RATING PREDICTION

Submitted By:

Junaid Shaikh

ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot.
Most of the concepts used to predict the price of used cars are learned from Data Trained Institute and below documentations.

Some of the reference sources are as follows:

- Medium.com
- StackOverflow

Contents

INTRODUCTION

BUSINESS PROBLEM FRAMING

Rating prediction is a well-known recommendation task aiming to predict a user's rating for those items which were not rated yet by her. Predictions are computed from users' explicit feedback, i.e. their ratings provided on some items in the past. Another type of feedback are user reviews provided on items which implicitly express users' opinions on items. Recent studies indicate that opinions inferred from users' reviews on items are strong predictors of user's implicit feedback or even ratings and thus, should be utilized in computation.

The rise in E-commerce has brought a significant rise in the importance of customer reviews. There are hundreds of review sites online and massive amounts of reviews for every product. Customers have changed their way of shopping and according to a recent survey, 70 percent of customers say that they use rating filters to filter out low rated items in their searches. The ability to successfully decide whether a review will be helpful to other customers and thus give the product more exposure is vital to companies that support these reviews, companies like Google, Amazon and Yelp!. There are two main methods to approach this problem. The first one is based on review text content analysis and uses the principles of natural language process (the NLP method). This method lacks the insights that can be drawn from the relationship between costumers and items. The second one is based on recommender systems, specifically on collaborative filtering, and focuses on the reviewer's point of view.

We have a client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. the reviewer will have to add stars (rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don't have rating. So we, we have to build an application which can predict the rating by seeing the review.

Conceptual Background OF The Domain Problem

Recommendation systems are an important unit in today's e-commerce applications, such as targeted advertising, personalized marketing and information retrieval. In recent years, the importance of contextual information has motivated generation of personalized recommendations according to the available contextual information of users. Compared to the traditional systems which mainly utilize user's rating history, review-based recommendation hopefully provides more relevant results to users. We introduce a review-based recommendation approach that obtains contextual information by mining user reviews. The proposed approach relates to features obtained by analysing textual reviews using methods developed in Natural Language Processing (NLP) and information retrieval discipline to compute a utility function over a given item. An item utility is a measure that shows how much it is preferred according to user's current context. In our system, the context inference is modelled as similarity between the user's reviews history and the item reviews history. As an example, application, we used our method to mine contextual data from customer's reviews of technical products and use it to produce review-based rating prediction. The predicted ratings can generate recommendations that are item-based and should appear at the recommended items list in the product page. Our evaluations (surprisingly) suggest that our system can help produce better prediction rating scores in comparison to the standard prediction methods.

As far as we know, all the recent works on recommendation techniques utilizing opinions inferred from user's reviews are either focused on the item recommendation task or use only the opinion information, completely leaving user's ratings out of consideration. The approach proposed in this report is filling this gap, providing a simple, personalized and scalable rating prediction framework utilizing both ratings provided by users and opinions inferred from their reviews. Experimental results provided on dataset containing user ratings and reviews from the real-world Amazon and Flipkart Product Review Data show the effectiveness of the proposed framework.

Review OF Literature

In real life, people's decision is often affected by friends action or recommendation. How to utilize social information has been extensively studied. Yang et al. propose the concept of "Trust Circles" in social network based on probabilistic matrix factorization. Jiang et al. propose another important factor, the individual preference. some websites do not always offer structured information, and all of these methods do not leverage user's unstructured information, i.e. reviews, explicit social networks information is not always available and it is difficult to provide a good prediction for each user. For this problem the sentiment factor term is used to improve social recommendation.

The rapid development of Web 2.0 and e-commerce has led to a proliferation in the number of online user reviews. Online reviews contain a wealth of sentiment information that is important for many decision-making processes, such as personal consumption decisions, commodity quality monitoring, and social opinion mining. Mining the sentiment and opinions that are contained in online reviews has become an important topic in natural language processing, machine learning, and Web mining.

Motivation OF The Problem Undertaken

The project was first provided to me by FlipRobo as a part of the internship program. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary objective. Many product reviews are not accompanied by a scale rating system, consisting only of a textual evaluation. In this case, it becomes daunting and time-consuming to compare different products in order to eventually make a choice between them. Therefore, models able to predict the user rating from the text review are critically important. Getting an overall sense of a textual review could in turn improve consumer experience. However, the motivation for taking this project was that it is relatively a new field of research. Here we have many options but less concrete solutions. The main motivation is to build a prototype of online hate and abuse review classifier which can used to classify hate and good comments so that it can be controlled and corrected according to the reviewer's choice.

Analytical Problem Framing

Mathematical/ Analytical Modeling OF The Problem

In this perticular problem the Ratings can be 1, 2, 3, 4 or 5, which represents the likely ness of the product to the customer. So clearly it is a multi-classification problem and I have to use all classification algorithms while building the model. We would perform one type of supervised learning algorithms: Classification. Here, we will only perform classification. Since there only 1 feature in the dataset, filtering the words is needed to prevent overfit. In order to determine the regularization parameter, throughout the project in classification part, we would first remove email, phone number, web address, spaces and stops words etc. In order to further improve our models, we also performed TFID in order to convert the tokens from the train documents into vectors so that machine can do further processing. I have used all the classification algorithms while building model then turned the best model and saved the best model.

Data Sources & Data Formats

The data set contains nearly 124984 samples with 3 features. Since Ratings is my target column and it is a categorical column with 5 categories so this problem is a Multi-Classification Problem. The Ratings can be 1, 2, 3, 4 or 5, which represents the likely n ess of the product to the customer. The data set includes:

- Comments: Review Content of the Review.
- Ratings: Ratings out of 5 stars.

This project is more about exploration, feature engineering and classification that can be do ne on this data. Since the data set is huge and includes multi classification of ratings, we ca n do good amount of data exploration and derive some interesting features using the Revie w column available.

We need to build a model that can predict Ratings of the reviewer.

Data Pre-processing

Data pre-processing is the process of converting raw data into a well readable format to be used by Machine Learning model. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model.

I have used following pre-processing steps:

- ✓ Importing necessary libraries and loading dataset as a data frame.
- ✓ Checked some statistical information like shape, number of unique values present, info, null values, value counts etc.
- ✓ Checked for null values and I replaced those null values using imputation method. And removed Unnamed: 0.

- ✓ Visualized each feature using seaborn and matplotlib libraries by plotting distribution plot and word cloud for each rating.
- ✓ Done text pre-processing techniques like Removing Punctuations and other special characters, Splitting the comments into individual words, Removing Stop Words, Stemming and Lemmatization.
- ✓ After getting a cleaned data used TF-IDF vectorizer. It'll help to transform the text data to feature vector which can be used as input in our 6 modelling. It is a common algorithm to transform text into numbers. It measures the originality of a word by comparing the frequency of appearance of a word in a document with the number of documents the words appear in. Mathematically, TF-IDF = TF(t*d) *IDF (t, d)

Data Inputs-Logic-Output Relationship

The dataset consists of 1 features with 1 label. The features are independent and label is dependent as our label varies the values(text) of our independent variables changes.

- I was able to see the words in the Review text with reference to there ratings using word cloud.

Hardware and Software Requirements and Tools Used

Hardware required:

- Processor — core i5 and above
- RAM — 8 GB or above
- SSD — 250GB or above

Software/s required: Anaconda

LIBRARIES:

The tools, libraries, and packages we used for accomplishing this project are pandas, NumPy, matplotlib, seaborn, SciPy, sklearns's, mlxtend, xgboost, joblib.

Through panda's library we loaded our csv file 'Data file' into data frame and performed data manipulation and analysis.

With the help of NumPy we worked with arrays.

With the help of matplotlib and seaborn we did plot various graphs and figures and done data visualization.

Train_test_split is a function in Sklearns's model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearns's Train_test_split will make random partitions for the two subsets.

With sklearns's StandardScaler package we scaled all the feature variables onto single scale. As these columns are different in scale, they are standardized to have common scale while building machine learning model. This is useful when you want to compare data that correspond to different units.

With sklearns's package we imported many regression models, we could obtain cross_val_score, which is an accuracy metric used to evaluate model, we could obtain best parameters of a model using GridsearchCV or RandomizedSearchCV, we could reduce skewness using power transform library of sklearns's.

Model/s Development and Evaluation

Testing of Identified Approaches (Algorithms)

I have converted text into feature vectors using TF-IDF vectorizer and separated our feature and labels. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models. Just making the Reviews more appropriate so that we'll get less word to process and get more accuracy. Removed extra spaces, converted email address into email keyword, and phone number etc. Tried to make Reviews small and more appropriate as much as possible.

Run & evaluate selected models

In this NLP based project we need to predict Ratings which is a multiclassification problem. I have converted the text into vectors using TFIDF vectorizer and separated our feature and labels then build the model using different algorithm

- RandomForest Classifier
- SVC
- ExtraTrees Classifier

Key Metrics for success in solving problem under consideration

I have used the following metrics for evaluation:

- I have used accuracy score, cross_val_score, multilabel_confusion_matrix all these evaluation metrics to select best suitable algorithm for our final model.
- Accuracy score is used when the True Positives and True negatives are more important. Accuracy can be used when the class distribution is similar.

Visualization

Word Cloud for particular ratings:

Rating 1:

Rating 2:

Rating 3:

Rating 4:

Rating 5:

From the above plots we can clearly see the words which are indication of Reviewer's opinion on products.

Here most frequent words used for each Rating is displayed in the word cloud.

Model Building

I have used group of classification algorithms, ran a for loop which contained the accuracy of the models along with different evaluation metrics.

First, we need to write a function which can find us best random state for train test split.

Then we shall iterate through all the models supporting classification algorithms to find the best models.

From above we get to know that the top models are:

- SVC
- ExtraTrees Classifier
- RandomForestClassifier

Fine tune all these models and find their best parameters to use.

Next, find the best random state for train test split.

Great all our algorithms are giving good cv scores. Among these algorithms I am selecting ExtraTrees Classifier as best fitting algorithm for our final model as it is giving least difference between accuracy and cv score

To analyze our model, we shall find the difference between actual and predicted value.

I have done hyperparameter tuning for **ExtraTreesClassifier** for the parameters like 'criterion', 'max_features', 'max_depth','class_wieght'.

```
parameters={'criterion':['gini', 'entropy'],'max_features':['auto', 'sqrt', 'log2'],'max_depth':[2,8,16,32,50],'class_weigh
clf = RandomizedSearchCV(ExtraTreesClassifier(), parameters, cv=5,scoring="roc_auc",n_jobs=-1, verbose=1)
clf.fit(x,y)
clf.best_params_
```

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
```

```
{'max_features': 'sqrt',
 'max_depth': 32,
 'criterion': 'gini',
 'class_weight': 'balanced'}
```

Best Random State w.r.t Best performing Model

```
besttrain(ExtraTreesClassifier(max_depth=32,criterion='gini',max_features='sqrt',class_weight='balanced'),x,y)
```

```
maximum accuracy_score is at random state : 73  and it is : 0.6425571068528223
```

And after doing hyperparameter tuning I got above parameters as best suitable parameters for our final model.

I have trained my final model using these parameters and it was unable to increase the accuracy of the model.

Conclusion

Key Features and conclusion of the study

In this project I have collected data of reviews and ratings for different products from amazon.in and flipkart.com.

we have tried to detect the Ratings in commercial websites on a scale of 1 to 5 on the basis of the reviews given by the users. We made use of natural language processing and machine learning algorithms in order to do so.

Then I have done different text processing for reviews column and chose equal number of texts from each rating class to eliminate problem of imbalance. By doing different EDA steps I have analysed the text.

We have checked frequently occurring words in our data as well as rarely occurring words.

After all these steps I have built function to train and test different algorithms and using various evaluation metrics I have selected **ExtraTreesClassifier** for our final model.

Finally, by doing hyperparameter tuning we got optimum parameters for our final model.

LEARNING OUTCOMES OF THE STUDY IN RESPECT OF DATA SCIENCE

I have scrapped the reviews and ratings of different technical products from flipkart.com and amazon.in websites. Improvement in computing technology has made it possible to examine social information that cannot previously be captured, processed and analysed. New analytical techniques (NLP) of machine learning can be used in property research. The power of visualization has helped us in understanding the data by graphical representation it has made me to understand what data is trying to say. Data cleaning is one of the most important steps to remove unrealistic values, punctuations, urls, email address and stop words. This study is an exploratory attempt to use 6 machine learning algorithms in estimating Rating, and then compare their results.

To conclude, the application of NLP in Rating classification is still at an early stage. We hope this study has moved a small step ahead in providing some methodological and empirical contributions to crediting institutes, and presenting an alternative approach to the valuation of Ratings.

LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

As we know the content of text in reviews is totally depends on the reviewer and they may rate differently which is totally depends on that particular person. So, it is difficult to predict ratings based on the reviews with higher accuracies. Still we can improve our accuracy by fetching more data and by doing extensive hyperparameter tuning.

While we couldn't reach out goal of maximum accuracy in Ratings prediction project, we did end up creating a system that can with some improvement and deep learning algorithms get very close to that goal. As with any project there is room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project.