Predictive And Prescriptive Analytics On Courses Offered By Udemy

Rahil N Modi
PES1201802826
Computer Science Engineering
PES University
rahilmodi123@gmail.com

Sooryanath.I.T
PES1201802827
Computer Science Engineering
PES University
sooryanathit@gmail.com

Himanshu Jain
PES1201802828
Computer Science Engineering
PES University
nhimanshujain@gmail.com

***Abstract*—MOOC provides a platform for learning at the user's convenience and at their pace. The right analysis of courses offered by these platforms help in improved learning experience and hence growth in one's career. This paper illustrates various kinds of data-driven analysis and insights that can be drawn from courses-related dataset. Experimental results suggest there is a shift towards MOOC platforms for a better learning and understanding of subjective and practical applications.**

***Index Terms*—Regression, Recommendation, Clustering, Descriptive Analysis, Visualization, Data-driven, Neural Networks, Matplotlib, Plotly**

I. Introduction

Analytics is the systematic computational analysis of data or statistics. It can be valuable in areas rich with recorded information; analytics relies on the simultaneous application of statistics, computer programming and operations research to quantify performance.

The major goal of performing analysis on the data is to be able to predict the patterns in the data and to be able to make informed and reasonable decisions. Predictive analytics is used to gain future knowledge about the patterns in the data and prescriptive analysis helps an entity modify their goals in order to maximize the profits.

The data in real-world is dynamic, large and inconsistent. The need for data analytics is at its peak and is helping various organizations in the decision-making process.

Educational sector of our country is expanding and the right path to establishing a strong foundation is by learning through the right courses and by the experienced and qualified instructors. The model building process of analyzing and prescribing the right courses at the right time is the heart of this project. The various factors affecting the popularity of an online course has to be analyzed through various statistical methods. Massive open online courses (MOOC) has gained importance in recent years. Students rely on these platforms to shape their careers and hence providing the right choice of the course to a student is very much essential.

The datasets used to examine the above problem consist of details regarding various domains of courses offered in Udemy MOOC. The domains are varied from IT and software related sectors to business, finance and accounting sectors. All this data has been accumulated from the academic year 2012 to 2020. The columns mainly include information on particular courses offered for each domain, average ratings, the estimated cost and discount if any offered to them. The predictive and prescriptive analysis would help us to understand the trend followed in Udemy courses over these years and what changes can be done to attract and fascinate more students over the next course of time.

The current dataset can be used to comparatively analyse different courses opted by students in a given range of academic years. Based on exploratory data analysis and visualization a number of patterns can be recognised on which courses students opt, what are trending topics in the market, which topics require more practical exercises and how are students utilizing the knowledge gained through these topics in the industry.

Visualization mainly includes different charts and plots done using R and Python to analyze the data in the dataset and infer on summaries and statistics accumulated over time. Few basic visualisations like bar plots, histograms, pie charts help us to understand the distribution while a few other charts like whisker plots, scatter plots help us to identify outliers and correlation between attributes in the dataset. Many sophisticated charts like Coxcomb plot, strip plot, multihist, word cloud, etc also disclose few hidden patterns and characteristics.

Many other techniques like dimensionality reduction using wavelet transform, Principal Component Analysis(PCA) and Singular Value Decomposition(SVD) can be done to identify and remove irrelevant attributes and reduce the problem to lower dimensions. These techniques form the heart and soul of predictive and prescriptive analysis. Once the data is known carefully and completely a few queries would help to evaluate whether the data gained regarding the dataset is sufficient to perform analysis or further survey is required to reach a possible solution.

II. Related Works

A. Analytics Curriculum for Undergraduate and Graduate Students

1) Assumptions: This study aims to develop an analytics curriculum for undergraduate and graduate programs by identifying skill-based gaps between industry and academia and then clustering them based on methodological and semantic similarities among other criteria. [1] Specifically, we compare industry requirements and related skills for analytics jobs to three types of analytics domains, that is, descriptive, predictive, and prescriptive analytics.

2) Approach: This paper relates to data and the part of the problem we are trying to solve that is predicting a few interesting courses offered by Udemy to students at a reasonable cost. It also helps Udemy to formulate a series of courses offered as a package to students so that they can get a marginal profit and students get a whole specialization in the domain selected. This paper aims to formulate a curriculum to undergraduate and postgraduate study in a manner that best serves the interests of the students. Finally, realizing the significant difference between undergraduate and graduate students in terms of expectations and maturity, we use personality-job fit theory to recommend strategies to better promote the field to undergraduate students.

3) Results: Defining a curriculum for undergraduate and graduate students taking descriptive, predictive, and prescriptive analysis into consideration. Personality job-fit theory adds on to this to make wonderful predictions and prescriptive analysis.[1]

B. Machine Learning Approaches to Predict Learning Outcomes in Massive Open Online Courses

1) Assumptions: The implementation and methodology take into account only behavioral data when investigating the effect of patterns in learner behavior on the user certification rate, and latent variables like emotional state are ignored. Two set experiments have been performed. In the first set of experiments, all features from the dataset were included. For the second, a subset of features is segmented by their weights/importance in accordance with the target variable and selected.

2) Approach: Related to educational data mining (EDM) on courses taken/offered, no_of_tests, duration of lectures, correlation/relationship between clickstreams, and courses completed. Knowledge extraction to enhance teaching strategies. Also emphasizes on introduction about the domain of learning Analytics that discusses crowd behaviors, experiences that contribute towards making significant decisions on MOOCs. In general, nonlinear classifiers have better accuracy in both experiments than the linear classifier. This indicates the nonlinear form of correlation between the predictor features and target in the learner dataset. Two set experiments have been performed. In the first set of experiments, all features from the dataset were included. For the second, a subset of features is segmented by their weights/importance in accordance with the target variable and selected.

3) Results: Choosing or adapting boosting methods and feature segmentation by ranking / assigning weights tends to deliver a better performance in case of predictive analysis, class balancing plays a major role in performing a predictive analysis, functional form between the target and predictor features need to be in line with the choice of models chosen for prediction (linear or non-linear). The simulation results in experiments indicate that Random Forest (RF) and Support Vector Machine (SVM) achieved ideal performance, with the accuracy values of 0.9881 and 0.9851 respectively.[2] Latent variables like the emotional state of a learner also need to be considered to perform a more accurate predictive analysis.

4) Limitations: The result shows average run time of machine learning models is much longer in performed experiments.[2] The learner emotional states of students are not considered here, which can be inferred from their interaction with online courses over time.

C. MoocRec: Learning Styles-Oriented MOOC Recommender and Search Engine

1) Assumptions: It uses (Felder and Silverman Learning Style Model) FSLSM to recommend courses. This model indexes only computer science courses. [3]

2) Approach: Massive Open Online Courses provide a large number of courses in different domains. A specific domain has multiple courses. Selecting the most suitable course based on factors like learning styles, individual needs, course quality makes a difference in effective learning. MoocRec model has been developed for personalized learning. It performs content analysis of MOOC data to find the best course which satisfies the learning style of the user. Recommending courses based on specific topic parameters is implemented in the MoocRec model for course recommendations.

3) Results: The major takeaway for this model is the use of topic modeling and text processing to filter most appropriate courses based on the user's search query.

4) Limitations: The current version of MoocRec is only limited to the Felder and Silverman learning style model. similarly, MoocRec currently indexes only computer science courses from edX and Coursera platforms.[3]

III. Proposed Problem Statement

MOOC aims to provide access to numerous amount of courses at a reasonable cost, here we try to address the methods by which the course instructors or the course providers can increase the number of course recipients or subscribers in an intuitive manner by introducing short and crisp lecture series and offering courses at a reasonable discount when compared to the competitive course providers, probably by increasing the assessment contents etc. What is the optimal number of lecture hours, assessments per course and duration of the courses to increase the ratings and the number of recipients for a course, such questions can be answered with appropriate analytics done on the set of features from the training dataset.

Capture the features for the courses with high subscriber count and ratings and analyse the reason (causation) behind the factors which are responsible for such numbers (a heuristic approach). Given the key performance parameters or indicators (KPI) for a course, how many subscribers may choose it (A regression analysis)? Suggest the user with top 10 highly rated/attempted courses when given with a category of course. A correlation analysis based on how every single KPI affects the overall ratings for a given course. how to predict the optimal cost for a course given its features like total lecture hours and assessments, this information will help the user set up the near to optimal cost for a course.

Perform various visualizations to find the strength of the relationship between various features of a course and find out what are the most commonly offered courses based on their Titles, what are the catchy phrases that must be present in the title of a course to attract the learners and so on. Provide insights on what a learner looks for and what must a course provider look to provide.

IV. Proposed Solution

A. Data Integration and Preprocessing

In the integration stage, the datasets are imported and merged if their schemas correlate with each other. This is an important task as much of preprocessing is needed. In our case, we are integrating two datasets with the same schema into one. Once the integration is done then all feature extraction techniques like replacing null values, removal of duplicates, remove redundant columns, etc., are done.

All columns with null values in the merged dataset are found. Appropriate methods are used to fill these values so that they can increase the efficiency of models when used further in runtime. Example: In the Development_IT dataset there was a column 'discount_price__amount' which had null values. It indicated the reduced amount per course from 'price_detail__amount'. It had null values for all those courses with no discount, hence we imputed the same value as 'discount_price__amount' into it so that people should pay the same amount without any discount.

After data integration, there were many duplicates seen in the data. All these were found and reduced to one. This way null values were imputed and duplicates were removed. Once preprocessing was done, the next step was to retain columns with maximum variation and remove all the redundant columns. To identify the redundant columns we performed an unsupervised learning technique called PCA(Principle Component Analysis) in R. This gave us a clear idea of all columns that have redundant information and can be removed. This way the dataset was preprocessed and merged before models were applied on it.

B. Simple Linear Regression

Modelling the response in terms of the num-subscribers for a course with respect to the num-reviews received for a course. The dependent feature here is the num-subscribers attribute whereas the num-reviews is the predictor.Attribute 'num-subscribers' denotes the number of people who have subscribed to the course, num-reviews denote the total number of ratings for a course and not the ratings themselves. These both serve as the KPI for the business value of a course on Udemy.

The reason to choose num-reviews as the predictor is because of best correlation coefficient (r) value with respect to the target variable. Before we adopt linear - regression model, we must verify the functional form between the interacting variables. A scatter plot between the num-reviews and num-subscribers is used to verify the functional form, which turned out be a linear relation as we can see a linear pattern. Hence, we proceed further with the linear regression model.

Data pre-processing is performed on the input data, before fitting it to the linear regression model. Outliers from the input data are removed and n samples are obtained. Training and Test validation sets are obtained and the model is fit onto input data. As result we obtain the intercept and co-efficient (slope) value for the given input data. Predictions are then performed on the test data. Evaluation metrics such as R-square, F-statistic, Adjusted R-square are then computed. Then we plot the residual plots to verify normality conditions.Even after eliminating outliers we see long tailed distribution indicating that the data is meant to have such values too.

From exploratory data analysis results we see that number of ratings for a course can hugely impact the target subscribers, this is quite an intuitive assumption and it happens to be the same in real world too.

C. Multi Linear Regression

It is a type of regression analysis where one variable is dependent on multiple independent variables. Further, the relationship between the dependent variable and model co-efficients must be linear while the relationship between the dependent variable and the independent variables can be non-linear too. It involves a few definite steps which are to be followed:

1) Collect or Extract data from the dataset.
2) Preprocess the data
3) Perform descriptive analytics
4) Decide on the modelling strategy
5) Divide the data into training, testing and validation data
6) Define the functional form of the regression.
7) Estimate the regression parameters
8) Perform regression model diagnostics.
9) Validate the data using validation data.

All these 9 steps are followed while creating a model on the multilinear regression model. In our case, we are trying to predict 'price_detail__amount' based on 'num_subscribers' and 'num_reviewers'. All initial steps are preprocessing, descriptive analytics where we check if the data follows the assumptions required to perform multilinear regression or not. If it is not satisfying then some adjustments are to be done. Once everything is fine the model is trained and then tested on test split of data. Metrics like the coefficient of determination (R-square), Durban Watson Test, T-test, AIC, BIC, etc are used to find the goodness of the model. In our case, the multi-linear model didn't perform well on data as it had a lot of outliers hence we went to non-linear models.

D. K-Nearest Neighbours

K-nearest neighbours approach to predict and classify if a course is paid or un-paid given the num-of-lectures published and the num-of-practice-tests for the courses. The heuristic approach here is that , when a course is about to enter the production phase the course providers can make decision on lending the course for free or for a priced amount based on the factors of practice tests and lectures in the course (these are known apriori).

Hence we model this as a K-NN problem in a 2D vector space . Based on the classification of K -nearest neighbours with respect to a sample input , we can determine to which class the course belongs to based on the similarity measures. The Dataset is created with equal split of target classes along with random sampling .

We set the hyper-parameters for K-NN model as specified below

1) K = 5 neighbours
2) Similarity metrics = Minkowski's Distance
3) p = 2

After fitting the model, we move onto make predictions with test data. Accuracy is used as the evaluation metric here because the dataset under consideration is equally split on the target class. Results are then visualized using the confusion matrices. We perform the elbow method of finding the optimal value of K, the K for which mispredictions are low. After performing the elbow method, we obtain the optimal value of K. Thus, based on this result, the course producers can predict the combination of total lectures and total practice tests for which the courses get paid or get lend out for free.

E. Artificial Neural Network

Artificial Neural Network (ANN) is a supervised non-linear model which consist of multiple layers each layer consisting of many perceptrons. Each perceptron consists of input, associated weights and bias, summation part, non-linear activation part and output. Weights and bias are hyperparameters which can be tuned to get minimum error function. There can be three types of layers: input layer, hidden layers and output layer. All these layers together make am ANN. Further, there is a loss function(error) associated with ANN to measure its goodness or correctness. In our case we have used ANN in two different forms:

1) ANN to predict a continuous target variable (regression type non-linear model).
2) ANN to predict a multi-classification problem where the target variable is categorical.

ANN was used to predict 'avg_rating' which is a continuous variable using 'num_subscribers', 'num_ratings' and 'num_published_lectures' as inputs. K-fold cross-validation is used to train the model to avoid overfitting. The model architecture is as follows:

1) Input layers - dimension = 3
2) Dense Layer - dimension = 128
3) Dense Layer - dimension = 64
4) Dense Layer - dimension = 32
5) Output Layer - dimension = 1

Loss function used is MSE (mean Square Error) and the metric used is MAE (Mean Absolute Error). Finally coefficient of determination was used to find the efficiency of the model.

For the multi-classification ANN model, 'num_published_practice_tests' was used as a target variable with 'ratings', 'num_reviews' and 'num_published_tests' as input. The target variable has 7 categorical data. One hot encoding technique is used to encode them. K-fold cross-validation is used to prevent the overfitting of the model. The architecture of the model is as follows:

1) Input Layer: dimension - 3
2) Dense Layer - dimension = 24
3) Dense Layer - dimension = 8
4) Output Layer - dimension = 7

The loss function used here is categorical cross-entropy and metric used is accuracy. Confusion matrix and many metrics like precision, recall, F1-score, etc are derived to check the goodness of the model. ANN being a non-linear model yielded good results than linear models hence outperformed all linear models.

F. Clustering

Clustering is one of the unsupervised learning technique used to group similar records into groups called clusters. The records are mapped to points in an n-dimensional space where n - number of features. A similarity metric is used to check the closeness of points from each other to group them into clusters. There are many clustering methods and based on the type of clustering method other parameters are defined for the model. We have used three types of clustering here:

1) K-means Clustering
2) Agglormerative Clustering
3) DBSCAN (Density-Based Spatial Clustering With Noise) Clustering

K-means Clustering is a type of clustering where the user explicitly specifies the k value which is the number of clusters. Based on the closest centroid(mean), the points are classified into the corresponding cluster. This type of clustering though is prone to outliers and cannot handle differently sized and density clusters.

In Agglomerative Clustering, there is no need to specify the number of clusters instead, the dendrogram (plot drawn while clustering) helps to get any number of clusters. It just considered all points as single clusters and starts merging cluster till a single cluster is formed. By cutting the dendrogram at the appropriate position we can obtain the required number of clusters. This approach is better than K-means but doesn't take density into consideration.

Next type of clustering used was DBSCAN Clustering which takes density into consideration for clustering. It also nullifies the effect of noise or outliers if present and gives us precise clusters. Only two parameters, radius(Eps) and Minimum Points is to be passed. The DBSCAN algorithm itself will form the best number of clusters based on density. Concept of core point, border point and noise point are used to divide all points into clusters.

In our implementation, we have used ‘discount_price__amount’ and ‘price_detail__amount’ as features to cluster so that we can find the schemes used to give a discount from the original amount throughout all courses. All three types of clustering are used to find the most optimal type of clustering pattern for it.

G. Recommender System

Collaborative filtering models are deployed by the organizations to know better about the user’s ratings and item preferences. An item-based recommender provides a comprehensive list of top-k similar items.

The initial step involves identifying important keywords from the title of the course. This is required to generate a similarity vector for any given course. The keywords are extracted by generating a MEGA_LIST, which is a list of all possible programming languages, frameworks, and other important keywords, and comparing it with the course title attribute of the dataset. This results in a “Category” column which is used to find similar courses.

The user provides an input of their desired course. The system generates all similar courses that the user might want to explore. This is done using cosine similarity between courses where every course title represents a vector in n-dimensional space. Further filtering of only relevant courses is done by looking for keywords in the title and comparing them with the Category column of other courses.

The recommender system also outputs the prediction of the rating of the input course. This prediction is done by taking the average of all the ratings of the similar courses given in the previous step. The evaluation of this prediction done by comparing it with the true ratings obtained from the dataset. The absolute percentage error is proved to be minimal and hence proves the efficiency of the recommender system in correctly identifying similar courses.

V. Experimental Results

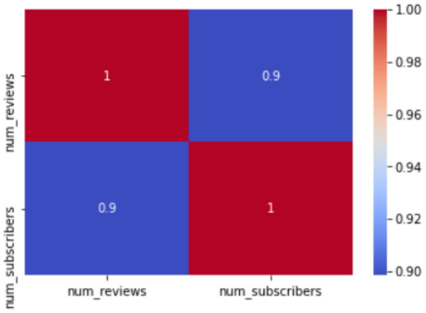
A. Simple Linear Regression

The SLR model , outputs decent results and confirms that our heuristic about relationship between num_reviews and num_subscribers is true .The Fig 1. shows the correlation between the interaction variables .

Fig 2. shows the evaluation metrics for the SLR model like R-square value , Adjusted R-square etc . The value of R-square indicates that the variation in the num_subscibers value is explained by the num_reviews attribute for 80.7% of the times.

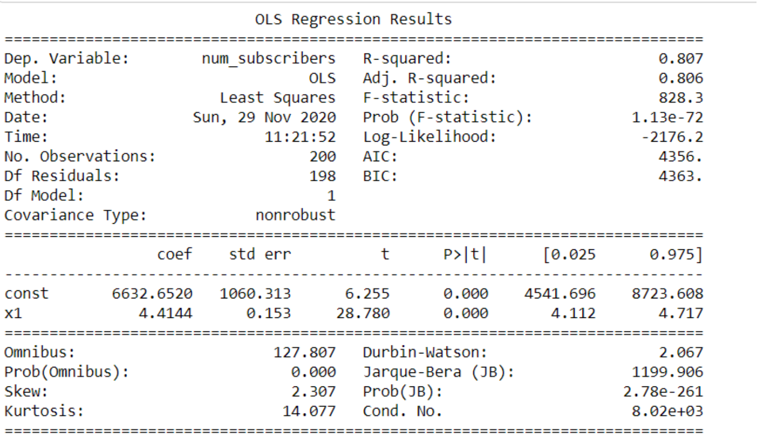
The co-efficient (slope) indicates that the number of subscribers increase by a rate of 4.441 times for unit increase in the number of reviews.

Fig3. shows the regression plot obtained . Fig4 indicates that the residuals obtained after the test phase are nearly following a normal distribution.



A correlation plot between the interaction variable (response and predictor)

Fig. 1. Correlogram between feature and Target variables



```
                            OLS Regression Results
==============================================================================
Dep. Variable:        num_subscribers   R-squared:                       0.807
Model:                            OLS   Adj. R-squared:                  0.806
Method:                 Least Squares   F-statistic:                     828.3
Date:                Sun, 29 Nov 2020   Prob (F-statistic):           1.13e-72
Time:                        11:21:52   Log-Likelihood:                -2176.2
No. Observations:                 200   AIC:                             4356.
Df Residuals:                     198   BIC:                             4363.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       6632.6520   1060.313      6.255      0.000    4541.696    8723.608
x1             4.4144      0.153     28.780      0.000       4.112       4.717
==============================================================================
Omnibus:                      127.807   Durbin-Watson:                   2.067
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1199.906
Skew:                           2.307   Prob(JB):                    2.78e-261
Kurtosis:                      14.077   Cond. No.                     8.02e+03
==============================================================================
```

Fig. 2. Evaluation metrics for the SLR Model

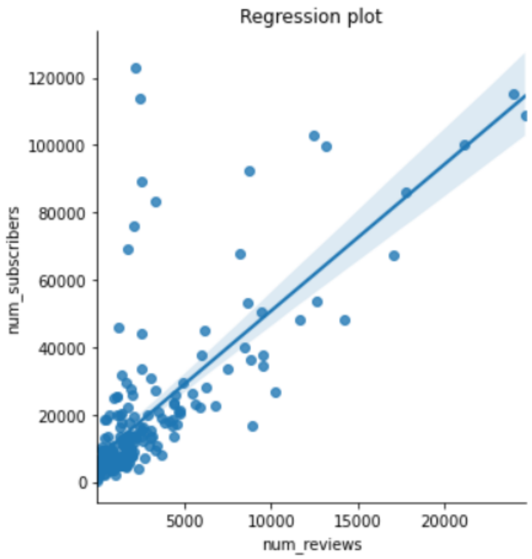


Fig. 3. Regression plot between feature variable and target variable

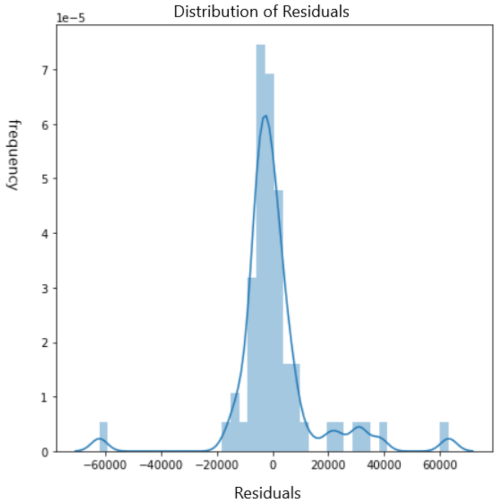
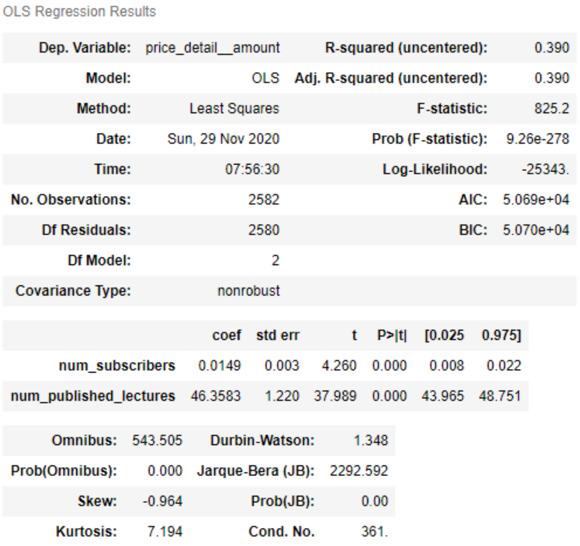


Fig. 4. Residual distribution plot after prediction phase

B. Multi Linear Regression

The model didn't fit as expected due to the presence of outliers. This could be interpreted by low R-squared value in the below figure5. Trying different combinations of dependent and independent variable didn't work. Hence we went to non-linear models.



OLS Regression Results

Dep. Variable:	price_detail__amount	R-squared (uncentered):	0.390
Model:	OLS	Adj. R-squared (uncentered):	0.390
Method:	Least Squares	F-statistic:	825.2
Date:	Sun, 29 Nov 2020	Prob (F-statistic):	9.26e-278
Time:	07:56:30	Log-Likelihood:	-25343.
No. Observations:	2582	AIC:	5.069e+04
Df Residuals:	2580	BIC:	5.070e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
num_subscribers	0.0149	0.003	4.260	0.000	0.008	0.022
num_published_lectures	46.3583	1.220	37.989	0.000	43.965	48.751

Omnibus:	543.505	Durbin-Watson:	1.348
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2292.592
Skew:	-0.964	Prob(JB):	0.00
Kurtosis:	7.194	Cond. No.	361.

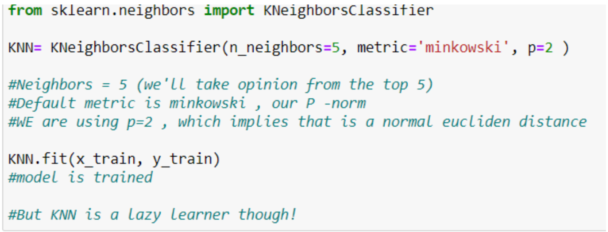
Fig. 5. Results of Multi-Linear Regression Model

C. K-Nearest Neighbours

K-NN predicts the class of the input instance based on the Minkowski's distance similarity. The fig 6. depicts the hyper parameters used in the K-NN classifier model .

Since KNN is a lazy learner , most of the computation is done only when the test instance arrives.The test phase results are then displayed using the confusion matrix . The metrics corresponding to the binary classifier are displayed , accuracy or F1 score would be appropriate metrics in this case as the target class is well balanced.

The elbow method is run for 10 values of k between 1 to 11 and it is observed that 5 is the optimal value for k with less value of mean prediction error.



```
from sklearn.neighbors import KNeighborsClassifier

KNN= KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2 )

#Neighbors = 5 (we'll take opinion from the top 5)
#Default metric is minkowski , our P -norm
#WE are using p=2 , which implies that is a normal eucliden distance

KNN.fit(x_train, y_train)
#model is trained

#But KNN is a lazy learner though!
```

Fig. 6. Hyper parameters for K-NN Classification

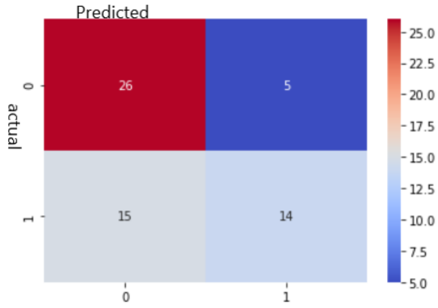
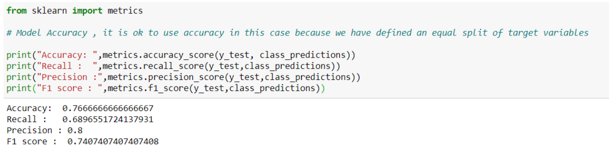


Fig. 7. Confusion matrix for test data



```
from sklearn import metrics

# Model Accuracy , it is ok to use accuracy in this case because we have defined an equal split of target variables

print("Accuracy: ",metrics.accuracy_score(y_test, class_predictions))
print("Recall :  ",metrics.recall_score(y_test,class_predictions))
print("Precision :",metrics.precision_score(y_test,class_predictions))
print("F1 score : ",metrics.f1_score(y_test,class_predictions))
```

```
Accuracy:  0.7666666666666667
Recall :   0.6896551724137931
Precision : 0.8
F1 score :  0.7407407407407408
```

Fig. 8. K-NN classification metrics for K=5 and binary target class

D. Artificial Neural Network

ANN gives better result than other supervised learning models like Multi-Linear Regression (MLR) and K-Nearest Neighbours (KNN). Further it can efficiently predict continuous and categorical target variables. It outperforms all other models if appropriate architecture is used. Figure10 and figure11 represent results of continuous and categorical target variables.

E. Clustering

Clustering is mainly used here to infer patterns out of 'discount_price__amount' and 'price_detail__amount'. We can fairly understand the difference between 3 types of clustering by seeing their graphs. The below figure13 represents dendrogram for agglomerative clustering.

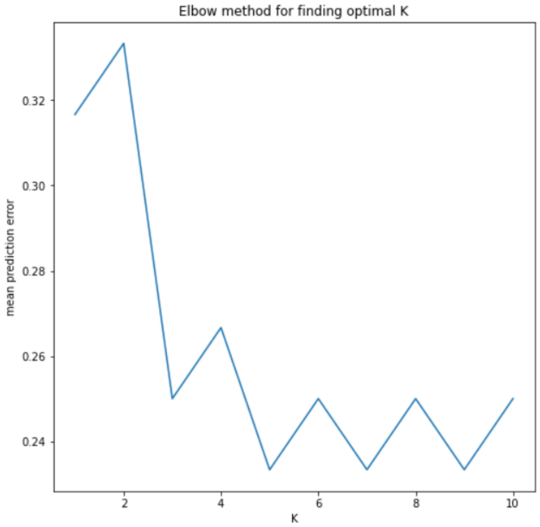
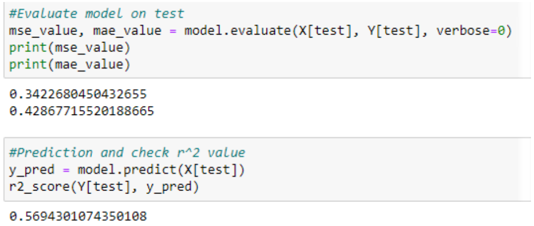


Fig. 9. Elbow method to find the optimal K



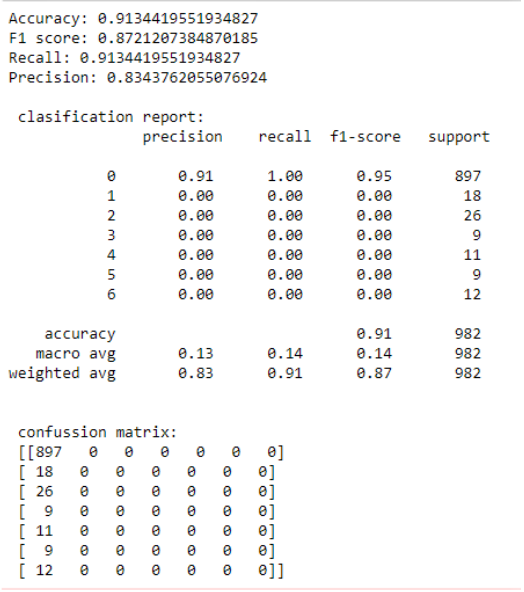
```
#Evaluate model on test
mse_value, mae_value = model.evaluate(X[test], Y[test], verbose=0)
print(mse_value)
print(mae_value)
```

```
0.3422680450432655
0.4286771552018866 5
```

```
#Prediction and check r^2 value
y_pred = model.predict(X[test])
r2_score(Y[test], y_pred)
```

```
0.5694301074350108
```

Fig. 10. Artificial Neural Network for continuous target variable



```
Accuracy: 0.9134419551934827
F1 score: 0.8721207384870185
Recall: 0.9134419551934827
Precision: 0.8343762055076924

 clasification report:
              precision    recall  f1-score   support

           0       0.91      1.00      0.95       897
           1       0.00      0.00      0.00        18
           2       0.00      0.00      0.00        26
           3       0.00      0.00      0.00         9
           4       0.00      0.00      0.00        11
           5       0.00      0.00      0.00         9
           6       0.00      0.00      0.00        12

    accuracy                           0.91       982
   macro avg       0.13      0.14      0.14       982
weighted avg       0.83      0.91      0.87       982


 confussion matrix:
 [[897   0   0   0   0   0   0]
 [ 18   0   0   0   0   0   0]
 [ 26   0   0   0   0   0   0]
 [  9   0   0   0   0   0   0]
 [ 11   0   0   0   0   0   0]
 [  9   0   0   0   0   0   0]
 [ 12   0   0   0   0   0   0]]
```

Fig. 11. Artificial Neural Network for categorical target variable

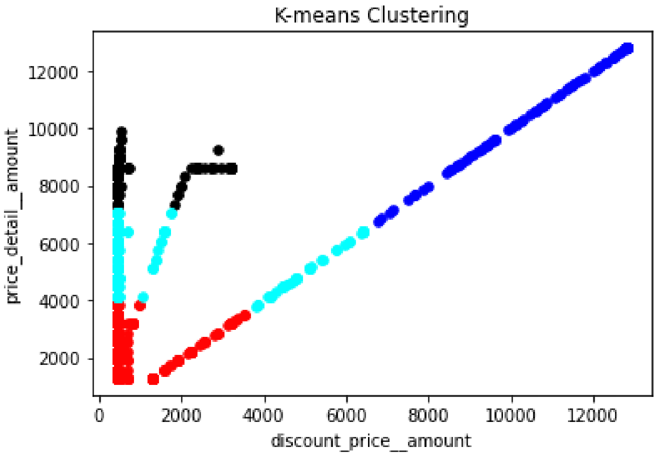


Fig. 12. K-Means Clustering

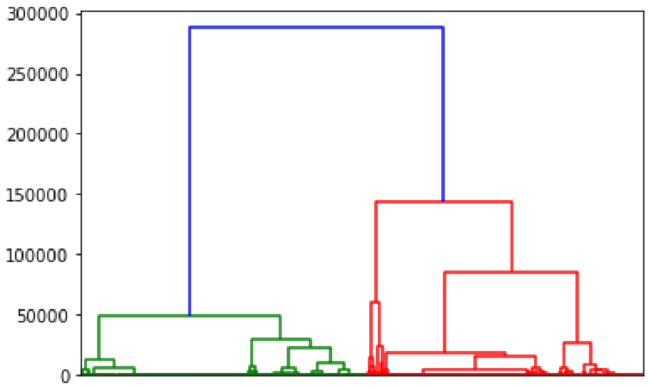


Fig. 13. Dendrogram for Agglomerative Clustering

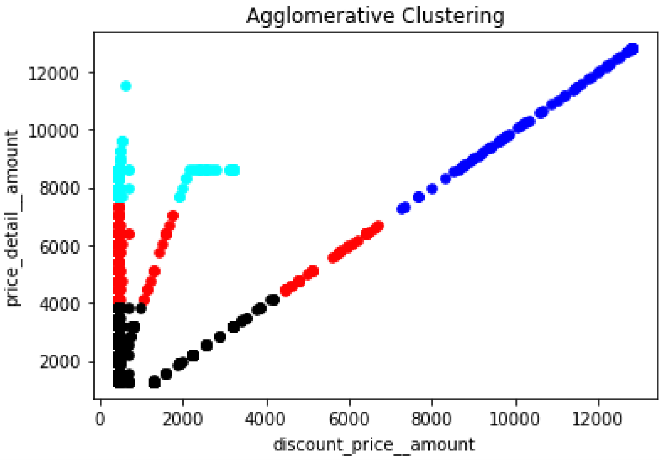


Fig. 14. Agglomerative Clustering

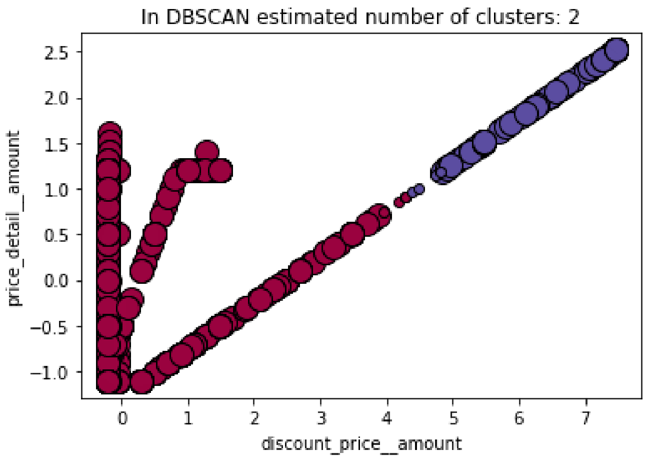


Fig. 15. DBSCAN Clustering

F. Recommender System

The item-based collaborative filtering model provides a list of similar courses that the user could explore to satisfy their requirements. The similar courses help the user to evaluate how good other courses are and can decide on which course will satisfy their requirements.

The evaluation of this recommender system is done using the absolute percentage error. This metric helps us understand what the user might rate a particular course. Ratings play an important role in correctly identifying the user's state of mind and thus helps the instructor of the course to improve their content which can ultimately improve the total subscribers to the course.

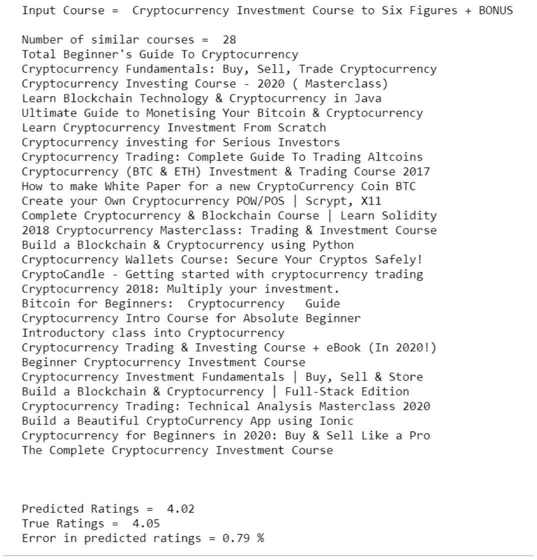
The models works extremely well given the Category column contains the keywords. When no keywords are extracted for a particular course, the model will not be able to select that course.

Figure 16 shows the course suggestions given to a user based on his query. Also, the ratings for the course are predicted.

VI. Conclusion

Prescriptive and Predictive Analysis mainly involves all those methods which would predict and add on value, somehow to the field in which it is been done. We have taken the MOOC courses data to do the analysis. Using all possible models like supervised , linear, non-linear, unsupervised and recommended systems we have tried to predict all those variables which would assist course developers and students to find out what would be profitable to them. MOOC designers can learn how they can combine courses and introduce new courses to gain the attraction of students. Students can understand which courses to selected in the interested domain.

The present model is restricted to software and IT courses. It can be further extended to courses of different domains. User specifications are not involved in the used dataset if they are available then a full fledged recommended system could be constructed.



```
Input Course =  Cryptocurrency Investment Course to Six Figures + BONUS

Number of similar courses =  28
Total Beginner's Guide To Cryptocurrency
Cryptocurrency Fundamentals: Buy, Sell, Trade Cryptocurrency
Cryptocurrency Investing Course - 2020 ( Masterclass)
Learn Blockchain Technology & Cryptocurrency in Java
Ultimate Guide to Monetising Your Bitcoin & Cryptocurrency
Learn Cryptocurrency Investment From Scratch
Cryptocurrency investing for Serious Investors
Cryptocurrency Trading: Complete Guide To Trading Altcoins
Cryptocurrency (BTC & ETH) Investment & Trading Course 2017
How to make White Paper for a new CryptoCurrency Coin BTC
Create your Own Cryptocurrency POW/POS | Scrypt, X11
Complete Cryptocurrency & Blockchain Course | Learn Solidity
2018 Cryptocurrency Masterclass: Trading & Investment Course
Build a Blockchain & Cryptocurrency using Python
Cryptocurrency Wallets Course: Secure Your Cryptos Safely!
CryptoCandle - Getting started with cryptocurrency trading
Cryptocurrency 2018: Multiply your investment.
Bitcoin for Beginners:  Cryptocurrency   Guide
Cryptocurrency Intro Course for Absolute Beginner
Introductory class into Cryptocurrency
Cryptocurrency Trading & Investing Course + eBook (In 2020!)
Beginner Cryptocurrency Investment Course
Cryptocurrency Investment Fundamentals | Buy, Sell & Store
Build a Blockchain & Cryptocurrency | Full-Stack Edition
Cryptocurrency Trading: Technical Analysis Masterclass 2020
Build a Beautiful CryptoCurrency App using Ionic
Cryptocurrency for Beginners in 2020: Buy & Sell Like a Pro
The Complete Cryptocurrency Investment Course



Predicted Ratings =  4.02
True Ratings =  4.05
Error in predicted ratings = 0.79 %
```

Fig. 16. Item-Based Collaborative Filtering

References

[1] Paul, Jomon A., and Leo MacDonald. ”Analytics Curriculum for Undergraduate and Graduate Students.” Decision Sciences Journal of Innovative Education 18.1 (2020): 22-58.

[2] R. Al-Shabandar, A. Hussain, A. Laws, R. Keight, J. Lunn and N. Radi, ”Machine learning approaches to predict learning outcomes in Massive open online courses,” 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 713-720, doi: 10.1109/IJCNN.2017.7965922.

[3] S. Aryal, A. S. Porawagama, M. G. S. Hasith, S. C. Thoradeniya, N. Kodagoda and K. Suriyawansa, ”MoocRec: Learning Styles-Oriented MOOC Recommender and Search Engine,” 2019 IEEE Global Engineering Education Conference (EDUCON), Dubai, United Arab Emirates, 2019, pp. 1167-1172, doi: 10.1109/EDUCON.2019.8725079.

Individual Contributions

A. Common Tasks

- Exploratory Data Analysis
- Data Visualization
- Data Integration
- Feature Extraction

B. Rahil N Modi

- Multi Linear Regression
- Artificial Neural Network
- Clustering

C. Sooryanath.I.T

- Simple Linear Regression
- K-Nearest Neighbours and its Analysis
- Statistical Testing of Models

D. Himanshu Jain

- Item-Based Collaborative Filtering Recommender System
- Naive Bayes Text Classification
- Basic Regression Analysis