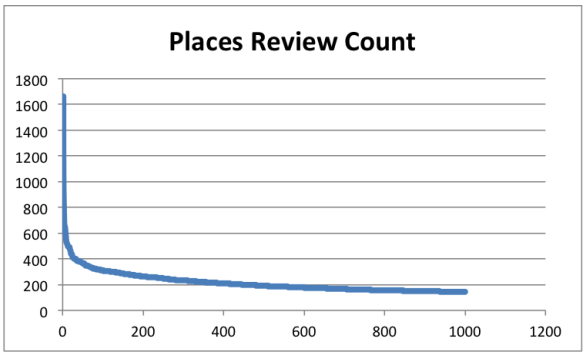
1 The Dataset

In this assignment we studied *Google Local's Maps and Restaurants* data. The goal was to extract restaurants' information from the dataset and to study how restaurants perform based on different features offered by the dataset. We studied three main features and how they affect a restaurant to stay in business; one was related to the geographical grouping of the restaurants, the other was how users' reviews affect a business and finally how users' ratings were involved. We used these three main features to predict whether a restaurant would stay in business or it would be closed.

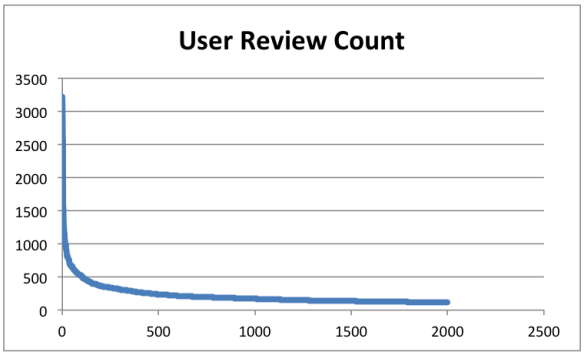
There were **3747937** users, **3114353** places, and **11453845** reviews in this dataset.

From the businesses in the dataset **3014137** were marked as open and **100215** were closed which means that **3.32%** of the businesses mentioned in the dataset were marked as closed.

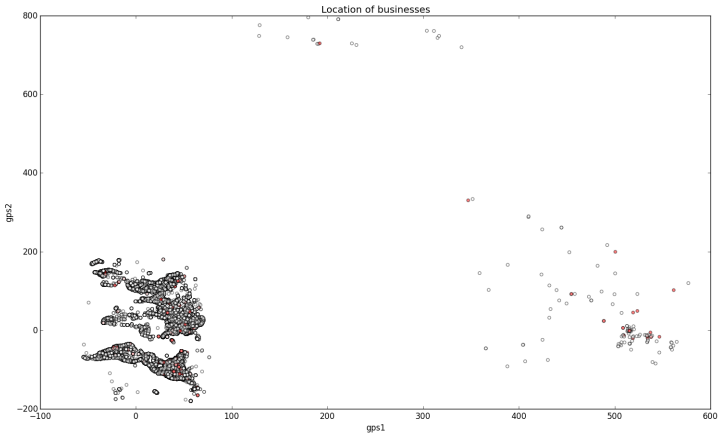
The place with the most number of reviews was *"Eiffel Tower"* with **1662** reviews. The number of reviews formed a sharp long-tail with only **126** places (**0.0040%** of the places) having **300** or more reviews and the rest with fewer reviews.



The user with username **"Ortelius Abraham"** had the highest number of reviews with **3220** reviews. Similar to the places, the number of reviews per user formed a long-tail with only **329** users (**0.0088%** of users) with **300** or more reviews.



Businesses formed clusters based on their location. The graph displays the concentration of all businesses both open and closed ones. There was a much larger number of reviews in the United States and Europe comparing to the rest of the world.



2 Predictive Task

In this report we studied how to predict whether a business is open or closed. The factors that we used in the prediction were the location of the place, the category of the business, the reviews corresponding to the business, and finally the ratings of each place.

The main challenge with this dataset was that it was a highly imbalanced dataset with only **3.32%** of the businesses marked as closed. Hence we did not rely much on *Classification Accuracy* or the *Classification error*; instead we used *True Negative Rate (TNR)*, *Balanced Error Rate (BER)*, and *F-score* to evaluate the result.

The prediction focused mainly on businesses which were categorized as "restaurant", "coffee-shop", or "fast-food". The main idea was that people's reviews on these types of businesses have a stronger impact than on businesses like grocery stores or tourist attractions.

For places like restaurants the location matters a lot, also the reviews that people leave for each place and the rating can be a determining factor on the health of the business in that place. It looked like that we had good amount of information that could be used to predict whether a business would survive or not.

3 Relevant Literature

The original data was based on Google's Restaurants and Maps on Google+. It was a large dataset and it was not possible to handle the data directly in a python application; to handle the data more effectively it was loaded into a MySQL database; then it was filtered and cleaned; all the aggregation and exploratory data analysis was performed inside of the database using SQL queries; eventually smaller excerpts of the data was exported as CSV files and loaded in Python to perform fitting, validation, and testing.

In the original data most of the reviews were related to an existing user and place; however, there were **2804835** reviews that did not have a known user or a place. For faster data retrieval and aggregation, the database enforced data integration, so all the records with no known user or place were dropped from the final analysis to satisfy indexing and foreign key constraints. For the remainder of this assignment we only worked with the reviews that had a known user and a known place entry. Also categories were cleaned and split into a separate table for faster data queries.

This analysis used *Support Vector Machines*, *Logistic Regression*, and *Linear Support Vector Machines* using *sklearn* library in python to fit and validate the data. *Linear Support Vector Machines* was later on added to fit large amount of data. The original *SVM*, due its quadratic running time, was unable to handle data that was larger than 100k records.

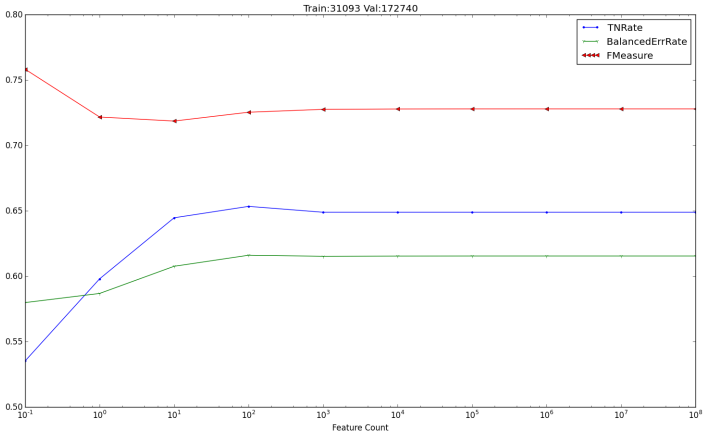
The preliminary assumption was that all three methods should yield similar results with some advantage on the non-linear SVM method.

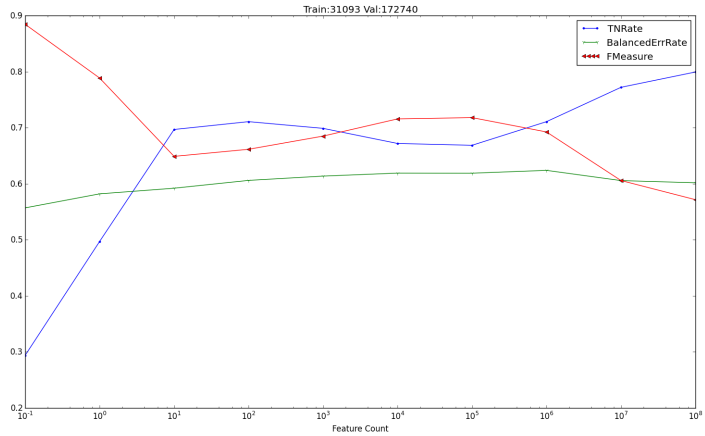
To compensate for the skewed data all three methods ran with automatic class weighting. Originally the code swept over possible class weight values in orders of magnitude but none of the manual values was able to outperform the automatic weight.

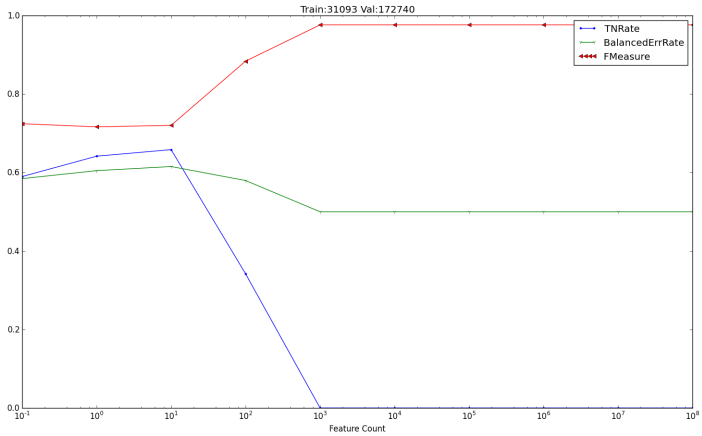
A validation set of size **5000** was used along with a training set of size **30000** to determine the best regularization factor for either of the methods. Based on the plot of the regularization factor vs. misc. error factors it was decided that the best values for each method were

Algorithm	C
LR	100
SVM	1e6
LinSVM	10

The x-axis on the following graphs are mislabeled as "feature" while it has to be "Regularization Factor - C"


Train:31093 Val:172740
TNRate
BalancedErrRate
FMeasure
Feature Count


Train:31093 Val:172740
TNRate
BalancedErrRate
FMeasure
Feature Count



4 Features

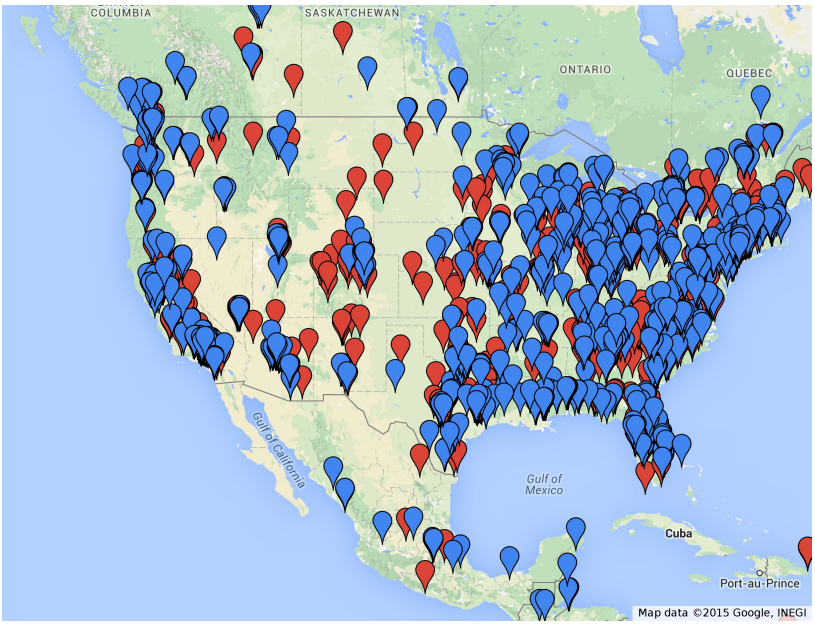
The original data was trimmed and filtered based on the following criteria.

- Places other than restaurants/coffee-shops/fast foods were filtered out. The categories included contained one of the words **restaurant, food, italian, mexican, hamburger, fast food, coffee, cafe, snack, pizza**. After this filtering only **882818** places were left
- All the users with fewer than **5** reviews were filtered out. After this filtering only **228426** users were left.
- The geographical locations were split into two fields
- For each review word counts were calculated
- For each review a sentiment score was calculated.

"Location", "Ratings" and "Reviews" were the three main features used to classify businesses. "Location" was one of the factors on how a business would perform;

The overlay map shows that the location is not a very definite factor in determining whether a place is closed or open but it can give us some clue.

The following map is a subset of data of the open/closed businesses in the United States



Using only the location of a place both SVM and LR gave **57%** of BER which was slightly better than random classification. If we enhanced the two features by adding their non-linear forms (e.g. gps^2) and only after feature scaling we could get:

Algorithm	TNR	F-Score
LR	0.654	0.684
SVM	0.655	0.682
LinSVM	0.628	0.696

Businesses which have a higher average of rating and receive more ratings should do better. In the analysis we did not consider the timing of the reviews (which can be important). When we only considered the average and the number of ratings for a business, SVM gave us **62%** and LR resulted in **61%** of BER. When we enhanced the two features by adding their non-linear forms we got:

Algorithm	TNR	F-Score
LR	0.685	0.637
SVM	0.420	0.800
LinSVM	0.682	0.772

This showed that rating on itself was a stronger factor.

Another main set of features that were included in the analysis were latent "Review" related features. There were two factors related to reviews included in the predictive model. One was the average number of words per review. The assumption was that the more people write about a place, the more they like it; negative reviews are many and terse!.

Another factor included was based on a basic sentiment analysis on the reviews. By using a dictionary of positive/negative words with their sentiment weight every review was given a score based on the positive and negative words that it contained. For each place the total sentiment score of all reviews was added up and averaged based on the number of reviews to give an idea on how positive/negative reviews were.

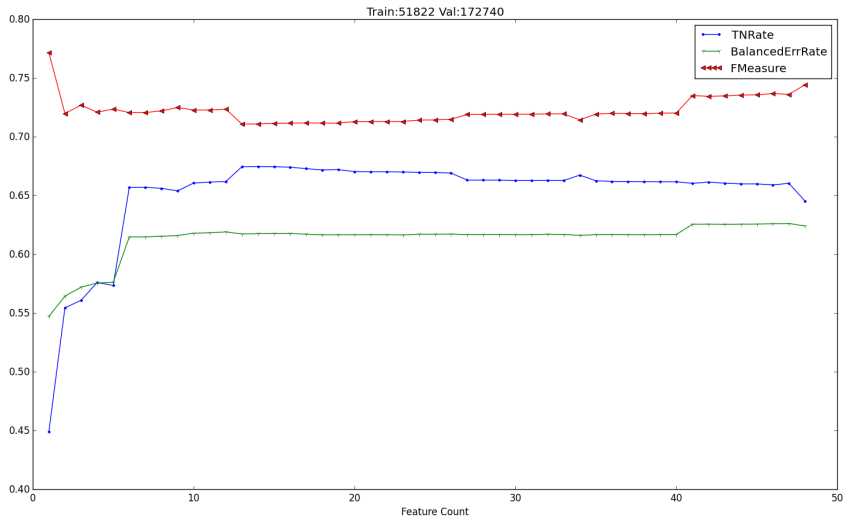
The original sentiment score was computed by using a dictionary of positive/negative words [1]. There were **117569** words in the dictionary with either a negative or a positive weight. If a word was not found in the dictionary it was given the score of 0. Unfortunately the dictionary covered only the English words.

After adding these two features there was an improvement in the F-Measure. The following results were from combining both review and rating features.

Algorithm	TNR	F-Score
LR	0.685	0.768
SVM	0.426	0.816
LinSVM	0.682	0.793

5 Model

The final set of features included "Average Rating", "Average Review Word Count", "Average Review Score", "GPS1", and ""GPS2". The features were further enhanced by adding their non-linear forms (e.g. $GPS1^2$,$GPS1 \times GPS2$) After adding enhanced features, there was the problem of overfitting, hence we used the validation set to determine what was the best number of features to select. We did not permute on the all possible feature combinations rather we selected a subset of features starting with the most effective ones and their non-linear factors The following graph shows that there was the maximum TNR at $\#features = 13$



The three algorithms yield almost the same result for the smaller training sets. The main advantage of *Linear SVM* was that we could run it on large subset of data. Nevertheless the following table summarizes the result of running the training on 50k data points. However unlike other tables this time, the test is actually done on the test set and not the validation set.

Algorithm	Error Rate	TNR	BER	F-Score
LR	0.393	0.648	0.626	0.746
SVM	0.422	0.705	0.628	0.703
LinSVM	0.394	0.648	0.628	0.745

6 Conclusion

After including the entire training set Linear SVM fitted on the entire training set, resulted in the following values

Algorithm	Error Rate	TNR	BER	F-Score
LinSVM	0.38	0.654	0.632	0.753

Automatic weighting had significantly increased the error rate but there was no other choice considering the skewed data. The geographical location was not a very determining factor. One reason might have been that there is no clear clustering on businesses that are closed in the same area and treating the coordinates as a linear factor had been wrong.

When it came to the ratings, there was no time component included. Ideally we needed to consider reviews that were written at some time before the closing

of a business. Also considering the number of reviews as a feature might not be logical considering that closed businesses must have fewer reviews in our data set. So the number of reviews is more a correlated factor not a causing factor.

If we could collect when the business was established and when they were closed then we could combine the dates with the review information and create stronger features.

The sentiment analysis was as basic as it could get. Although it had a relatively strong prediction effect, it should have been done more carefully and throughly.

References

[1] SentiWordNet v3.0.0 (1 June 2010)
`http://sentiwordnet.isti.cnr.it`