BigBasket Item Rating Prediction

Aditya Edassery
PES1UG20CS019
Department of CSE, PES
aditya.edassery@gmail.com

Data Analytics
UE20CS312
Department of CSE
PES University

Rudransh Kulkarni
PES1UG20CS361
Department of CSE, PES
rudranshklkrn@gmail.com

Abstract **BigBasket is the largest online grocery supermarket in India. Launched in 2011, the business has only been soaring to greater heights. Although new competitors have been able to set their foot in this market, BigBasket still controls majority of the market share. The primary objective of this research paper is to predict the item ratings on BigBasket to better understand what sub-categories generally do better, meaning more profits and revenue to the company. We have done a thorough comparison of the models that have been implemented in previous papers through our literature survey and a summary of the results obtained by these papers and the limitations in their approach have been mentioned in the appropriate section as well. Through our trial and research, the Ridge Regressive Model stands as the most viable in making these predictions. The other models we have used and mentioned below are, ANN, Decision Trees, Lasso Regression and MLR**

Keywords— BigBasket, Item Ratings, Lasso Regression, ANN, Decision Trees, Ridge Regression, MLR

I. INTRODUCTION

BigBasket (registered as, Supermarket Grocery Supplies Pvt. Ltd.) is an Indian online grocery delivery service. The company primarily delivers grocery goods found in convenience stores, home essentials and food supplies to its customers. BigBasket was founded in December 2011 and has its headquarters in Bengaluru, India. BigBasket was founded by V. S. Sudhakar, Hari Menon, Vipul Parekh, Abhinay Choudhari and V. S. Ramesh in 2011. In March 2019, BigBasket raised US$150 million in investment from Mirae Asset, Alibaba Group and CDC Group giving the company a valuation of over US$1 billion. In February 2021, the Tata Group acquired a 64.3% stake in BigBasket for around ₹9,500 crore (US$1.3 billion). As of 2022, BigBasket caters mainly to tier 3 and tier 4 towns with its community group buying approach, wherein users can create groups through messaging and social media platforms to buy products at reduced prices. Reports also emerged in the same year that BigBasket plans to roll out an express delivery service called BB Now (now operative in many cities of India) to focus more on quick commerce, wherein goods are delivered within 15 to 30 minutes upon the placement of orders. On 14 October 2020, BigBasket faced a massive security breach after the data of almost 20 million users was leaked onto the dark web. The database of BigBasket users placed was up for sale for over US$40,000 and the file included the customers' full names, email ID, hashed password, pin, contact numbers, full addresses, date of birth, location, and IP addresses of login, among other details. This goes to show one of the main drawbacks of online businesses and their customer identity issues. Consumer loyalty is characterized as a measure of the execution based on audits and input from the clients. Fulfillment level is an impalpable property which can help

decide if the organization can fulfill the client desires or not. Consumer loyalty can be clarified if the client is coming consistently to purchase from the store which demonstrates that the administrations are sufficiently effective to pull in the client. Audits and input (in our case, **ratings**) are likewise gathered which helps in knowing the fulfillment level of the client. A client who picks your item finished the contenders, that demonstrates the clients are steadfast and happy with your administrations, dependability can be utilized to judge the consumer loyalty. Along these lines, the steadfastness of the customer's, organizations increasing upper hand over the contenders and present different inventive items and administrations. [1]

II. RELATED WORK

In this section, we summarize and explain the approach followed in various research papers. Furthermore, the final result and any limitations have also been mentioned.

While paper [1] isn't necessarily a data analytics related paper, it sheds light on some of the important aspects of ecommerce with BigBasket in particular. With its notable mentions of consumer loyalty and audits and inputs, it only makes this paper of ours and its work more relevant. It talks about how ratings (referred in this paper as inputs) are important for the business and how they are effective parameters to measure consumer loyalty which in turn tells us how successful a business really is and is going to be.

Paper [2] named "Ridge Regression Learning Algorithm in Dual Variables" speaks about the model we successfully implemented in our project in full depth. In this project of ours, the model with the least error was Ridge Regression and it gave us highly accurate predictions. It speaks in detail of Ridge Regression itself while also educating the reader about how to use ANOVAs to better improve your Regression model. This related work was really helpful to us in our project.

Paper [3] speaks about Lasso Regression. While looking for a viable model for our project, we went into a spiral to reduce our prediction error by trying to eliminate the effects of multicollinearity and we landed on this concept of Lasso Regression. Paper [3] being concise and to the point really helped us understand what Lasso Regression is and how shrinking of values can really help reduce the prediction error. This really improved our prediction accuracy from our previous models, however, this model was immediately trumped by Ridge Regression model, which also focuses on reducing the effects of multicollinearity.

Paper [4] does a similar work on predicting ratings of products but takes a completely different aspect of customer inputs into consideration. It completely neglects the numerical (0-5) rating and bases their rating system on the descriptive review given by the customer. It uses an algorithm they call the FLRRRA algorithm. This algorithm takes the user descriptive reviews, does all the preprocessing to the data and then computes if the review is positive or negative using sentiment analysis on the term frequency parameter of the descriptive review. Post this, it predicts the numerical rating of products by their descriptive review. The primary assumption of this paper is that, customer numerical ratings might not be accurate their descriptive reviews tell us more about what they feel about the product.

III.PROBLEM STATEMENT

The dataset used in this project has been taken from Kaggle which is available at the following link:
https://www.kaggle.com/datasets/surajjha101/bigbasket-entire-product-list-28k-datapoints

Our problem statement is to evaluate the BigBasket dataset which consists of over 28,000 products being sold on their platform. Our goal with this evaluation is to effectively predict item ratings for certain category of items on the platform. With our EDA, we also try to summarize the most efficient revenue options for our case study subject, BigBasket.

A.   Dataset
This dataset contains 10 attributes with simple meaning and which are described as follows:

B. index - Simply the Index!

C. product - Title of the product (as they're listed)

D. category - Category into which product has been classified

E. sub_category - Subcategory into which product has been kept

F. brand - Brand of the product

G. sale_price - Price at which product is being sold on the site

H. market_price - Market price of the product

I. type - Type into which product falls

J. rating - Rating the product has got from its consumers

K. description - Description of the dataset (in detail)

Exploratory Data Analysis and Visualizations
The dataset has 27555 rows and 10 columns in total. Out of the 10 columns, it was found that 3 columns were of type float64, 6 column was of type object which was String and 1 column was type of int64. Takes us a total memory of 2.1+MB.
After data preprocessing it was found that while majority of the columns had no missing values in the data, the ratings

column had over 8626 missing values(index(0), product(1), category(0), sub_category(0), brand(1), sale_price(0), market_price(0), type(0), rating(8626), description(115)). On drawing a bar-plot we can see the effective variation of mean rating with category of item on the platform.
In Fig. 1., the bar-plot shows nearly similar rating means. However, some categories such as Baby Care, Beverages, Food grains, Oil & Masala show better rating averages as compared to the others
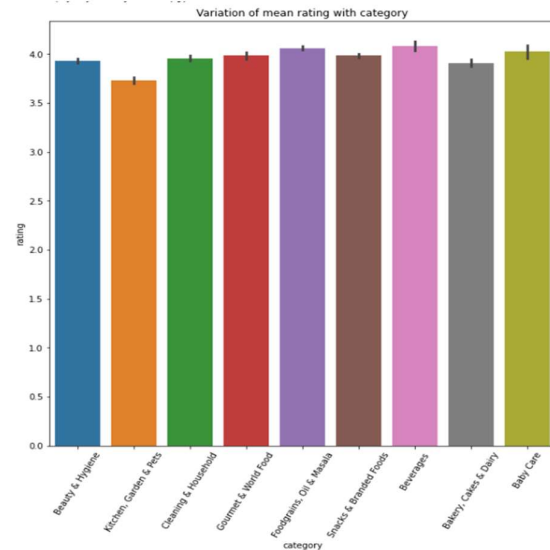

Fig. 1. Bar-plot (mean rating vs category)

In Fig. 2., the difference in the mean prices for each category can clearly be seen in this graph. Baby Care and Kitchen, Garden & Pets have much higher average prices for their products whereas, Snacks & Branded Foods and Bakery, Cakes & Dairy have much lower average prices.
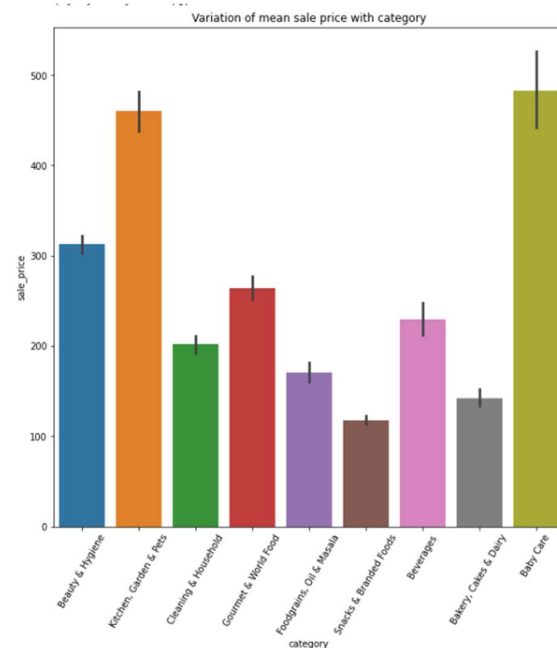

Fig. 2. Bar-plot (mean price vs category)

In Fig. 3., we can see a clear distribution of the number of customers and their ratings of the products on the platform. We can see that less than 25% of the people rate products less than 3.5 while more than 75% of the people rate products less than 4.5. Mean rating is around 4. We can infer from this that exceptionally high/low ratings are either an error from the consumer or a really good/bad product.
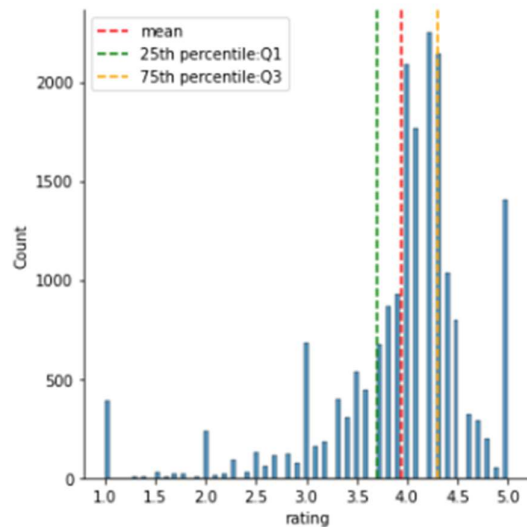

Fig. 3. Customer Rating (0-5)

To check whether rating varies with category, we perform ANOVA. From Fig. 4a., we can conclude that since the p-value is much lesser than 0.05 we can reject the null hypothesis i.e. we can say that the rating of the product **does** vary with category.

	sum_sq	df	F	PR(>F)
C(category)	165.972329	8.0	38.584027	1.943145e-61
Residual	10172.702175	18919.0	NaN	NaN

Fig. 4a. ANOVA

To check whether rating varies with sale price, we perform ANOVA. From Fig. 4b., we can conclude that Since the p-value is much lesser than 0.05 we can reject the null hypothesis i.e. we can say that the rating of the product **does** vary with sale price.

	sum_sq	df	F	PR(>F)
C(sale_price)	1787.490017	2446.0	1.40846	9.805353e-32
Residual	8551.184487	16481.0	NaN	NaN

Fig. 4b. ANOVA

In Fig. 5a. & 5b., we perform correlation analysis on the concerned attributes. We can infer from the plot that there is no linear correlation existent here.
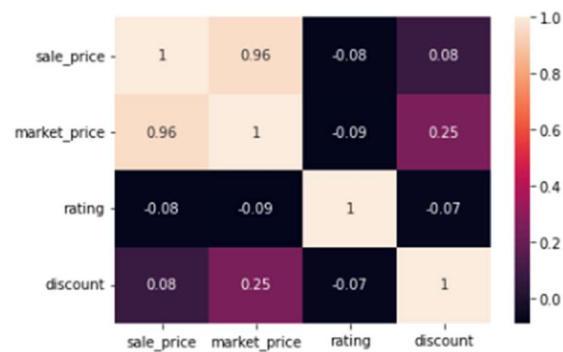

Fig 5a.

-0.0792820446662662

Fig 5b.

On running Spearman's correlation coefficient tests, we see that samples are correlated as in the case of rating and sale price as shown in Fig 6a. and from Fig 6b. we can see that samples are yet again correlated as in the case of rating and category of products.

```
Spearmans correlation coefficient: -0.049
Samples are correlated (reject H0) p=-0.049
```

Fig 6a. (rating & sale price)

```
Spearmans correlation coefficient: -0.057
Samples are correlated (reject H0) p=-0.057
```

Fig 6b. (rating & category)

As seen in Fig. 7., There is not much evident correlation between the selected features, apart from the obvious correlation between sale price and market price, which has a value of +0.97

Fig. 7.

Concluding from the EDA done and as represented in the Fig 8. (referring from but not only owing to) we can say that BigBasket must focus more of its selling to the categories of Food grains, Oil & Masala and Beverages as they not only show higher mean ratings but also pose significantly lower risk in terms of costs, inventory, storage and shelf life. On

the other hand, categories such as Kitchen, Garden & Pets should be given a backseat as their ratings are suffering compared to other categories and even their cost and pricing is more on the premium side which could mean higher risk for BigBasket in the long-term.
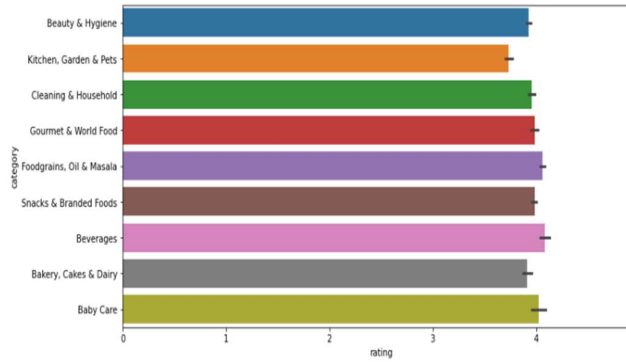

Fig 8.

IV. PROPOSED METHODOLOGY

After exploring the different models used, we have decided to implement the Lasso Regressive Model after comparing the accuracies of ANN, Decision Trees, Ridge Regression and MLR techniques.

A. MLR

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression is to model the linear relationship between the explanatory (independent) variables and response (dependent) variables. In essence, multiple regression is the extension of ordinary least-squares (OLS) regression because it involves more than one explanatory variable.

$$y^i = \beta^0 + \beta^1 x^{i1} + \beta^2 x^{i2} + \ldots + \beta^p x^{ip} + \epsilon$$

where, for **i=n** observations:
y^i=dependent variable
x^i=explanatory variables
β^0=y-intercept (constant term)
β^p=slope coefficients for each explanatory variable
ϵ= the model's error term (also known as the residuals)

In our case, owing to the presence of a categorical independent variable 'Category', we set dummy variables while attempting a regression model. We drop the sale price or market price column as they are highly correlated and there is no fear of lost data as we have already created a discount column.

B. Lasso Regression

The word "LASSO" stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. It is a statistical formula for the regularization of data models and feature selection. Regularization is an important concept that is used to avoid overfitting of the data, especially when the trained and test data are much varying.

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

"LASSO (Least Absolute Shrinkage and Selection Operator) regression, a shrinkage and variable selection method for regression models, is an attractive option as it addresses both problems"[3]

We will be testing this in our case to check to in the shrinkage of data values is meaningful or not and to see if it actually shows any improvement in the accuracy of the model.

C. Ridge Regression

Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values. The cost function for Ridge Regression:

$$Min(||Y - X(theta)||^2 + \lambda||theta||^2)$$

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

Since our dataset suffers substantially from multi-collinearity we can expect this model to give better results than that of the other regression models.

D. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

We will be trying to predict our ratings by running the parameters through a Decision Tree.

E. ANN

An artificial neural network has three or more layers that are interconnected. The first layer consists of input neurons. Those neurons send data on to the deeper layers, which in turn will send the final output data to the last output layer. All the inner layers are hidden and are formed by units which adaptively change the information received from layer to layer through a series of transformations. Each layer acts both as an input and output layer that allows the ANN to understand more complex objects. Collectively, these inner layers are called the neural layer.

The units in the neural layer try to learn about the information gathered by weighing it according to the ANN's internal system. These guidelines allow units to generate a transformed result, which is then provided as an output to the next layer.

In our case, we run the ANN model for 100 epochs.

V. EXPERIMENTAL RESULTS

A. MLR

Post running the MLR model on the test data, we check for multicollinearity. This is done by calculating the VIF values for all columns.

	VIF	Column
9	1.666173	Kitchen, Garden & Pets
4	1.539421	Beauty & Hygiene
1	1.174131	discount
8	1.104751	Gourmet & World Food
6	1.103807	Cleaning & Household
0	1.098214	sale_price
7	1.096549	Foodgrains, Oil & Masala
2	1.056409	Baby Care
10	1.040090	Snacks & Branded Foods
5	1.030441	Beverages
3	1.015672	Bakery, Cakes & Dairy

Fig 9.

The R-squared error is quite low, and there also appears to be no variables with multicollinearity.

```
OLS Regression Results
==============================================================================
Dep. Variable:               rating   R-squared:                       0.020
Model:                          OLS   Adj. R-squared:                  0.020
Method:               Least Squares   F-statistic:                     38.88
Date:              Mon, 14 Nov 2022   Prob (F-statistic):           1.53e-76
Time:                      15:43:23   Log-Likelihood:                -20942.
No. Observations:             18928   AIC:                         4.191e+04
Df Residuals:                 18917   BIC:                         4.199e+04
Df Model:                        10
Covariance Type:          nonrobust
=============================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------
sale_price               -0.0001   1.53e-05     -7.652      0.000      -0.000   -8.71e-05
discount                 -0.0018      0.000     -4.618      0.000      -0.003      -0.001
Baby Care                 4.0915      0.034    121.148      0.000       4.025       4.158
Bakery, Cakes & Dairy     3.9406      0.029    137.787      0.000       3.885       3.997
Beauty & Hygiene          3.9922      0.012    325.016      0.000       3.968       4.016
Beverages                 4.1282      0.030    139.819      0.000       4.070       4.186
Cleaning & Household      3.9991      0.017    238.360      0.000       3.966       4.032
Foodgrains, Oil & Masala  4.1009      0.016    250.608      0.000       4.069       4.133
Gourmet & World Food      4.0273      0.016    256.455      0.000       3.997       4.058
Kitchen, Garden & Pets    3.8326      0.019    203.233      0.000       3.796       3.870
Snacks & Branded Foods    4.0078      0.015    266.833      0.000       3.978       4.037
==============================================================================
Omnibus:                   6263.679   Durbin-Watson:                   2.026
Prob(Omnibus):                0.000   Jarque-Bera (JB):            22826.012
Skew:                        -1.648   Prob(JB):                         0.00
Kurtosis:                     7.252   Cond. No.                     2.91e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.91e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Fig 10.

As is evident from the accuracy values mentioned above as well as the plot drawn, the accuracy of this model is extremely low and is not viable for consideration.

MSE	0.540
MAE	0.4925626089454111

B. Lasso Regression

We now attempt the Lasso Regression. "LASSO regression aims to identify the variables and corresponding regression coefficients that lead to a model that minimizes the prediction error. This is achieved by imposing a constraint on the model parameters, which 'shrinks' the regression coefficients towards zero, that is by forcing the sum of the absolute value of the regression coefficients to be less than a fixed value (λ). In a practical sense this constrains the complexity of the model." [3]

```
lasso=Lasso(alpha = 0.001)
lasso.fit(x_train,y_train)

Lasso(alpha=0.001)
```

```
y_pred=lasso.predict(x_test)
y_pred[:20]

array([4.04095306, 3.74838159, 3.9784221 , 3.94053874, 3.94254936,
       3.99245467, 4.0401096 , 3.91929507, 4.07724728, 3.98683039,
       3.92825026, 4.09587811, 3.91356391, 3.9756651 , 4.00348266,
       3.95242992, 3.95308749, 4.07614351, 3.9756651 , 3.95630799])
```

```
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)

Accuracy of the model is 1.84
```

Fig 11.

The Lasso regression provided us with a subpar accuracy and extremely high RMSE. Hence, it was deemed unviable.

MSE	0.540
MAE	0.4925470852561415

C. Ridge Regression

Although our data did not show high levels of multicollinearity, Ridge Regression did fare as the best model giving us the lowest MSE of 0.540. "This allowed the algorithm (Ridge Regression) to overcome the "curse of dimensionality" and run efficiently, even though a very large number of parameters were being considered. Experimental results show that Ridge Regression performs well."[2]
MSE: 0.540
R^2: 0.018
Ridge Regression, although known to eliminate multicollinearity gave us very low error values, giving us highly accurate predictions. (our data did not have high multicollinearity as seen in the VIF values earlier)

```
pipeline = make_pipeline(StandardScaler(), Ridge(alpha=1.0))
pipeline.fit(x_train, y_train)
y_train_pred = pipeline.predict(x_train)
y_test_pred = pipeline.predict(x_test)
```

```
print('MSE train: %.3f, test: %.3f' % (mean_squared_error(y_train, y_train_pred),
                mean_squared_error(y_test, y_test_pred)))

print('R^2 train: %.3f, test: %.3f' % (r2_score(y_train, y_train_pred), r2_score(y_test, y_test_pred)))
```
```
MSE train: 0.534, test: 0.540
R^2 train: 0.020, test: 0.018
```

Fig 12.

MSE	0.540
MAE	0.492512114685379

D. Decision Trees

A Decision Tree regressor also proved to be unviable.

```
y = df2.iloc[:,[1]] # rating
x = df2.iloc[:,[0,2,3,4,5,6,7,8,9,10,11]] #remaining columns
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
```

```
DecisionTreeRegressor(random_state=0)
```

+ Code   + Markdown

```
Accuracy=r2_score(y_test,y_pred)*100
print(" Accuracy of the model is %.2f" %Accuracy)
```

```
Accuracy of the model is -29.11
```

Fig 13.

MSE	0.710
MAE	0.5515917687253344

E. ANN

The ANN model was suggested to us during our peer review and was coincidentally the model we had highest hopes for, however, noting no improvement in the loss function or accuracy after 10 epochs, the model was abandoned.

```
#Initialising ANN
ann = tf.keras.models.Sequential()
ann.add(tf.keras.layers.Dense(units=6,activation="relu"))
ann.add(tf.keras.layers.Dense(units=6,activation="relu"))
ann.add(tf.keras.layers.Dense(units=1,activation="sigmoid"))
ann.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])
```

```
ann.fit(x_train,y_train,batch_size=32,epochs = 10)
```

```
Epoch 1/10
474/474 [==============================] - 1s 1ms/step - loss: -33.9957 - accuracy: 0.0198
Epoch 2/10
474/474 [==============================] - 1s 1ms/step - loss: -574.0963 - accuracy: 0.0198
Epoch 3/10
474/474 [==============================] - 1s 1ms/step - loss: -2403.1296 - accuracy: 0.0198
Epoch 4/10
474/474 [==============================] - 1s 1ms/step - loss: -5993.2319 - accuracy: 0.0198
Epoch 5/10
474/474 [==============================] - 1s 1ms/step - loss: -11632.8877 - accuracy: 0.0198
Epoch 6/10
474/474 [==============================] - 1s 1ms/step - loss: -19534.7852 - accuracy: 0.0198
Epoch 7/10
474/474 [==============================] - 1s 2ms/step - loss: -29833.4023 - accuracy: 0.0198
Epoch 8/10
474/474 [==============================] - 1s 1ms/step - loss: -42666.9453 - accuracy: 0.0198
Epoch 9/10
474/474 [==============================] - 1s 1ms/step - loss: -58158.7344 - accuracy: 0.0198
Epoch 10/10
474/474 [==============================] - 1s 1ms/step - loss: -76436.6875 - accuracy: 0.0198
<keras.callbacks.History at 0x7fe568a98210>
```

Fig 14.

MSE	9.269
MAE	2.9526941362916004

F. Final Results

	MSE	MAE
MLR	0.540	0.4925626089454111
Lasso Regression	0.540	0.4925470852561415
Ridge Regression	0.540	0.492512114685379
Decision Trees	0.710	0.5515917687253344
ANN	9.269	2.9526941362916004

From the table above, we can notice that Ridge Regression collectively has the lowest MSE and MAE values making the most accurate model of the bunch.

	MSE	MAE
MLR	0.540	0.4925626089454111
Lasso Regression	0.540	0.4925470852561415
Ridge Regression	0.540	0.492512114685379
Decision Trees	0.710	0.5515917687253344
ANN	9.269	2.9526941362916004

Regression models were in general better performing compared to the others. In the regression models, Ridge Regression fared comparatively better and was our most accurate model tested.

VI. PEER REVIEW

For the peer review, we were paired with team 'DataDive'. They gave us some valuable feedback and suggestions which we took into consideration and then worked upon them.

Team DataDive suggested we try out ANN as a model for our project. This was a great advice and we implemented the model and ran it on our data. However, the accuracy of this model was really low and hence we had to scrap the model and try other models we had previously tested.

```
Epoch 10/10
474/474 [==============================] - 1s 1ms/step - loss: -76436.6875 - accuracy: 0.0
```

Fig 15.

Team DataDive also suggested we do Spearman's Correlation Coefficient as part of our EDA and it was a really helpful suggestion as we could reinforce the fact that three of our parameters in our data are correlated. This can be seen in Fig 6a. and 6b., the results of Spearman's Correlation Coefficient Test.

VII. CONCLUSIONS

We implemented and tested various models, namely, MLR. Lasso Regression, Ridge Regression, Decision Trees and ANN. Upon comparing the MSE values of all these models, we got a satisfactory and very low MSE value for the Ridge Regression Model with MSE as low as 0.540, which is a huge improvement in comparison to the other regression models that we tried. On the other hand, Decision Trees and ANN models suffered very high error values and were hence discarded from the experiment.
We would in the future like to venture out and try other models such as RNN and GRU to train our data on and test if the accuracy of these models fares any better than the ones stated above.
Concluding our experiment, we were able to predict ratings of products on BigBasket with very low error rates by correlating with multiple other parameters and building the most efficient model for it, which in our case is Ridge Regression.

VIII. CONTRIBUTIONS OF TEAM MEMBERS

All of the team members have put in an immense amount of effort to bring this project to the stage at which it is today. Each of us worked on separate parts with details as follows-
Aditya - EDA + Modeling
Rudransh – Report Curation + Literature Review & Related Work Study

Both of us were involved in choosing and evaluating the different areas and aspects of the dataset. Overall project plan was a team-effort as well.

REFERENCES

[1]   Ravi Naik, K.J., & Patel, Suhas,"A Study on Grocery's Online Shopping at Big Basket", Acharya Institute of Technology (2018)

[2]   C. Saunders, A. Gammerman, V. Vovk," Ridge Regression Learning Algorithm in Dual Variables", Appears in Proceedings of the 15th International Conference on Machine Learning, ICML '98. (1998)

[3]   J. Ranstam, J. A. Cook, "**LASSO regression** ", *British Journal of Surgery*, Volume 105, Issue 10, September 2018, Page 1348, (2018)

[4]   D.R. Kumar Raja, S. Pushpa, "Feature level review table generation for E-Commerce websites to produce qualitative rating of the products", Future Computing and Informatics Journal, Volume 2, Issue 2, December 2017, Pages 118-124 (2017)