is of $[0, 10^8]$. Since we used the Euclidean distance in our model, the term differences of the budget would dominate the sum. As a result, the "effect" of the `imdb_score` in our model would be diminished.

6. A potential issue with this classification model is that we are using only three attributes to predict whether a movie is likely to win the Academy Award. Nevertheless, there are still many attributes that play a significant role and that could help us improve our accuracy. Moreover, there is no guarantee that the Euclidean Distance is the best similarity measure for our application. There is a variety of other similarity measures that we could use in our model such as the *Manhattan Distance, Chebyshev Distance or the Minkowski Distance*. Moreover, in our model we chose the value of K to be 1. Nevertheless, the value of K plays a significant role in the accuracy of our model, therefore we could experiment further with our model by increasing K.

Extra Credit

Table 1 displays the accuracy of the model for all different similarity measures that were previously suggested. For consistency, the formulas for each of these methods are also included:

$$D_{EUCL}(\vec{d_1}, \vec{d_2}) = \sqrt{\sum_{i=1}^{m} \left[\vec{d_1}(i) - \vec{d_2}(i)\right]^2}$$

$$D_{MANH}(\vec{d_1}, \vec{d_2}) = \sum_{i=1}^{m} \left|\vec{d_1}(i) - \vec{d_2}(i)\right|$$

$$D_{CHEB}(\vec{d_1}, \vec{d_2}) = \max \left|\vec{d_1}(i) - \vec{d_2}(i)\right|$$

$$D_{MINK}(\vec{d_1}, \vec{d_2}, p) = \sqrt[p]{\sum_{i=1}^{m} \left[\vec{d_1}(i) - \vec{d_2}(i)\right]^p}$$

Table 1: Accuracy for different similarity measures and $K = 1$

Method	Accuracy(%)	Standard Deviation(%)
Euclidean	90.4051	5.1992
Manhattan	90.0975	4.5370
Chebyshev	90.2002	4.7263
Minkowski (p=5)	90.0977	4.7843

Using the table we notice that the Euclidean Distance outperforms all the other Distances. As a result we take the Euclidian distance and we experiment with the different values of K, summarizing our results in Table 2.

Table 2: Accuracy for different values of $K = 1$ using the Euclidean Distance

K	Accuracy(%)	Standard Deviation(%)
1	90.4051	5.1992
10	93.9215	2.8447
20	94.3999	2.6574
50	94.2288	2.8853

We see that the best performance increases as we reach $K = 20$ and then decreases as we further increase K to reach 50. As a result we notice that in order to produce more accurate predictions we need to increase the value of K which implies that we are using more "neighbors" in order to classify a given movie. This is logical, in the sense that we are using more information to make a decision. Moreover, it is worth mentioning that the standard deviation decreases as we increase K which means that we are now more certain about our classification. Nevertheless, we see that the as we increase the value of K to reach 50 the accuracy of our model starts decreasing. The reason for this is overfitting, which basically means that we are attempting to match too closely our training data while at the same time becoming less flexible in classifying our test data.

Appendix

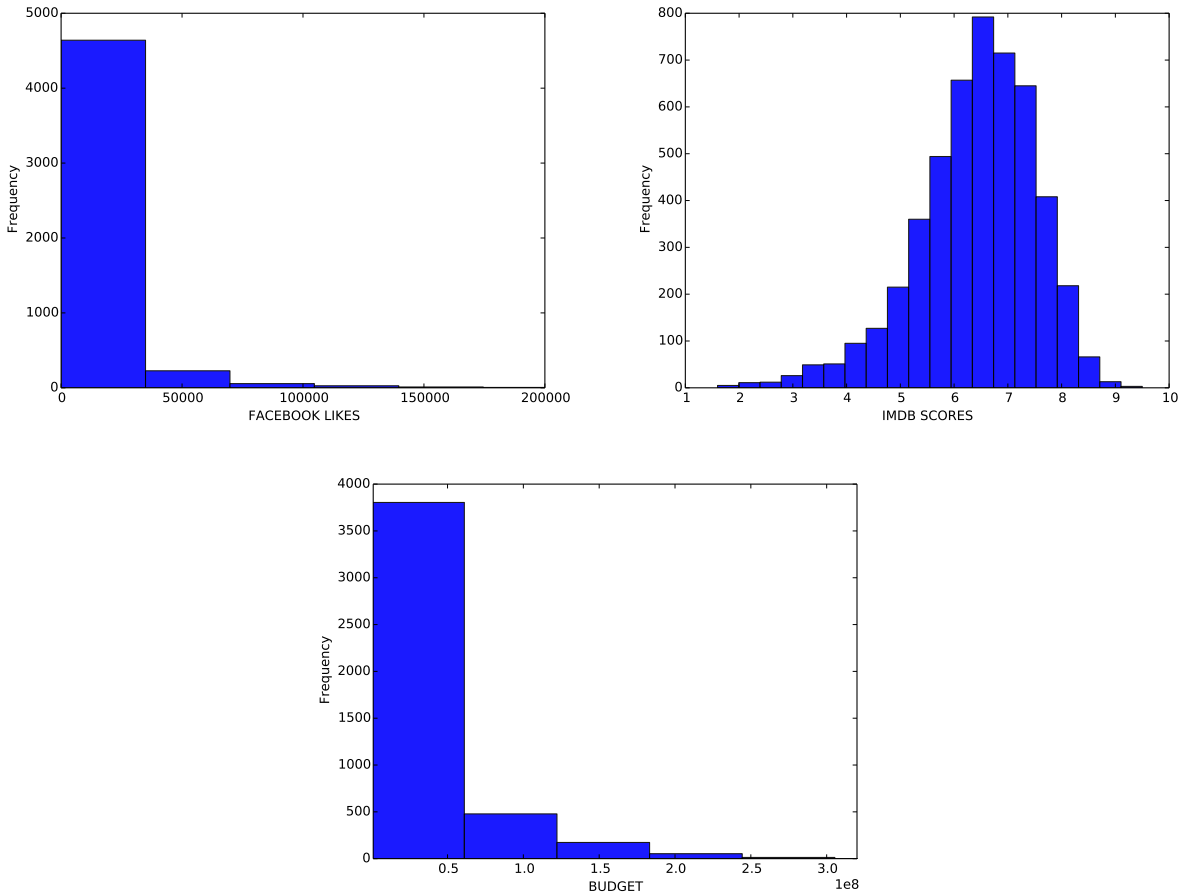

Figure 1: Histograms for the 3 selected attributes for Question 1