Text Predictive Analytics Using Dimensionality Reduction Techniques in Regression and Classification Models (Submitted November 2016)

Tung Hoang[1] – 303367, Tae-young Kee[2] – Student number

[1]Department of Information and Service Management, Aalto University School of Business, Finland
[2]Department of Information and Service Management, Aalto University School of Business, Finland

ABSTRACT

The goal of this project is to perform machine learning algorithms to make predictions on sample text datasets. To predict numerical output, we used linear regression and polynomial regression methods. In order to make categorical prediction, we used different classification algorithms such as Multinomial Naïve Bayes, Logistics Regression, and Linear Discriminant Analysis (LDA). For evaluation metrics, we used F-score for classification and mean squared error for regression. We also applied LDA as dimensionality reduction (in regression) and forward search method using coefficients to select the most important features. Regression algorithm with highest performance was 2-degree polynomial regression with forward search method. Forward search method is also proved to be effective with three classifiers among which Naïve Bayes out-performs others on a high-dimensional and sparse data.

***Keywords* — classification, forward search, F-score, mean squared error, predictive, regression, text analytics**

I. INTRODUCTION

Yelp is one of the most popular sites where users can rate and search for reviews local businesses. For different local businesses, users rate them from 1-5 stars and leave reviews. Also, users can vote on helpful or useful reviews from other users. We could use this huge amount of text review data in predicting two things: user rating on local business, number of votes for each review. Findings from first idea can be extended to many different areas such as books or TV series where users leave lot of reviews through blog or articles but not numerical rating. Insights from second idea could be applied to any other area where text assessment is needed.

In this project, we want to apply various supervised learning algorithms we have learnt from the Machine Learning course on text analytics problems and predict relevant metrics as accurately as possible. We use regression models to predict the number of votes and for user rating we use classification algorithms to predict whether the review received more than 3 stars or less. We also experiment to fine-tune the based algorithms with the forward search feature selection method.

II. METHODS AND ALGORITHMS

A. Evaluation Metrics

1) Mean squared errors (MSE)

For Regression, we use mean squared error as the evaluation metric in order to measure the prediction performance. MSE measures the average of the squares of difference between the predictor and what is predicted. When there are n observations, MSE is calculated individually by the equation below:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

where $\hat{Y}_i$ is the vector of n predictions and Y_i is the vector of actual values.

2) F-score

We use the F-score as the evaluation metric to measure our classifiers' performances. This metric measures the accuracy of the algorithm using the statistics precision (p) and recall (r). The F-score, also known as F1 Score or F-measure, is calculated as:

$$F1 = 2\frac{pr}{p+r} \quad \text{where: } p = \frac{TP}{TP+FP} \text{ and } r = \frac{TP}{TP+FN}$$

In the equation above: TP, FP, FN are the number of True Positives, False Positives, and False Negatives respectively.

In the binary classification problem, the F1 metric is more reliable than the accuracy score alone because it weights recall and precision likewise so that the relative contribution of precision and recall to the score are equal. Consequently, the F1 metric will favor moderately good performance on both precision and recall rather than extremely good performance on one and poor on the other. An F1 score reaches its best value at 1 and worst score at 0. [1]

B. Regression

Regression is a one of a parametric method that aims to describe output r, dependent variables, as a function of the input, independent variables. Output is described as a sum of deterministic function of the input and random noise.

$$r = f(x) + \epsilon$$

Since $f(x)$ is unknown, our effort is to approximate by using estimator, $g(x/\theta)$ defined by a set of parameters θ

In **multivariate linear regression**, output r is written as a linear function, weighted sum of multiple input variables and noise. Multivariate linear model is:

$$r^t = g(x^t|w_0, w_1, w_2, \dots, w_d) + \epsilon$$
$$= w_0 + w_1 x_1^t + w_1 x_1^t + \cdots + w_d x_d^t + \epsilon$$

Here error is assumed to have normal distribution with mean 0 and constant variance.

$\epsilon \sim N(0, \sigma^2)$ And the parameters are decided by maximizing the likelihood, which is equivalent to minimizing the sum of squred errors.

$$E(w_0, w_1, w_2, \ldots, w_d|X) = \frac{1}{2}\sum_t (r^t - w_0 - w_1 x_1^t - w_2 x_2^t - \cdots \cdot - w_d x_d^t)^2$$

The **multivariate polynomial regression** is the extension of linear regression by constructing polynomial features from the coefficients.

For instance, if we want to fit a praboloid instead of palne to the data, two-dimensional linear regression model as (1) can be transformed as (2) but as we can see, resulting polynomial regression model is still in the class of linear regression.

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 \quad (1)$$

$$\hat{y}(w, x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2 \quad (2)$$

As Alpaydin pointed out (Alpaydin, 2010), linear regerssion gives us useful information regarding the features. By looking at coefficients we can see whether each feature has either positive or negative effect on the output, Also, if all features has the same range, by looking at absolute values of coefficients, we can rank the features based on absolute valus of coefficients and figure out which features are important and which are not. [2]

C. Classification

1) Multinomial Naïve Bayes

We select a generative model like multinomial Naïve Bayes classifier as the first basic choice for classification as this probabilistic learning method is suitable for the data with multiple discrete features such as a bag of word. Naïve Bayes is traditionally used and proved to be the most suitable for text classification. [3]

In the Multinomial Naïve Bayes text classification, the probability of the comment (document) d being in class c is computed as:

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

where $P(t_k|c)$ is the conditional probability of term t_k occurring in the comment of class c.

$P(c)$ is the prior probability of a comment occurring in class c.

The goal of the algorithm is to find the best class for the document (comment) by maximizing a posterior (MAP) class cMAP:

$$c_{MAP} = \arg\max_{c \in C} P(c|d) = \arg\max_{c \in C} P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

the MAP estimate may encounter one problem that if some word in the training data only occurred in one class (for example, the word 'bad' only occurs in down-vote comment), the maximum likelihood estimation for the other class will be zero. To avoid this problem, ie. eliminating the zeros, we use add-one or Laplace smoothing so that we add one to each of the word count. the conditional probability of the word t_k that appears in a document belonging to class c is estimated by:

$$\hat{P}(t_k|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct} + 1)}$$

where T_{ct} is the number of occurrence of t in the training documents from class c and V is the total number of terms in the vocabulary.

2) Logistics Regression

In addition to generative model like Naïve Bayes, we considered Logistic Regression (or maximum-entropy classifier) as a discriminative model that is likely to be effective in a text classification problem. It is also useful when the coefficients of the logistic regression can be used for interpreting and selecting features (for feature selection process).

Logistic Regression model aims to predict the state of a logic variable Y belonging to a binary class {True/False} through a function $f : R \to [0,1]$. The standard logistic function is a sigmoid function and is defined as:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Considering a text classification problem where we have a training data set of n samples.

$x_i \in R^m, \forall i \in [1..n]$ is the input vector of dimension m for a sample n and $y_i \in \{0,1\}\ \forall i \in [1..n]$ is the value to predict corresponding to it. And x_{ij} denotes the elements in dimension j of the vector x_i. The objective of the logistic regression is to compute the sigmoid function to find the conditional mean $E(Y|X = x)$ reduced to the output space of [0,1]:

$$E(Y|X = x_i) = \frac{1}{1 + e^{-\beta^T x_i}}$$

where $\beta \in R^m$ is the vector composed of $\beta_j$elements so that $\beta^T x_i = \sum_{j=1}^m \beta_j x_{ij}$

We need to maximize the above expected value of Y given x_i by maximizing the log-likelihood function:

$$\ln(l(\beta)) = \sum_{i=1}^n [y_i \ln(\hat{y}_i) + (1 - y_i)\ln(1 - \hat{y}_i)]$$

We then use gradient descent algorithm to find the maxima of the above log-likelihood function. [4]

3) Linear Discriminant Analysis (LDA)

We also consider Linear Discriminant Analysis (LDA) as another possible binary text classifier and also a dimensionality reduction method.

The objective of LDA is to perform dimensionality reduction while preserving as much as of the class discriminatory information as possible. Consider a binary classification problem, where we have a set of D-dimensional samples $\{x_1 x_2, \dots, x_N\}$, and there N_1 points belongs to class $C_1$and N_2 points belongs to class C_2 . LDA seek to obtain a scalar y by projecting samples x onto a line:

$$y = w^T x$$

One approach to separate the two classes when projected onto w us by separating the projected class means, which is choosing w to maximize:

$$m_2 - m_1 = w^T(m2 - m1)$$

with the constraint of w to have unit length, ie. $\sum_i w_i^2 = 1$

where m1 and m2 are the mean vectors of two class C1 and C2 respectively.

To solve the problem of overlapping when projected onto the line joining two means, Fisher proposed to maximize a function that create a large separation between the projected class means while giving a small variance within each class, thus minimizing the class overlap. The solution is known as Fisher's linear discriminant, which is a specific choice of direction for projection of the means onto one dimension:

$$w \propto S_w^{-1}(m_2 - m_1)$$

where S_w is the total within-class covariance matrix, given by

$$S_w = \sum_{n \in C_1} (x_n - m_1)(x_n - m_1)^T + \sum_{n \in C_2} (x_n - m_2)(x_n - m_2)^T$$

Note that LDA can also be used to perform supervised dimensionality reduction, by projecting the input data to a linear subspace consisting of the directions which maximize the separation between classes. The dimension of the output is necessarily less than the number of classes. Thus LDA is in general a strong dimensionality reduction algorithm; however, it only makes senses in a multiclass setting. Therefore, LDA can be used as a classifier itself in a binary classification problem while in a multi-classes dataset (or regression on a ordinal variable), it can be used as a feature extraction before the regression step. [5]

D. Feature Selection

In feature selection, we are interested in finding subset of attributes that can describe most of the information that original data contains. In case of d features, there exist 2^d subsets. There are various methods to select subset of original features but we focused on forward search method in this report. Forward search is the method which begins with no feature and adds one best feature at each step. This method is relatively intuitive and easy to understand but need to have proper algorithm to choose the next best feature to add.

To implement the forward search method, we utilize the structure of the linear or logistics regression model for the feature ranking, ie. using coefficients of the regression model for selecting the features. The idea is that when all features are on the same scale, the most important features should have the highest coefficients in the model while features uncorrelated with the output variables should have close to zero coefficient values. Thus we want to experiment to see if simply using a certain number of the most important features can actually improve the models. We will rank the features by its coefficients from the regression and run the models on the most important feature, then add the next features in the rank and iterate the process until all the original features are included in the models.

III. EXPERIMENTS

In this section, we will first describe the dataset that we use for the text analytics problem. Next we present each step of how we conducted various experiments on various regression and classification algorithms also in combination with the forward search method.

A. Yelp's Review Dataset

Data used for regression consists of 6000 rows and 52 columns. Out of 52 columns, 1st column is id, and 50 columns are features and the last column is the number of user votes that shows how many users found this review as useful. Features are word counts of 50 different words generated from the bag-of-words model.

The data used for classification is almost identical to the regression set except for the last column, which is a binary: 0 means users do not find the review useful (received less than 3 stars) and 1 means that the review is voted as useful (more than 3 stars)

B. K-fold cross validation

In all the experiments, we use K-fold cross-validation (with k = 10) to assess how accurately the predictive models will perform in the unseen data (test data). The original training sample is randomly partitioned into 10 equal sized subsamples (N = 500 each). For each iteration, one subsample is retained as the validation set for testing the model, while the remaining 9 subsets (N=4000) are used as training data. The cross-validation process is repeated 10 times, with each of the 10 subsamples used exactly once as the validation set. The results of 10 iterations are finally averaged to create a single estimation of the whole training dataset.

C. Dimensionality Reduction

1) Feature Selection with Forward Search

For feature selection in regression, we first conducted linear regression with scikit learn's machine learning library to normalize the features and calculate coefficients of features to figure out the most important features for prediction. After the coefficients are calculated, we changed them to absolute value and ranked the features by descending order of coefficients.

Similarly, in the classification methods, we first conduct the basic logistic regressions with all the features in the data set then examine the coefficients. The next step is ranking the features based on their importance as we have done in the regression.

2) Feature Extraction with Linear Discriminant Analysis (LDA)

In order to perform supervised dimensionality reduction with feature extraction method, we used Linear Discriminant Analysis algorithm in the sci-kit learn library. We plugged in both input and output data from training set to build the dimensionality reduction algorithm. With this algorithm, we created new data with reduced dimensions for both training data and validation data. With new training data, we built both linear regressor and 2-degree polynomial regressor and tested the model performance by using the new validation data. We experimented with all possible number of dimensions from 1 to 50 by changing 'n_components' parameters in the library.

D. Regression Experiments

We built 50 models using training data with Linear Regression library from scikit-learn by implementing forward search method. The First model was built by using 1 best feature with the highest coefficient and the second model was built by using two features with the first and the second highest coefficient and so on. With these 50 models, we calculated prediction error on validation data set and observed the model performance based on number of features included.

To find the right degree for polynomial regression, we first created different degrees of model with 7 most important features that we found from above experiment. To implement the experiment, we combined polynomial regressor and linear regressor using 'pipeline' in sklearn library. 'Pipeline' is used to combine several steps that can be validated together while setting different parameters. We tried to change degree of polynomial regressor from 1 to 10 but due to limited computing capacity we could only see results of model less than 7-degree. This wasn't big problem in carrying out the experiment since model performance was the highest with 2-degree polynomial and it decreased quite rapidly as the degree increases. After figuring out that 2-degree polynomial regressor has much better performance than normal linear regressor, we conducted same experiment with forward search method as we did with linear regression.

E. Classification Experiments

First, we run Multinomial Naïve Bayes, Logistics Regression and Linear Discriminant Analysis (for binary class) using the sklearn packages on the Yelp training dataset using all the 50 features as the basic algorithms for the text classification problems. The results of these algorithms will be compared and used as the benchmark for the next experiments where we select a certain number of features or reduce the dimensionality of the data.

Next we use the forward search method to rank the features based on their importance, i.e, the absolute value of the coefficients as the result from the logistic regression. We combine the three classification algorithms with the forward search by iterating through 1 to 50 ranked features.

After comparing and finding the best tuning of the classification algorithm, we apply it to the remaining 1000 test data, evaluate and compare the result to the training data.

IV. RESULTS

A. Regression Experiments Result

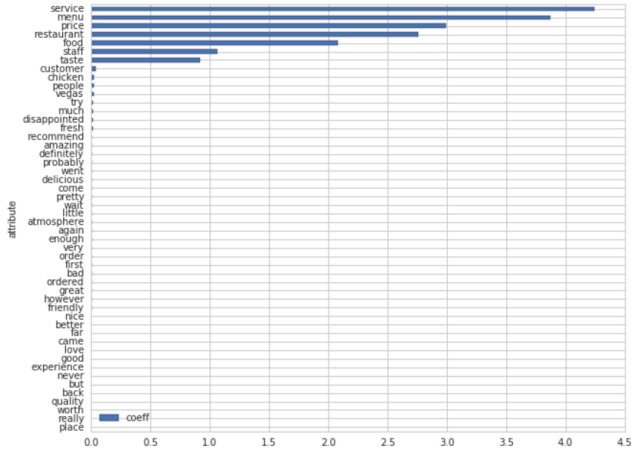

Figure1 – Feature coefficient from linear regression analysis

Figure 1 is the coefficients of each features resulting from linear regression analysis. We can observe that top 7 features have much higher coefficients than the rest. Mean squared errors of 10 different linear regression models with different number of features are shown in **Table 1** below.

Number of attributes	Mse	columns
1	6.213048	service
2	4.267664	service,menu
3	3.543569	service,menu,price
4	2.520092	service,menu,price,restaurant
5	0.287177	service,menu,price,restaurant,food
6	0.090329	service,menu,price,restaurant,food,staff
7	0.047817	service,menu,price,restaurant,food,staff,taste
8	0.047977	service,menu,price,restaurant,food,staff,taste...
9	0.048194	service,menu,price,restaurant,food,staff,taste...
10	0.048243	service,menu,price,restaurant,food,staff,taste...

Table 1. Mean squared error on subsets of feature

Model performance was the highest with mean squared error 0.047816 when model includes 7 features and remains the same level as the model include more and more attributes. If we include all 50 features without feature selection, mean squared error is 0.048562.

Table 2 is the result of experiments with different degrees of polynomial regression with 7 features. 2-degree polynomial regression has the highest performance with mean squared error 0.02697.

degree	MSE
1	0.04781701
2	0.02697002
3	3.58223E+18
4	7.75565E+17
5	1.054444

Table 1. Mean Squared Error on different degrees of polynomial regression

When we experimented with 50 different models with 2-degree polynomial regression, the tendency of model performance was similar with that of linear regression models. The model with 7 features had the highest performance with mean squared error 0.02697 and it remained almost same level until the last model with 50 features.

In the experiment with Linear Discriminant Analysis, both for linear regression and 2-degree polynomial regression model performance improved little bit as number of components increases and remained quite steady after certain point. For linear regression, mean squared error reaches its minimum 0.048477 and for 2-degree polynomial regression minimum mean squared error was 0.035162 when the number of components was 19.

A comparison of mean squared error on test dataset using different dimensionality reduction method and Regression algorithm is shown on **Figure 2**. We can see that 2-degree polynomial repressor with forward search dimensionality reduction method results in the best performance when the model includes 7 most important features.

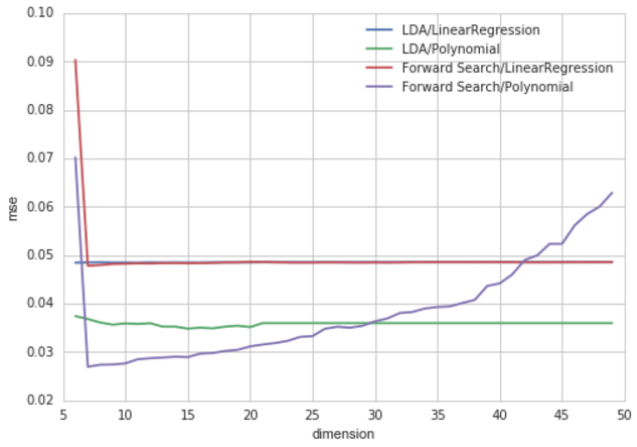

Figure 1. Regression Models Comparison

In order to validate the model and calculate the accurate model performance, we built a final model with all training data (5000rows) and tested with test data (1000rows) that we left untouched. The mean squared error of 2-degree polynomial regression with 7 most important features were 0.03231.

B. Classification Experiments Result

The experimental results of the first base classifiers are shown in the **Table 3** (the F1-measure scores are rounded to three decimals). Accordingly, Multinomial Naïve Bayes has the highest F-Measure score.

Base Classifier	F-Score
Multinomial Naïve Bayes	**0.794**
Logistics Regression	0.786
Linear Discriminant Analysis	0.789

Table 3. Performances of the three classifiers

The performance of three classifications combined with feature selection pre-process can be seen in the **Figure 3** below.

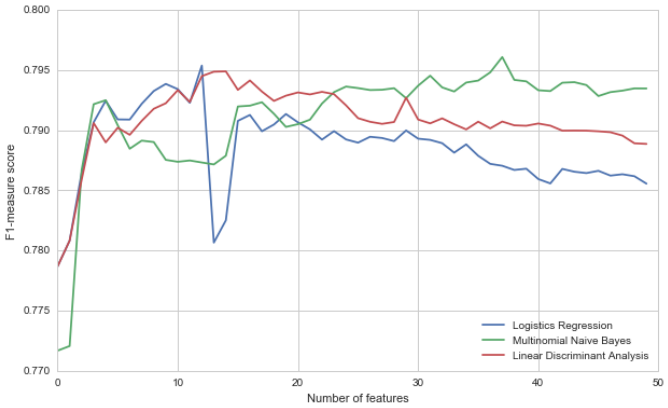

Figure 3. Performance of 3 classifiers using different number of features

The experiment results shows that all the base classifiers perform better with certain number of most important features. In other words, not all the words in the bag-of-word provide useful but rather noisy or irrelevant piece of information to the classification algorithm.

For example, for Logistic Regressions, the best performance is F-Score = 0.795 when the input include the first 13 most important features (words). Linear Discriminant Analysis (LDA) has the highest F-Score of 0.795 when 14 or 15 features with the highest coefficients are used as the input. However, for Multinomial Naïve Bayes, the optimal number of features is 38 with the F-score of 0.796, which is the highest among all the experiments.

When we apply the Multinomial Naïve Bayes classifier with 38 selected features on the testing dataset, the resulted F-score is 0.802, which is very close to the training data result.

V. DISCUSSION

A. Regression Models

The most important words in predicting number of votes were nouns such as service, menu, price, restaurant, food, staff rather than adjectives like disappointed, good, etc. So users prefer reviews that give information on those aspects rather than reviews that only describe emotional impression about the restaurant.

Linear Regression model with all 50 features had already good performance with mean square d error 0.048562 but we could improve it little by selecting top 7 most important features. Feature extraction through Linear Discriminant Analysis didn't succeed at improving model accuracy by reducing dimension. However, by experimenting with polynomial regression, we could improve model performance greatly with mean squared error 0.02967. Model performance on test data set wasn't as good as the one on the training data but still pretty good with mean squared error 0.03231.

B. Classification Models

In the classification problem, Multinomial Naïve Bayes performs better than the two other classifiers Logistic Regression and LDA. However, when we applied the forward feature selection technique to reduce the dimensionality of the original dataset, both Logistic Regressions and LDA will quickly reach the best performance with a small number of features (approximately one third of the original features). On the contrary, it takes more number of features (38 out of 50) for Naïve Bayes algorithm to have the best performance. In general, Naïve Bayes outperforms the other two algorithms when the number of features is high.

Thus we may infer that Logistic Regression and LDA are more sensitive to the high dimensionality of the datasets and a dimensionality reduction technique will greatly improve their results. On the other hand, Multinomial Naïve Bayes is a better algorithm to classify a high-dimensional and sparse matrix like the Yelp's bag-of-word dataset.

VI. CONCLUSION

In this project, we have experimented with various algorithms and dimensionality reduction technique to perform two tasks on the Yelp's review text dataset: regression of the voting and rating classification. To evaluate the effectiveness of different algorithms, we use Mean Square Error for regression and F-Score for classification.

We concluded that higher degree polynomial regression performs well with the combination of the forward search feature selection method. On the other hand, Multinomial Naïve Bayes out-performs other classification algorithms when the dataset such as the bag-of-word has a large number of features. However, Logistics Regression and Linear Discriminant Analysis also have effective results when combined with a dimensionality reduction algorithm.

Possibly future research can evaluate the performances of those algorithms based on other metrics, for example, Log Loss or Accuracy and Precision metrics for the classification problem. Also, further improvement could be using and compare these results with more advanced algorithms such as random forest for selecting the attributes and ensemble methods such as boosting and bagging in the context of text predictive analytics.

REFERENCES

[1] *Powers, David M W (2011).* "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation" . *Journal of Machine Learning Technologies. 2 (1): 37–63.*

[2] *Ethem Alpaydin (2010). Introduction to Machine Learning. The MIT Press, London. pp73, 103-105*

[3] *C.D. Manning, P. Raghavan and H. Schuetze (2008). Introduction to Information Retrieval. Cambridge University Press, pp. 234-265. http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html*

[4] *Zhu, X (2011). "Text categorization with logistic regression" http://pages.cs.wisc.edu/~jerryzhu/cs838/LR.pdf*

[5] *Christopher, M., Bishop (2006), Pattern Recognition and Machine Learning, pp. 186 - 189*