Predicting TripAdvisor Review Ratings in Dublin

Diego Molamphy - 18326635
Arnav Bhattacharya - 22307812
Honglin Li - 19315272

Introduction

TripAdvisor is a reviews and ratings website for hotels, restaurants etc. For every entity listed on the site, there are customer reviews. Each customer review has a rating 1-5 attached to it. When searching for entities, TripAdvisor allows the user to apply a number of different filters based on location, type, price etc.

Our idea for this project came about as we were curious to see if we could use machine learning to be able to predict Dublin-based restaurant TripAdvisor review ratings based on the review text. We felt that this would be interesting as we would get an insight into what words in the reviews text have a strong influence on the review rating (specific to Dublin).

The input to our algorithm is textual reviews and the output is the rating. We thought that from this input and output, we would be able to not only create a model that is able to predict the rating for any text-based review on Dublin restaurants, we would also be able to distinguish words that heavily influence the ratings.

Dataset and Features

To retrieve the dataset, we decided to web-scrape the TripAdvisor Ireland website. The code for the web-scraping was developed using Python 3.9.2.

Firstly, we sent a get request to the TripAdvisor Ireland website page filtered for restaurants in Dublin. We did this by using the Python ‘requests’ package and by getting the url for the TripAdvisor Ireland website page filtered for restaurants in Dublin. This get request returned the html code for the page. From the html code, we got the link for each of the restaurants reviews page by using the Python ‘bs4’ package. For each link, we got the subsequent link for the first six pages of reviews. These were all added to an array. We then looped through this array, sending a get request to each link (page of reviews). For each page of reviews, we got the first couple hundred words of each review and the rating attached to that review and added it to a dataframe each time. We repeated this process for the first 5 pages of restaurants in Dublin on TripAdvisor.

Each page had 30 restaurants and we took the first 5 pages of restaurants in Dublin. Each page of restaurant reviews had 15 reviews and we took the first 6 pages of reviews. Some pages had additional restaurants that came up as sponsored ads. As a result, we gathered a total of 13,950 data points i.e. 13,950 review-rating pairs. We then outputted the final dataframe to a CSV file.

Methods

Upon preprocessing the data, we used the following models for predicting the rating from their respective reviews:

A. Logistic Regression:
   Logistic Regression is a type of a statistical model which is used for predicting the probability of an event happening. $logit(p_i) = ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i$
   The response variable is modeled with Bernoulli distribution: $ln(L) = \sum_{i=1}^{N}\left[ln(1 - p_i) + y_i ln(\frac{p_i}{1-p_i})\right]$
   In order to maximise the above equation we need to find the most optimal p_i value.

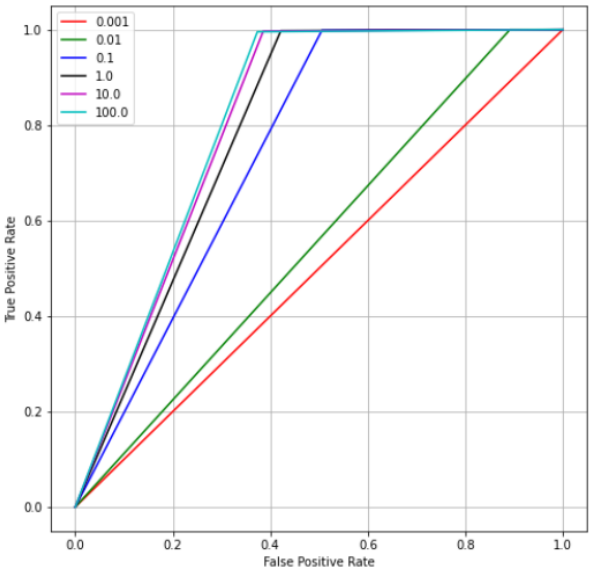
B. Lasso Regression: Lasso or the Least Absolute Shrinkage and Selection Operator is a type of linear regression model that incorporates variable selection and regularisation to increase the prediction accuracy. Regularization adds a penalty to change the cost function. Lasso Regression uses L1 penalty: $R(\theta) = \sum_{j=1}^{n}|\theta_j|$
   In Lasso Regression, the non-important features have a weight of approximately 0, making them useless when predicting new values. In our case, the hyperparameter that needs tuning is the C value used in the alpha parameter. Alpha value is defined by the following equation: $\alpha = \left(\frac{1}{2C}\right)$. So, alpha is inversely proportional to the C value.
C. k-Fold Lasso Regression: To improve the efficiency of the model, we used k-fold cross validation as the hyperparameter tuning. Of the previous model.
D. Ridge Regression: Ridge regression is used to analyse data with multicollinearity. When data contains multicollinearity, the least square are unbiased and variance is huge. This leads to a drastic reduction in prediction accuracy. Ridge regression adds penalty to cost function based of the following formula: $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h\theta(x^i - y^i\right)^2$. This model applies a L2 penalty.
E. Sequential Model: Sequential Model is a linear stack of layers which have single-input and single-output layer.
F. K-Nearest Neighbours Model: This model is used to solve regression as well as classification problems. KNN Model assumes that similar things exist in close proximity and groups them accordingly. Then predicts the data based on which group it would be in.

Experiments/Results/Discussion(2-3 pages)

We first generated the wordcloud for the entire review texts and then only for the extreme ends of the rating variable to observe the most commonly occurring words in all the cases. The data was divided in 80/20 train/test split. The wordclouds for extreme ends of rating were not upto our expectations. The wordcloud for rating=5 was expected to contain positive words that are used as compliments and vice versa for rating=1 wordcloud. SInce both the wordclouds had few commonly occurring words, we realised the need for preprocessing the data so that we could remove highly repetitive but neutral words from the review texts. The WordClouds are as follows:

The first wordcloud in the first row is for the entire dataset. The second word cloud in the first row is for the reviews which have rating=5(maximum). The wordcloud in the second row is for the reviews which have rating=1(minimum)

Logistic Regression:

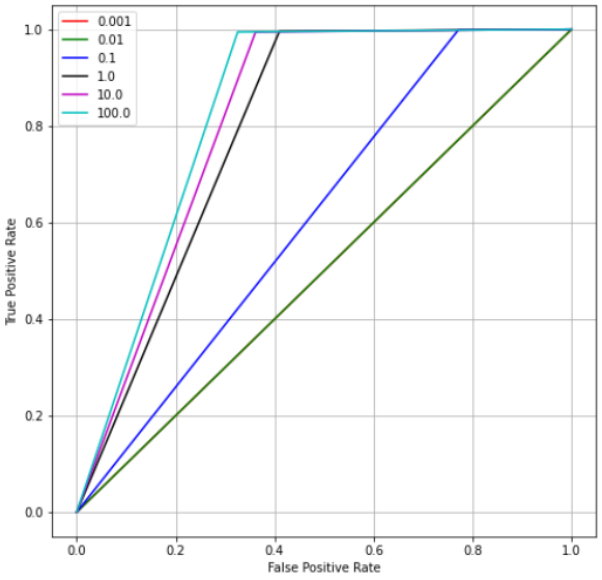


	Coefficient	Feature_Name	Coefficient_Magnitude
8242	-3.666503	worst	3.666503
5642	-3.479044	rude	3.479044
255	3.277603	amazing	3.277603
872	-3.053858	avoid	3.053858
2333	3.026600	excellent	3.026600
1867	2.954878	delicious	2.954878
5222	-2.919360	poor	2.919360
882	-2.566507	awful	2.566507
6441	-2.548053	terrible	2.548053
6404	-2.520006	tasteless	2.520006

L2 regularized Logistic Regression

A Logistic Regression Model is used to predict the ratings from the user reviews. We have tested the model against the following values for C: [0.001, 0.01, 0.1, 1, 10, 100]

As observed from the plot of true positive rate vs true negative rate, the model with L2 penalty term performs best when C is set to 100. The AUC score on testing data is 81.08%



	Coefficient	Feature_Name	Coefficient_Magnitude
8242	-14.023270	worst	14.023270
5642	-8.221406	rude	8.221406
882	-7.518995	awful	7.518995
6404	-7.424626	tasteless	7.424626
2003	-7.421044	disappointing	7.421044
2194	-7.083061	elsewhere	7.083061
255	7.013597	amazing	7.013597
6359	-6.925253	table more	6.925253
3101	-6.513759	great but	6.513759
5506	-6.422373	refused	6.422373

L1 regularized Logistic Regression

A Logistic Regression Model is used to predict the ratings from the user reviews. We have tested the model against the following values for C: [0.001, 0.01, 0.1, 1, 10, 100]

As observed from the plot of true positive rate vs true negative rate, the model with L2 penalty term performs best when C is set to 100. The AUC score on testing data is 83.4%

Lasso Regression:

```
Alpha: 0.0001
Train RMSE: 0.38555867471967503
Coefficients of Norm1 Complexity: 583.650256709029
Sum of Coefficients of Complexity: 583.650256709029
Test RMSE: 0.8219702344289035

Alpha: 0.001
Train RMSE: 0.6569618664863636
Coefficients of Norm1 Complexity: 48.02394510734982
Sum of Coefficients of Complexity: 48.02394510734982
Test RMSE: 0.7402389007178193

Alpha: 0.01
Train RMSE: 0.8166390558936223
Coefficients of Norm1 Complexity: 4.664010364922347
Sum of Coefficients of Complexity: 4.664010364922347
Test RMSE: 0.8616419735162372

Alpha: 0.1
Train RMSE: 0.9394280883611887
Coefficients of Norm1 Complexity: 0.04557462063433614
Sum of Coefficients of Complexity: 0.04557462063433614
Test RMSE: 0.9805743596820564
```

From the above outputs, we can observe that the Root mean square error for testing data is the least when the alpha value is set to 0.001. As we keep increasing the alpha value, the root mean square error increases.

	Coefficient	Feature_Name	Coefficient_Magnitude
10342	-1.177288	worst	1.177288
7097	-1.046972	rude	1.046972
6571	-0.828471	poor	0.828471
2515	-0.771026	disappointing	0.771026
1432	-0.714822	bland	0.714822
6300	-0.704059	overpriced	0.704059
8064	-0.698265	tasteless	0.698265
6926	-0.678376	refused	0.678376
9368	-0.644252	very disappointed	0.644252
8110	-0.628030	terrible	0.628030

On viewing the features based on the alpha value with the least root mean square error, we can confirm that it is the best fitting model as it focuses on the negative words in the reviews on which the review rating depends the most.

Ridge Regression:

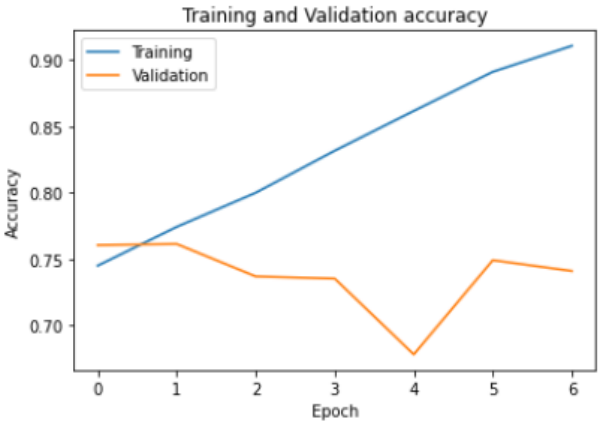
```
Alpha: 0.0001
Train RMSE: 0.12412254097490032
Coefficients of Norm1 Complexity: 6399.979421992019
Sum of Coefficients of Complexity: 6399.979421992019
Test RMSE: 2.6629275756461395

Alpha: 0.001
Train RMSE: 0.12457899319101728
Coefficients of Norm1 Complexity: 6185.579522113088
Sum of Coefficients of Complexity: 6185.579522113088
Test RMSE: 2.584994872757877

Alpha: 0.01
Train RMSE: 0.130488462762954
Coefficients of Norm1 Complexity: 5014.070363445464
Sum of Coefficients of Complexity: 5014.070363445464
Test RMSE: 2.1513618520647926

Alpha: 0.1
Train RMSE: 0.1628376409152819
Coefficients of Norm1 Complexity: 3071.5996107429473
Sum of Coefficients of Complexity: 3071.5996107429473
Test RMSE: 1.4478947936427193
```

From the above outputs, we can observe that the Root mean square error for testing data is the least when the alpha value is set to 0.1. As we keep decreasing the alpha value, the root mean square error increases.

	Coefficient	Feature_Name	Coefficient_Magnitude
8587	2.085027	the voucher	2.085027
6926	-2.062897	refused	2.062897
4553	1.980815	interested in	1.980815
7749	-1.964348	stag	1.964348
2816	1.895392	entered the	1.895392
2767	-1.888995	empty tables	1.888995
1331	1.819499	beside us	1.819499
7367	-1.748869	settled	1.748869
985	1.679129	at 6pm	1.679129
9357	-1.668395	very best	1.668395

On viewing the features based on the alpha value with the least root mean square error, we can confirm that it is the best fitting model as it focuses on the positive words in the reviews on which the review rating depends the most.

Sequential Model:



From the above plots, it can be easily identified that on increasing the epoch(the number of iteration), the loss decreases and the accuracy increases steadily.

```
TestAccuracy: 0.7430107526881721
              precision    recall  f1-score   support

           1       0.64      0.29      0.40        97
           2       0.25      0.41      0.31        73
           3       0.26      0.27      0.27       121
           4       0.34      0.28      0.31       394
           5       0.86      0.89      0.87      2105

   micro avg       0.74      0.74      0.74      2790
   macro avg       0.47      0.43      0.43      2790
weighted avg       0.74      0.74      0.74      2790
 samples avg       0.74      0.74      0.74      2790
```

The above accuracy score states the accuracy of the sequential model on testing data to be 74.3% which can be confirmed from the classification report

Summary(100-200 words)

Model	Accuracy(in percentage)
Logistic Regression	83.4
Lasso Regression	74
Ridge Regression	44
Sequential Model	75

From the above table we can clearly identify that the Logistic Regression Model performs best for u with an accuracy of 83.4%.

Contributions

Weekly meetings were held to discuss decisions, progress made and steps forward.

Diego

- Report
  - Introduction
  - Dataset & Features - Dataset
- Code
  - Web-Scraping
  - kNN Classifer and K-Fold for K

Arnav

- Report
  - Methods
- Code
  - Sequential Model
  - Logistic Regression
  - Lasso Regression
  - Ridge Regression

Honglin

- Report
  - Experiments/Results/Discussion
- Code
  - K-Fold for Sequential Model
  - Logistic Regression
  - Lasso Regression
  - Ridge Regression

Link to Github repository:
https://github.com/molamphd/ML_Group_Project