Udemy Course Subscribers

Predicting Number of Subscribers for an Udemy Course

Sarah Fetter
School of Engineering
University of St. Thomas
St. Paul MN USA
fett6874@stthomas.edu

Paul Mcgie
School of Engineering
University of St. Thomas
St. Paul MN USA
pmcgie@stthomas.edu

Robin Aherns
School of Engineering
University of St. Thomas
St. Paul MN USA
robin.aherns@stthomas.edu

ABSTRACT

Udemy is a Massive Open Online Course platform (MOOC). It offers both paid and free courses. Udemy has over 35 million subscribers, as well as over 57,000 instructors teaching over 130,0000 courses. Both businesses and individuals are able to use Udemy to give their employees opportunities to develop and improve various skills (Udemy, 2020). The dataset we have chosen contains records of 3,678 Udemy courses across four different subject areas: Business Finance, Graphic Design, Musical Instruments, and Web Design. This dataset was downloaded from the website Kaggle, created by a user named Chase Willden, who is associated with the Concept Course, a business that works to provide educational content. Willden created this dataset with analytics in mind, and therefore there are no null values. Willden removed columns that contained null values and aggregated values into a single csv file. This introduces a possible bias into our results, as there is not a record of what columns Willden removed or how he aggregated the results. We picked this dataset because MOOCs are becoming increasingly popular ways for companies to offer additional trainings for their employees, and it will be interesting to see what variables are correlated with the more popular courses.

This model will be a regression, in which we attempt to predict the number of subscribers (num_subscribers) based on variables such as whether or not the course costs money(is_paid, categorical), the cost of the course (price, quantitative), , the number of lectures in the course (num_lectures, quantitative), the level of the course (level, categorical), the course content duration (content_duration, quantitative), the time stamp associated with when the course was published (published_timestamp), and the subject associated with the course (subject, categorical). While there are 12 attributes in this data set, there are three that will not be able to be used in a regression: the course ID (course_id), course title (course_title), and the course url (url). These variables contain unique categorical values, and therefore cannot be used as is in a regression model, as this will add far too many dimensions to the regression without providing much value. The course_title attribute, however, will be used to create two new sentiment metrics: subjectivity and polarity. This dataset has 4 categorical variables that we will explore, which range from 2-4 categories each. This data also has 6 numerical variables that we will examine in the various regression models that we explore.

1 Data Prep 1

In creating a boxplot of the num_subscribers variable, we see there are outliers. In fact, the range of the outlier values is so large that it is impossible to see the boxplot itself (Figure 1).

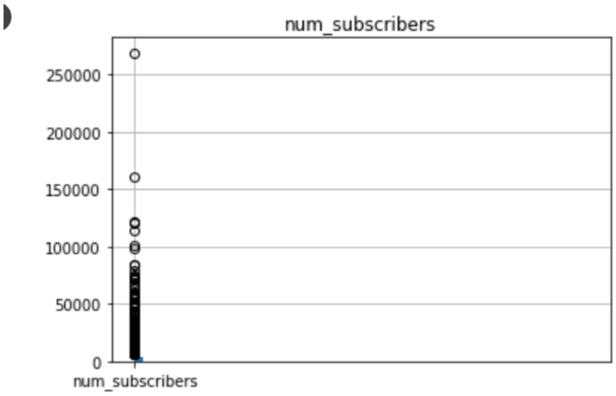


Figure 1: Boxplot of Data before outliers dropped

The third quartile value is 2,546. Using the standard Q3+1.5*Q3 formula to identify outliers, there are 427 values above the outlier identifier value of 6,365. After we drop these outliers, the boxplot is now represented in Figure 2 below:

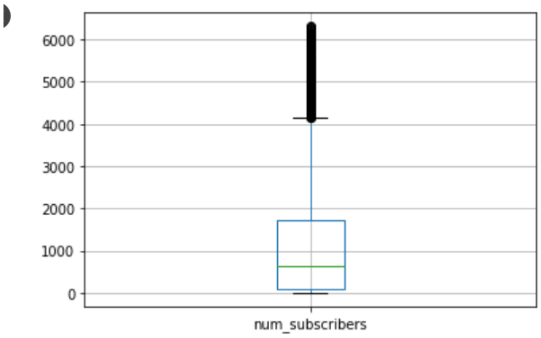


Figure 2: Boxplot of Data after outliers dropped

The next step in data preparation is to remove the url column, as every value is unique, so it will not be useful in a regression model. We then extract the date and time from the time stamp column in order to create season and time_of_day attributes. Season contains four values: ‘Summer’, ‘Fall’, ‘Winter’, and ‘Spring’, which are allocated based on the month. Time_of_day contains two categories: ‘Morning’ and ‘Afternoon’. The extracted timestamp columns of month, hour, and published_timestamp are then dropped, and the columns are re-ordered so that num_subscribers column is located at the end of the dataset.

We then perform one-hot encoding and drop the first dummy variable for each of our categorical variables, create the calculated value avg_content_duration_per_lecture, and then drop the columns num_reviews, course_title and course_id.

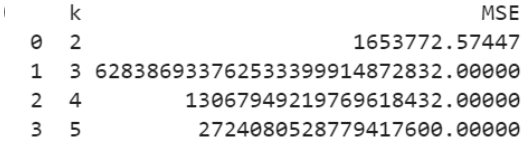
We also add in two sentiment metrics to the dataset, looking at polarity and subjectivity. These two metrics are created by looking at the course_description field. We believe that proper naming of the course, even from a marketing standpoint could increase subscribers. In doing so, we leverage a python library called TextBlob. For the polarity column, a polarity of zero that references a negative statement whereas a polarity of one reference a positive statement. For the subjectivity column, a subjectivity of zero refers to a statement that is more objective whereas a subjectivity of one refers to a more subjective course title.

Finally, we split the data into a training set and a test set, with the test set containing 30% of our data, and normalize the data in order to ensure it is ready to be fit into our predictive models.

2 Models on Original Dataset 1

We will first perform the following models on the original and complete dataset: Polynomial Regression, Regularization with Ridge, Regularization with Lasso, Regularization with Elastic Net, Multiple Linear Regression, K-fold, Normal Regressions, Decision Tree Regression, and Random Forest Regression

2.1.1 Polynomial Regression. We ran the polynomial regression on the original dataset, with degree values of 2-5. As seen in Figure 3 below, although all the degree values (k) have a large Mean Squared Error (MSE), the Mean Squared Error grows significantly larger when using higher than a degree 2 polynomial regression. Overall, we draw the conclusion that due to a larger Mean Squared Error, polynomial regression is not a good fit for predicting the number of subscribers.



```
      k                                MSE
  0   2                     1653772.57447
  1   3 628386933762533399914872832.00000
  2   4        13067949219769618432.00000
  3   5         2724080528779417600.00000
```

Figure 3: Polynomial Regression MSE outputs for different degrees k

2.1.2 Regularization with Ridge.   The first of the regularized regressions ran on the model was Ridge regression. Ridge regression constrains the coefficients and normalizes them. Per Bhattacharyya, “ridge regression shrinks the coefficients and helps to reduce the model complexity and multi-collinearity” (Bhattacharyya, 2020). Running Ridge regression with all the variables and an alpha value of .01 gives the training set score (the r^2 value) of 0 .275 and the test set score of 0.245. The mean squared error is 1,546,940.45 for an alpha value of .01 and is 1,545,776.22 for alpha value of 100. In this case, increasing the alpha increased the r^2 on the test set slightly lowered the mean squared error of the model. On the below plot (Figure 4), for most of the coefficients the magnitude of the coefficient did not change greatly between linear regression and ridge regression regardless of the alpha values of .01 and 100.

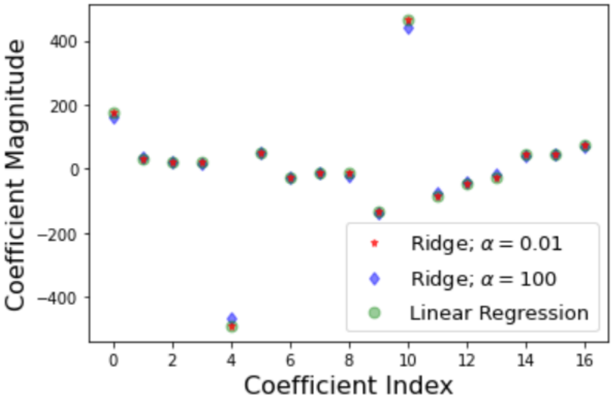


Figure 4: Ridge Regression Coefficient Magnitude

2.1.3 Regularization with Lasso.   Lasso regression was also completed on this dataset. While lasso regression is similar to ridge regression, the main difference is that in lasso regression some of the coefficients can end up at zero. This further helps with model selection and helps to avoid overfitting. Running lasso on the entire dataset with no change to alpha, we get a score(r^2) of 0.275 for training, 0.245 for test and number of features selected is 17. We can see that as we increase the alpha it does not really change the r^2 value for the models. When alpha is .01, the training r^2 is 0.275153992 and the test r^2 is 0.2450751664. An alpha value of .0001 has an r^2 of 0.275153993 for training and 0.2450758 for test, while straight linear regression gives an r^2 of 0.275153993 for training and 0.24507508 for test. In comparing the mean squared error on all the models, it can be seen that changing the alpha does have an immaterial affect the Mean Squared Error. For Lasso, without limiting the alpha we get a Mean Squared Error of 1,546,967.259, whereas alpha of .01 is 1,546,940.53 and alpha of .0001 gave a Mean Squared Error of 1,546,940.69. There is minimal difference between .01 and .0001.

It can be seen on the below plot (Figure 5) that the coefficients do not change greatly given the change in alpha.

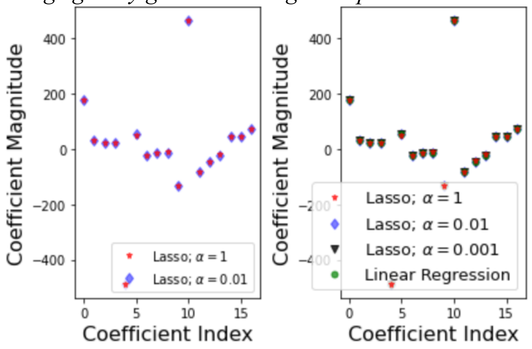
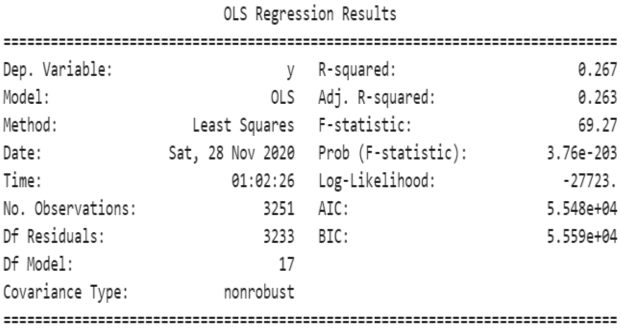


Figure 5: Lasso Regression Coefficient Magnitude

2.1.4 Regularization with Elastic Net. The final regularization used is the Elastic Net, which is a combination of Ridge and Lasso regularization regressions. Running elastic net with an alpha equal to 1 gives a Mean Squared Error of 1,590,279.63. An alpha value equal to .01 gives a Mean Squared Error of 1,546,679.71, and an alpha equal to .001 gives a Mean Squared Error of 1,546,912.90. Similar to Lasso, the lowest Mean Square Error is for an alpha equal to .01

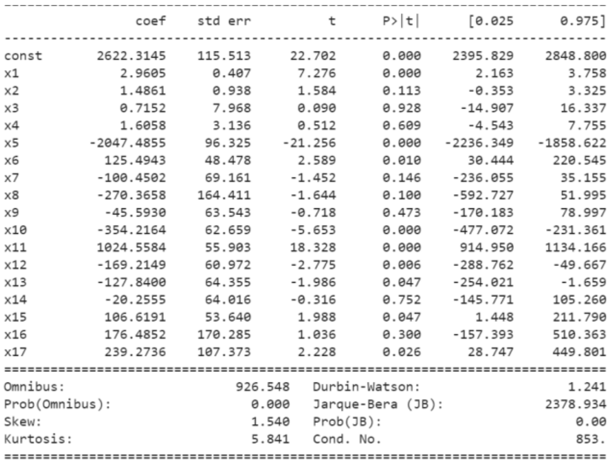
2.1.5 Multiple Linear Regression. For multi-linear regression, when using all attributes, we get an r^2 score of .267 and an adjusted R-squared of .263, which indicates a low level of correlation and accuracy in our prediction.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.267
Model:                            OLS   Adj. R-squared:                  0.263
Method:                 Least Squares   F-statistic:                     69.27
Date:                Sat, 28 Nov 2020   Prob (F-statistic):          3.76e-203
Time:                        01:02:26   Log-Likelihood:                -27723.
No. Observations:                3251   AIC:                         5.548e+04
Df Residuals:                    3233   BIC:                         5.559e+04
Df Model:                          17
Covariance Type:            nonrobust
==============================================================================
```

Figure 6: Multiple Linear Regression Result Output

Looking at the first regression results (Figure 7), attributes price, is_paid_True, level_Beginner_Level, subject_Musical Instruments, subject_Web_Development, season_spring, season_Summer, time_of_day_Morning, and subjectivity as indicated by p-values would appear to be the most optimal attributes.



```
------------------------------------------------------------------------------
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       2622.3145    115.513     22.702      0.000    2395.829    2848.800
x1             2.9605      0.407      7.276      0.000       2.163       3.758
x2             1.4861      0.938      1.584      0.113      -0.353       3.325
x3             0.7152      7.968      0.090      0.928     -14.907      16.337
x4             1.6058      3.136      0.512      0.609      -4.543       7.755
x5         -2047.4855     96.325    -21.256      0.000   -2236.349   -1858.622
x6           125.4943     48.478      2.589      0.010      30.444     220.545
x7          -100.4502     69.161     -1.452      0.146    -236.055      35.155
x8          -270.3658    164.411     -1.644      0.100    -592.727      51.995
x9           -45.5930     63.543     -0.718      0.473    -170.183      78.997
x10         -354.2164     62.659     -5.653      0.000    -477.072    -231.361
x11         1024.5584     55.903     18.328      0.000     914.950    1134.166
x12         -169.2149     60.972     -2.775      0.006    -288.762     -49.667
x13         -127.8400     64.355     -1.986      0.047    -254.021      -1.659
x14          -20.2555     64.016     -0.316      0.752    -145.771     105.260
x15          106.6191     53.640      1.988      0.047       1.448     211.790
x16          176.4852    170.285      1.036      0.300    -157.393     510.363
x17          239.2736    107.373      2.228      0.026      28.747     449.801
==============================================================================
Omnibus:                      926.548   Durbin-Watson:                   1.241
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2378.934
Skew:                           1.540   Prob(JB):                         0.00
Kurtosis:                       5.841   Cond. No.                         853.
==============================================================================
```

Figure 7: Multiple Linear Regression Coefficient Detail

2.1.6 K-fold. When using k-fold validation, we used the validation process with a GridSearchCV machine learning algorithm. In doing so, we ran the model with five splits and graphically showed the number of optimal features to select. In finding those optimal features, we ran a recursive feature elimination process and used r-squared as a scoring mechanism. From this standpoint, we see that explained variance really comes from five features as pictured below in Figure 8:

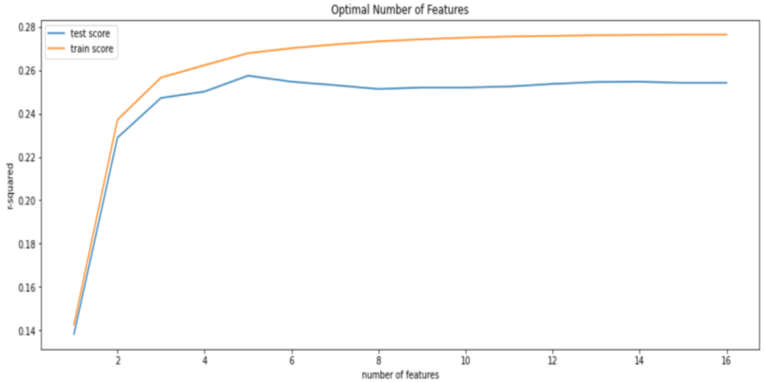
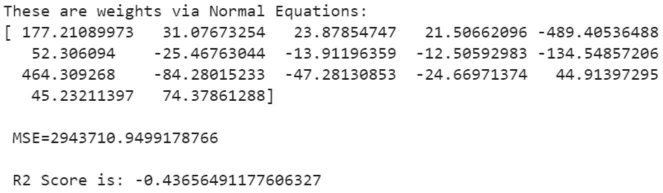


Figure 8: Features as Hyperparameter and GridSearchCV

2.1.7 Normal Regression. After running normal equations, we see that the weights are most prominent around x1, x5, x10,x11, which equates to the features of price, whether the course is paid or not, and whether the subject is musical instruments or web development. As seen below in Figure 9, the respective weights are as follows:



```
These are weights via Normal Equations:
[ 177.21089973   31.07673254   23.87854747   21.50662096 -489.40536488
   52.306094    -25.46763044  -13.91196359  -12.50592983 -134.54857206
  464.309268    -84.28015233  -47.28130853  -24.66971374   44.91397295
   45.23211397   74.37861288]

 MSE=2943710.9499178766

 R2 Score is: -0.43656491177606327
```

Figure 9: Normal Equation Weights

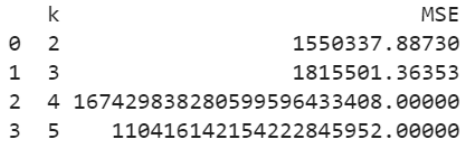
2.1.8 Decision Tree Regression. After running the decision tree regression with a criterion of Mean Squared Error to evaluate the quality of the split, we end up with an r^2 value of 0.245075 and a MSE of 3,090,883.86. Due to the high Mean Squared Error and relatively low r^2 value, we can determine that the Decision Tree Regression does not do an optimal job of predicting the number of subscribers.

2.1.9 Random Forest Regression. After running the Random Forest Regression, the resulting regression gives a Mean Absolute Error of 934.43 degrees, an r^2 score of 0.179696, and uses a total of 49 estimators. With an r^2 of only 0.179696, this model is worse than the regularized regressions and the above decision tree regression, as this r^2 value is lower than the other models.

3 Dimensionality Reduction 1

Next, we explore backwards elimination as a form of dimensionality reduction. After 9 iterations, the remaining variables were: 'price', 'num_lectures', 'is_paid_True', 'level_Beginner Level', 'subject_Musical Instruments', 'subject_Web Development', 'season_Spring', 'season_Summer', and 'subjectivity'.

3.1.1 Polynomial Regression. In running polynomial regressions with degree values of 2-5 on the backwards elimination dataset, as seen in the below table (Figure 10), all the degree values (k) have a large Mean Squared Error (MSE). As in the polynomial regressions on the entire dataset, a second-degree polynomial has the lowest Mean Squared Error. Unlike in the first set of polynomial regressions, the third-degree polynomial Mean Squared Error is only marginally higher than the second-degree polynomial Mean Squared Error. Regardless, these large Mean Squared Error values still indicate that polynomial regression is not an optimal model for predicting the number of subscribers.



```
   k                          MSE
0  2                1550337.88730
1  3                1815501.36353
2  4 167429838280599596433408.00000
3  5    110416142154222845952.00000
```

Figure 10: Backwards Elimination Polynomial Regression MSE outputs for different degrees k

3*.1.2 Regularization with Ridge. Ridge regression was run on the remaining variables after backwards elimination was completed. For an alpha of .01, the r^2 is only 0.27296 for the training and 0.24395 for the test set. Running alpha at 100 only slightly lowered the training r^2 for a score of 0.27247 and for test it slightly increased to a value of it is now 0.24426. The Mean Squared Error for the test set with alpha of .01 is 1,549,248.869 and for an alpha of 100 only slightly lower at 1,548,613.315. The results are worse than when all the variables being used. Based on the graph in Figure 11, we see that the ridge with alpha of 100 shows coefficients slightly different from linear regression, whereas an alpha of .01 is very close to the linear regression coefficients.*

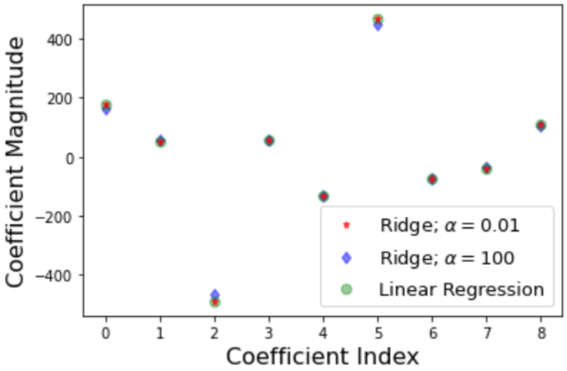


Figure 11: Regularization with Ridge after Backwards Elimination

3.1.3 Regularization with Lasso. Lasso regression after performing backwards elimination also appears to give worse results than when using all variables. The r^2 for the training when not specifying an alpha value is only 0.27295 and the test r^2 is only 0.24388 using 9 features. When changing alpha to .01, the r^2 is only 0.27296 for the training and 0.24395 for the test. Further reducing the alpha value to .0001 gives an r^2 of only 0.27296 and for the test set it is 0.24395. Comparing it to the normal linear regression r^2, there is limited improvement. The Mean Squared Error on all the Lasso regression runs is high. For the regression run with no specified alpha, the Mean Squared Error is 1,549,397.127, for an alpha of .001 the Mean Squared Error is 1,549,249.057 and Mean Squared Error for alpha of .01 is 1,549,250.445. As seen in the graphs in Figure 12, there is not much difference between the coefficients for all the lasso regression models versus the linear regression model.

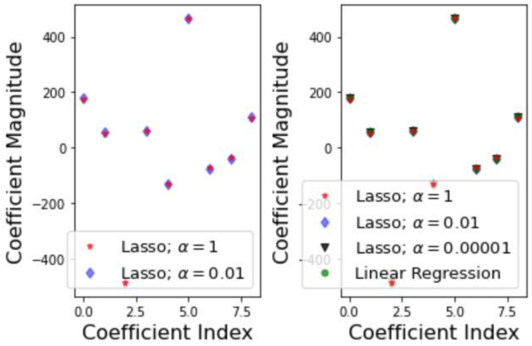


Figure 12: Regularization with Lasso Coefficient Magnitude after Backwards Elimination

3.1.4 Regularization with Elastic Net. *Elastic Net was also run after backwards elimination. With an alpha value of 1, the regression has a Mean Squared Error of 1,595,486.641, an alpha value of .01 has a Mean Squared Error of 1,549,067.906, and an alpha value of .001 has a Mean Squared Error of 1,549,229.524. In this case an alpha of .01 has the lowest r^2 value, however, it is still higher than the elastic net that was run with all the variables versus only the attributes determined after backwards elimination.*

3.1.5 Multiple Linear Regression. *For multi-linear regression, when using all attributes, we get an r^2 score of 0.265 and an adjusted R-squared of .263, which indicates a low level of correlation and accuracy in our prediction. Looking at the initial regression results, attributes price, is_paid_True, level_Beginner_Level, subject_Musical Instruments, subject_Web_Development, season_spring, season_Summer, time_of_day_Morning, and subjectivity as indicated by p-values would appear to be the most optimal attributes. In breaking it down further, our model tells us that generally, or at least from this dataset our intercept is of 2,622 subscribers (rounding up). Furthermore, we can see that for every dollar increase we would expect an increase of 2.96 subscribers. With every lecture added within the course content, we would see an increase of 1.63 subscribers. Also, if the course is classified as paid, we see a dramatic drop of 2,051.92 subscribers.*

For a beginner level course, we are likely to see a boost in 149.4 subscribers, but if the subject is based on musical instruments, we would see a decline of 340.43 subscribers. It is a better idea to have a course subject that is centered around web development as would indicate a boost of 1,035.64 subscribers. The time of year a course is posted is also impactful, as having it in the spring will likely lead to a loss of 171.10 subscribers, and posting a course in the summer is likely to have a decrease of 129.25 subscribers. Finally, it seems like people prefer course titles that appear more subjective. If you had a course title that appears 100% subjective (a value of 1), you will likely see an increase in 320.45 subscribers. Overall, while there is still a great deal of unexplained variance, the low p-value does tell us that the features mentioned do influence the dependent variable. With that said, we can explain much of our r-squared output from these 9 features.

3.1.6 K-fold. *In re-running k-fold validation with GridSearchCV with our new set of variables, we see a slightly different story. We see that all features have an impact, but there are two that have the most impact. In the end, we find that by setting up features as the hyperparameter and plotting the results, that 22.8% of the explained variance comes from two features, ispaid_true and subject_webdevelopment.*

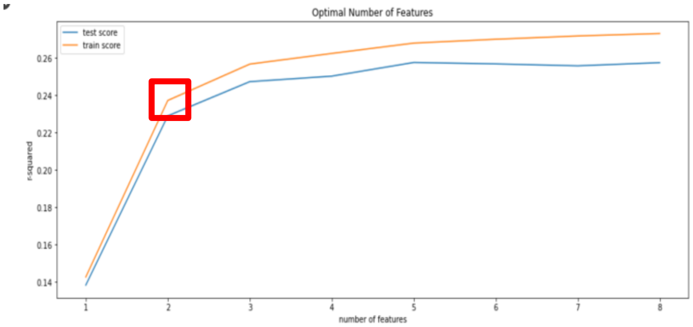
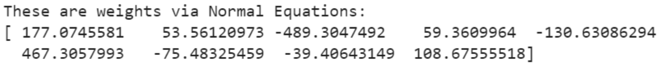


Figure 13: Features as Hyperparameter and GridSearchCV after Backwards Elimination

3.1.7 Normal Regression. *After running normal equations, we see that our weights are most prominent around x3, x6, which equates to the features whether it is paid or not, and whether the subject is web development. This is an interesting find, as this backs up the k-fold GridSearchCV analysis.*



```
These are weights via Normal Equations:
[ 177.0745581    53.56120973 -489.3047492    59.3609964  -130.63086294
  467.3057993   -75.48325459  -39.40643149  108.67555518]
```

Figure 14: Normal Equation Weights

3.1.8 Decision Tree Regression. *After running the decision tree regression on the attributes from the backwards elimination dimensionality reduction with a criterion of Mean Squared Error to evaluate the quality of the split, we end up with an r^2 value of -0.37489 and a MSE of 2,817,339.5280. Due to the high MSE and negative r^2 value, we can determine that the Decision Tree Regression does not do a great job of predicting the number of subscribers.*

3.1.9 After running Random Forest Regression after backwards elimination, the Mean Absolute Error is 1152.33 degrees while the R2 score is -0.3879.

3 Dimensionality Reduction 2

Our next attempt at dimensionality reduction was by using PCA. PCA rotates the dataset in such a way that the new rotated features are statistically uncorrelated. We ran our dataset through a loop that looked at several different n-component *values, ranging from 1 to 17, to determine which n-component value gave the highest r^2 value. As seen in figure 12 below, running PCA with 12 n-components gives us an r^2 value of 0.237, after which we see minimal improvement in the r^2. Using 12 components, however, does not do much to reduce dimensionality. Based on the below figure, using 7 n-components gives an r^2 of .18046, thus doing a better job of reducing the dimensionality while maintaining a decent size of the r^2 value.*

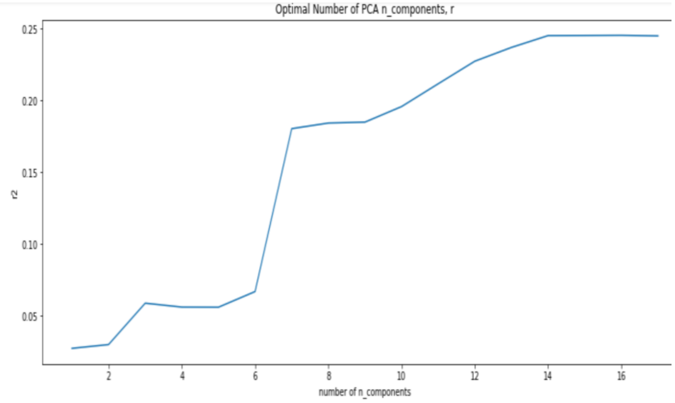


Figure 15: Optimal Number of n-components for PCA dimensionality reduction, r^2

In looking at the Mean Squared Error, we see similar results. A n-component value of 7 gives us a Mean Squared Error of 1,679,353.33008, and that is still the most significant decrease in Mean Squared Error.

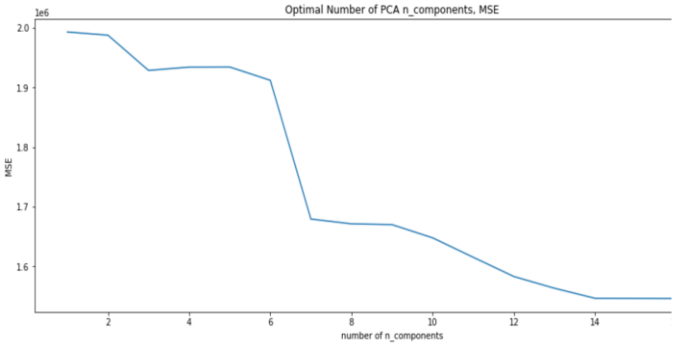


Figure 16: Optimal Number of n-components for PCA dimensionality reduction, MSE

3 Dimensionality Reduction 3

As the third form of dimensionality reduction, Kernel Principal Component Analysis (KPCA) was run to examine if a non-linear transformation would fit the data. After running KPCA with different number of components the ideal kernel number was 10, which gave a Mean Squared Error of 1,638,054.14737. While admittedly this is not an optimal score, it does give a lower Mean Squared Error than using the normal equations regression method did so it does appear that it did improve the model in some aspects. However, given that the optimal number of components was 10, this does not appear to do much in terms of dimensionality reduction, as is does not eliminate many of the components overall.

4 Conclusion 1

Overall, model accuracy and r-squared was less than optimal. As with any data science project though, it usually leads to more questions than answers, and a need to explore further to understand what really drives the predictions. Our Udemy dataset was limited in certain attributes, but if you are really trying to understand how to increase subscribers, we feel that other models and additional data would help.

For example, with more data, if you had individual users and courses in which they subscribed to, association rule learning would be a great analysis. We believe the correct pairing of content would supplemental your subscribers on a per course level and provider greater lift. Furthermore, if we were to expand our dataset and pull in ratings from other users, we could build out content-based or collaborative filtering recommendations for users. Other information that would provide further insight would be more related to engagement, as it may be beneficial to determine how long is the viewer engaged with the content before turning it off. Positive reviews and positive engagement would provide greater insight into how to set up class structure in future offerings. In fact, looking at the Udemy API, you see completion_ratio as a possible feature to extract. All in all, we went with a dataset pulled from Kaggle, which provided a cleaner dataset, but disregarded multiple attributes that could have been extracted from Udemy's API.

The models we built featured data at a higher level, as we were looking at data by course. It would likely be better to have access to data at a transactional, user level. Even within our course-level analysis we might want to segment further to find an optimal price point, eliminating non-paid courses and seeing how we could maximize our financial gain pairing price and subscribers. Other avenues to explore are to segment out the objects within our course titles to identify what specific topics or skillsets people want to learn. For example, SQL courses may get more subscribers than python courses, or guitar lesson courses may get more subscribers than harmonica lesson courses.

In the end, none of models outshined the others, one of our best models was the multi-linear regression model generated after backwards elimination with an accuracy of 24.4% and an r-squared of .265. In several cases, we received a negative r^2 score, which means that it does not follow a trend line at all, so it fits worse than a horizontal line. With that said, as mentioned earlier, we did derive that much of our explained variance comes from two features, ispaid_true and subject_web_development and have a substantial impact on subscriber volumes compared to the other features.

REFERENCES

[1] Chase Willden. 2020. Udemy Courses. (May 2020). Retrieved November 1, 2020 from https://www.kaggle.com/andrewmvd/udemy-courses

[2] Saptashwa Bhattacharyya. 2020. Ridge and Lasso Regression: L1 and L2 Regularization. (September 2020). Retrieved December 5, 2020 from https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

[3] Daniel Pelliccia. 2020. Principal Component Regression in Python. (September 2020). Retrieved December 5, 2020 from https://nirpyresearch.com/principal-component-regression-python/

[4] Udemy. 2020. Learn about Udemy culture, mission, and careers: About Us. (September 2020). Retrieved December 9, 2020 from https://about.udemy.com/?locale=en-us