EEE 490: Introduction to Big Data Analytics

Graduate Admission Prediction

Student Name: Bekassyl Syzdykov (ID: 201584803)

Symbat Kurasbek (ID: 201498156)

Course Instructor: Amin Zollanvari

Course TA: Meruyert Temirbekova

Nur-Sultan

2019

Abstract

Nowadays, lots of students are willing to continue their studies to get degrees beyond the bachelor's degree. This is a tough process, which requires a lot of time and effort. Moreover, students are spending their money to take different standardized tests required for graduate degree admission, such as IELTS, TOEFL, GRE, GMAT, and etc. In addition, even just to apply for a graduate degree application fee is unavoidable for every applicant, and it can range individually based on the requirements of universities.

However, this is not the only problem students are facing, while applying for a graduate degree. Students have to report a list of documents including GRE result, IELTS/TOEFL result, Letter of Recommendation, Statement of Purpose, Research output and etc. Based on the results of each submitted document, universities make their final decision.

The main aim of this study is to find out the impact of each feature (submitted documents) on the final decision of the university and to predict the probability of being admitted. Therefore, a number of machine learning models will be constructed to predict the chance of admittance based on the documents that students report to the admission committee.

Introduction

Every student who is interested in deepening the knowledge acquired during Bachelor's degree experiences all difficulties associated with application for a Master's degree. These difficulties may include anxiety, study stress during the preparation for standardized tests, continuous effort to maintain high GPA, mental health issues which are followed by sleep difficulties, and obsessive feeling of failure.

Selection process at most of the universities involves consideration of many factors when making final decision about student: applicant's academic indicators and achievements, characteristics that make applicant unique, and priorities specifically set by particular colleges. This process is vague and non-transparent. Sometimes it may seem confusing which criteria affect the result of selection process the most. Students are not admitted based on only one factor; indeed, combination of all factors determine the chances of particular student to be successfully admitted. American colleges and universities evaluate many things to determine admission. Usually, most of the universities consider similar factors during reviewing tons of applications.

In this paper, detailed analysis of these factors is performed. The supervised machine learning regression algorithms are used in order to predict the probability of enrollment of the students to the graduate programs. Dataset containing the information about students who have already applied before was taken to develop a machine learning model and then accuracy of designed model were calculated. Today, most of the students have problems with application to the universities. Students often inadequately assess their potential and abilities, and, as a consequence, they end up applying to the low-ranked universities. The main aim of the project is to help students with the application process to the universities for which they have higher chance of admission. In general, graduate admission process can be considered as a mapping problem between students and universities, since both sides are looking for the most attractive options.

According to the UNESCO report, US is one of the most popular countries for international students to pursue their graduate studies [1]. It was reported by the State Department that the number of international students being accepted to US universities peaked at 1 million for the first time in 2015-2016. Moreover, applying to the most of these universities implies the high non-refundable application fees ranging from $50 to $125, and since each student usually applies to numerous universities at once, the total application cost reaches the border of hundreds of million USD per year. Due to intense flow of applications, work of admission committees becomes problematic, since they have to spend a lot of time reviewing all applications. By implementing our machine learning model, we can assist both sides by precisely evaluating each student’s profile and providing the chance of successful admission. Students will have an opportunity to considerably reduce the amount of money spent during the application process by applying to a smaller set of universities with higher chance for him/her to be accepted. Consequently, this will result in smaller number of applications which in its turn will remove some of the burden from the admission committee.

Innovative idea of the project is that stochastic gradient approach over a number of iterations was applied to get a better accuracy level than previously conducted projects. All those projects are shown in the reference list.

Quick Data Recap and Feature Scaling

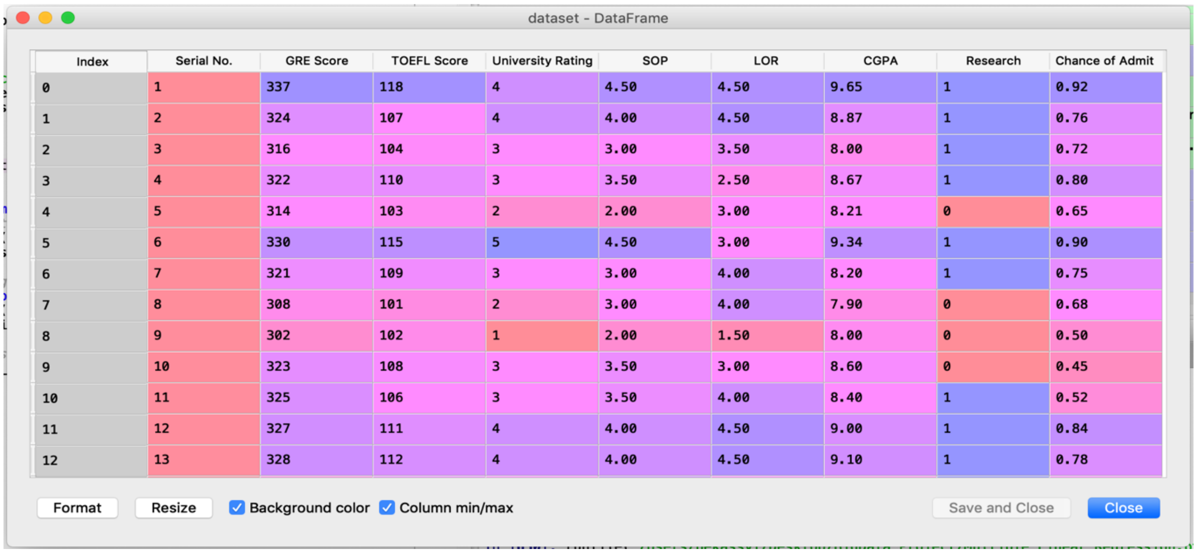
The dataset used to train machine learning models has seven features and a dependent variable that ranges between 0 and 1, showing the probability of getting admitted to the graduate program. In total, there are 500 samples.

Independent Variables: (See figure 1)

- **GRE score** is a numeric variable ranging from 0 to 340, only Integer values.
- **TOEFL score** is a numeric variable ranging from 0 to 120, only integer values
- **University Ranking** ranges from 1 to 4, so that the top universities are labeled as 4.
- **CGPA** is calculated out of 10. It can take rational values as well.
- **Research output** is a Boolean variable. It is either 0 or 1. Here, 1 stands for research experience.
- **Statement of Purpose and Letter of Recommendation quality** ranges from 0 to 5, and it can take rational values as well.

Dependent Variable: (See figure 1)

- **Chance of Admittance:** it ranges from 0 to 1.



dataset - DataFrame

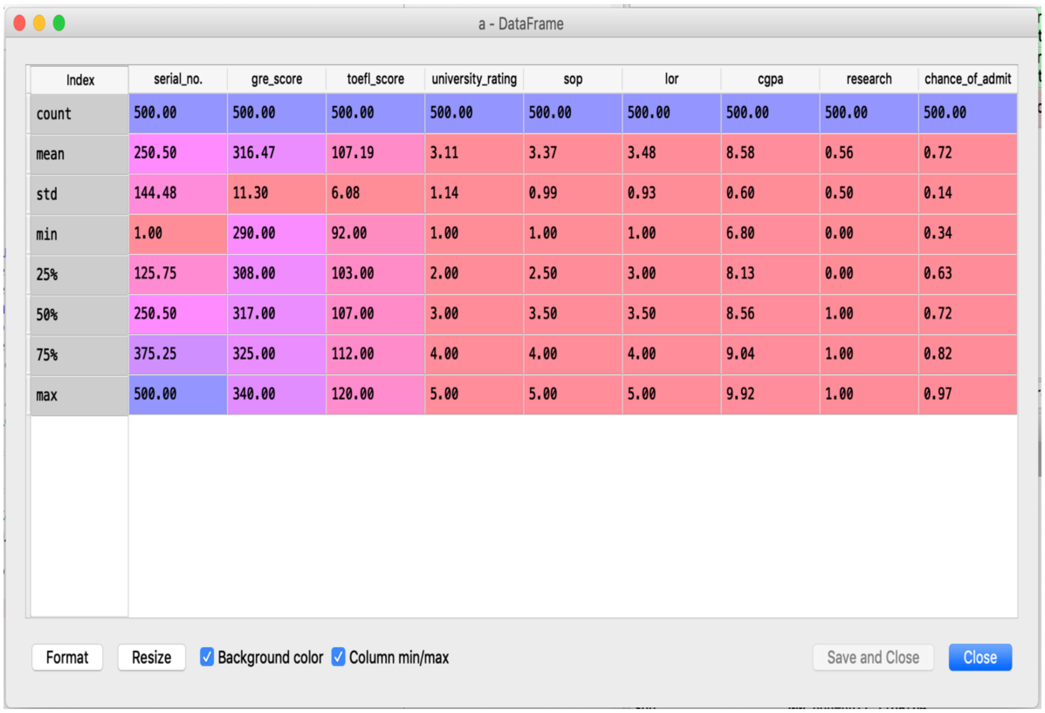
Index	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.50	4.50	9.65	1	0.92
1	2	324	107	4	4.00	4.50	8.87	1	0.76
2	3	316	104	3	3.00	3.50	8.00	1	0.72
3	4	322	110	3	3.50	2.50	8.67	1	0.80
4	5	314	103	2	2.00	3.00	8.21	0	0.65
5	6	330	115	5	4.50	3.00	9.34	1	0.90
6	7	321	109	3	3.00	4.00	8.20	1	0.75
7	8	308	101	2	3.00	4.00	7.90	0	0.68
8	9	302	102	1	2.00	1.50	8.00	0	0.50
9	10	323	108	3	3.50	3.00	8.60	0	0.45
10	11	325	106	3	3.50	4.00	8.40	1	0.52
11	12	327	111	4	4.00	4.50	9.00	1	0.84
12	13	328	112	4	4.00	4.50	9.10	1	0.78

☑ Background color ☑ Column min/max

Figure 1. Dataset

In order to get a clear overview of the dataset, the statistical summary was obtained. As it be seen from the figure 2, mean value of GRE feature is 316.47, whereas the mean of CGPA is 8.58. Basically, this implies that feature scaling has to be performed before training the models.

There are number of different feature scaling methods such as Standardization (also referred as Z-Score normalization), Min-max scaling, Binarizing, Normalizing and etc. Standardization scaling method will be used in this project, where the mean value is subtracted from each feature value then divided by the standard deviation.



Index	serial_no.	gre_score	toefl_score	university_rating	sop	lor	cgpa	research	chance_of_admit
count	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00
mean	250.50	316.47	107.19	3.11	3.37	3.48	8.58	0.56	0.72
std	144.48	11.30	6.08	1.14	0.99	0.93	0.60	0.50	0.14
min	1.00	290.00	92.00	1.00	1.00	1.00	6.80	0.00	0.34
25%	125.75	308.00	103.00	2.00	2.50	3.00	8.13	0.00	0.63
50%	250.50	317.00	107.00	3.00	3.50	3.50	8.56	1.00	0.72
75%	375.25	325.00	112.00	4.00	4.00	4.00	9.04	1.00	0.82
max	500.00	340.00	120.00	5.00	5.00	5.00	9.92	1.00	0.97

Figure 2. Statistical Summary

Next, it is important to analyze the correlation of each feature with the target value. As it can be seen from the figure 3, there is a positive correlation between CGPA, TOEFL result, IELTS result and the target output, meaning that both the target value and the feature moves in the same direction.

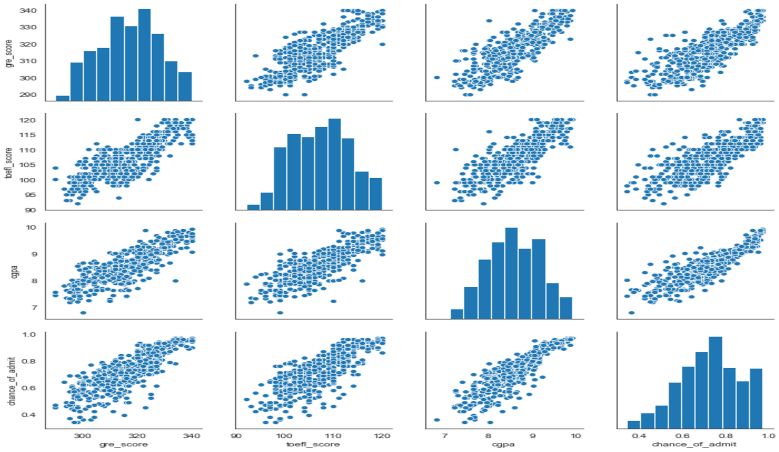
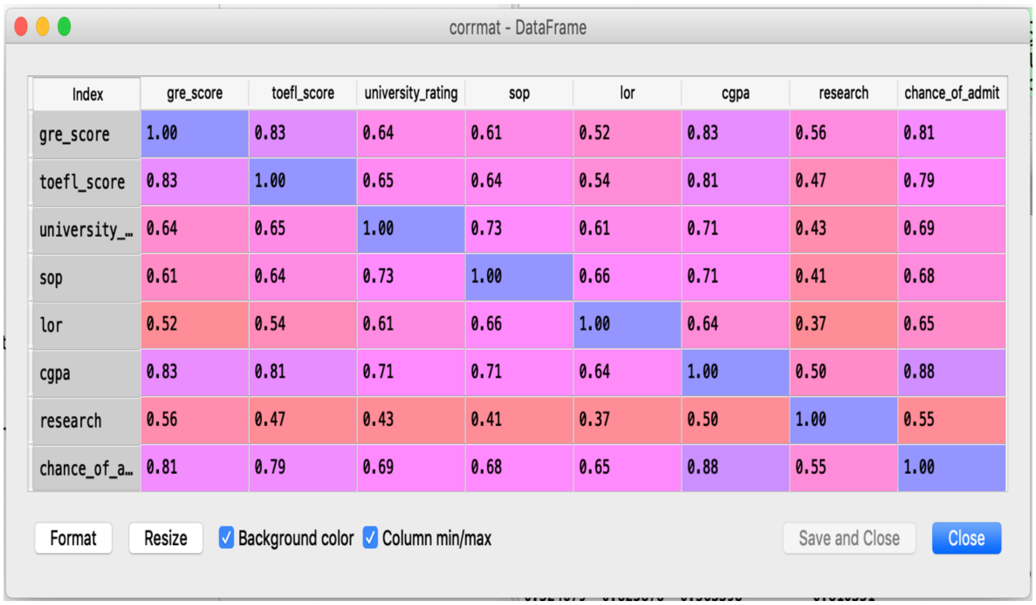


Figure 3. Visualization of Positive Correlation

Afterwards, correlation factors for each feature were obtained. As it can be seen from the figure 4, the highest correlation factors belong to CGPA, GRE result, TOEFL result. Basically, this implies that the impact of these features on the target value is higher than of others.



Index	gre_score	toefl_score	university_rating	sop	lor	cgpa	research	chance_of_admit
gre_score	1.00	0.83	0.64	0.61	0.52	0.83	0.56	0.81
toefl_score	0.83	1.00	0.65	0.64	0.54	0.81	0.47	0.79
university_…	0.64	0.65	1.00	0.73	0.61	0.71	0.43	0.69
sop	0.61	0.64	0.73	1.00	0.66	0.71	0.41	0.68
lor	0.52	0.54	0.61	0.66	1.00	0.64	0.37	0.65
cgpa	0.83	0.81	0.71	0.71	0.64	1.00	0.50	0.88
research	0.56	0.47	0.43	0.41	0.37	0.50	1.00	0.55
chance_of_a…	0.81	0.79	0.69	0.68	0.65	0.88	0.55	1.00

☑ Background color ☑ Column min/max

Figure 4. Correlation of Features

Regression Metrics used to evaluate the Models

In this project statistical method **Mean Squared Error (MSE)** was used to evaluate model by measuring the average of the squares of the errors which corresponds to the average squared difference between the estimated values and what is estimated. The resultant error must be always positive and not 0 due to randomness. [2] The formula of MSE is given by:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(a_i - \tilde{a}_\iota)^2$$

Root-Mean Squared Error (RMSE) is the most commonly used scale-dependent measure which is based on the squared errors. It can be compared with **Mean Absolute Error (MAE)**, since both metrics are widely used, despite the fact that MAE is easier to interpret and compute than RMSE. [2] RMSE can be evaluated as follows:

$$RMSE = \sqrt{mean(e_t^2)}$$

One of the regression metrics used in our project is the **Root Relative Squared Error (RRSE)** which is good analogue to RMSE. In fact, this metric is relative to what it would have been if a simple predictor had been used during the evaluation of the model. As a simple predictor, average of the actual values from training data is taken. The root elative squared error $E_i$of an individual model i can be calculated by the following equation:

$$RRSE = \sqrt{\frac{\sum_{i=1}^{N}(p_{(i)} - a_i)^2}{\sum_{i=1}^{N}(a_i - \bar{a})^2}}$$

Resulted value is given in percentage, where ideal case is given as value of 0.
Another evaluation metric **Root Absolute Error (RAE)** is very similar to RRSE with one distinction that it uses absolute values of error in evaluation. As in RRSE, results are expressed in percentage. Its ideal case is also represented by the value of 0. [2] The RAE formula is given below:

$$RAE = \frac{\sum_{i=1}^{N} |p_{(i)} - a_i|}{\sum_{i=1}^{N} |a_i - \bar{a}|}$$

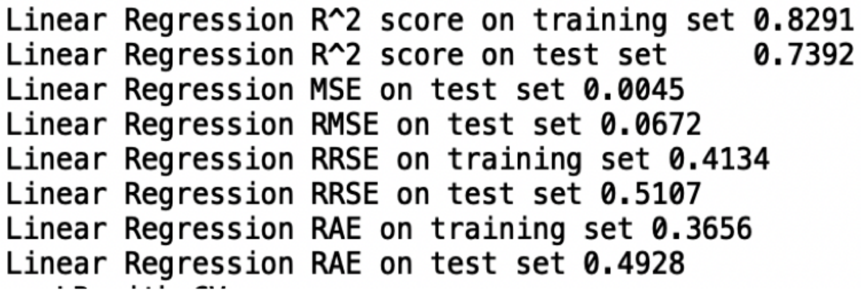
The last statistical measure used is **R-squared (R^2)** which represents the proportion of the dependent variable's variance that can be explained by an independent variables in a regression model. In other words, R-squared explains to what extent the variance of one variable explains the variance of the second variable. For instance, if the resulted value of R-squared is 0.75 for designed model, then the fourth of the observed variation can be explained by the applied inputs. [2] Calculation of R-squared can be performed as given below:

$$R^2 = 1 - \frac{Explained\ Variation}{Total\ Variation}$$

Constructed Models and their performance

1. ***<u>Linear Regression [3] with training and test set]:</u>***

In this case, we divided our dataset into training and test set. We gave 400 samples for training set and 100 sample for test set. Then, we created an object of LinearRegression() class. This object was fitted to the training set and was applied to the test set to predict the values for the test set. For Linear Regression Model with training and test set, we obtained following results:



```
Linear Regression R^2 score on training set 0.8291
Linear Regression R^2 score on test set     0.7392
Linear Regression MSE on test set 0.0045
Linear Regression RMSE on test set 0.0672
Linear Regression RRSE on training set 0.4134
Linear Regression RRSE on test set 0.5107
Linear Regression RAE on training set 0.3656
Linear Regression RAE on test set 0.4928
```

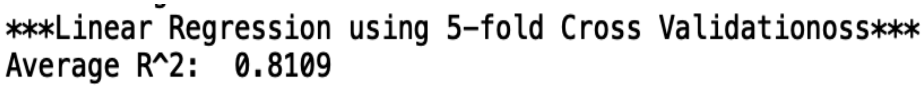
Discussion:

The accuracy of Linear Regression model with training and test set is 73.92 %, which is not a satisfactory result. In the next sections, we will try to improve this result, by Stochastic Gradient Descent, by adjusting the weights over 100000 iterations. As it was explained in the class, Holdout Estimator tends to be quite pessimistically biased.

RRSE and RAE is 51.07% and 49.28 % respectively. It is important to note that the value of RRSE and RAE has to decrease to indicate the improvement of the model.

2. ***Linear Regression [3] using 5-fold cross-validation:***

This time, we will report an error of linear regression by cross-validation error estimation technique. It is important to note that the model itself is the same as in the previous case. However, this time, error estimation techniques is different, meaning that obtaining less accuracy does not imply the improvement in the model. The accuracy in this case was as follows:



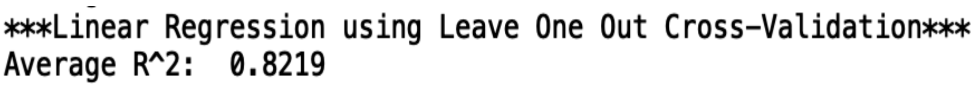
```
***Linear Regression using 5-fold Cross Validationoss***
Average R^2:  0.8109
```

Discussion:

As it can be seen, the accuracy is 81.09 %, which is quite more than in previous case. However, this does not imply model improvement, as it is simply an another way of reporting an error.

3. ***Linear Regression [3] using leave-one-out cross validation:***

The only difference of this error estimation technique is that here the size of each fold is simply one sample, meaning that each time one sample is set aside as a test set. In this case, we obtained following result:



```
***Linear Regression using Leave One Out Cross-Validation***
Average R^2:  0.8219
```

Discussion:

Here, the accuracy value is slightly more than in previous case: 82.19%. Similarly, it does not imply the improvement in the model. This is just another way of error estimation.

4. ***Ridge Regression [4]: 5-fold cross validation by looping***

This time, ridge regression model was constructed by creating an object of Ridge() class. By looping, we found out the optimal value of alpha for ridge regression (approximately 10^{-1}), as it can be seen from the figure 5.

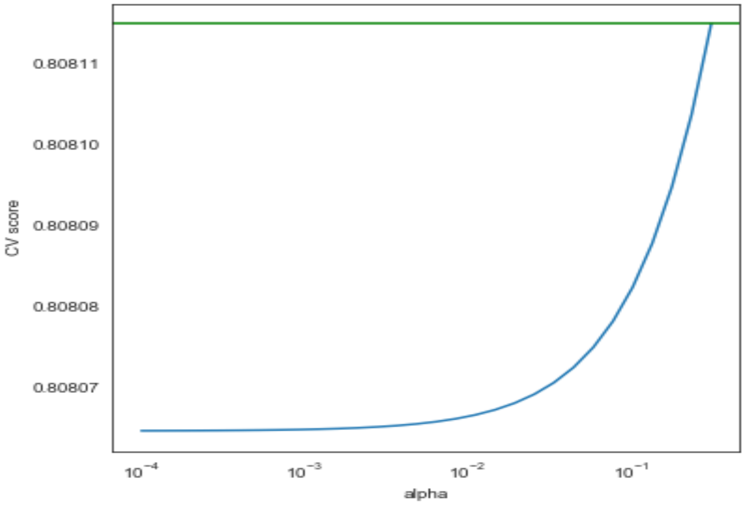


Figure 5. Alpha vs Accuracy for Ridge Regression

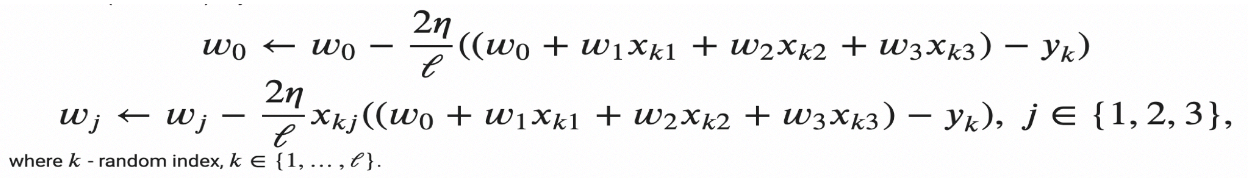
Discussion:

The accuracy of ridge regression model was around 80.85%, by the use of 5-fold cross validation technique. Linear Regression using 5-fold cross validation technique gave an accuracy of 81.09 %. This implies that Linear regression model outperforms Ridge regression model in this case.

5. ***Linear Regression using Stochastic Gradient Descent [5] approach***

Now, it is time to move to our approach to maximize the level of accuracy. All the models, that were previously constructed such as Linear Regression, Ridge Regression were already used by others to predict the chance of admit. We only implemented different error estimation techniques, to compare the accuracy levels.

We implemented stochastic gradient approach, where we will adjust our weights by minimizing the error of the model over each iteration. Here, we split our data into training and test sets. The main formula is given below:

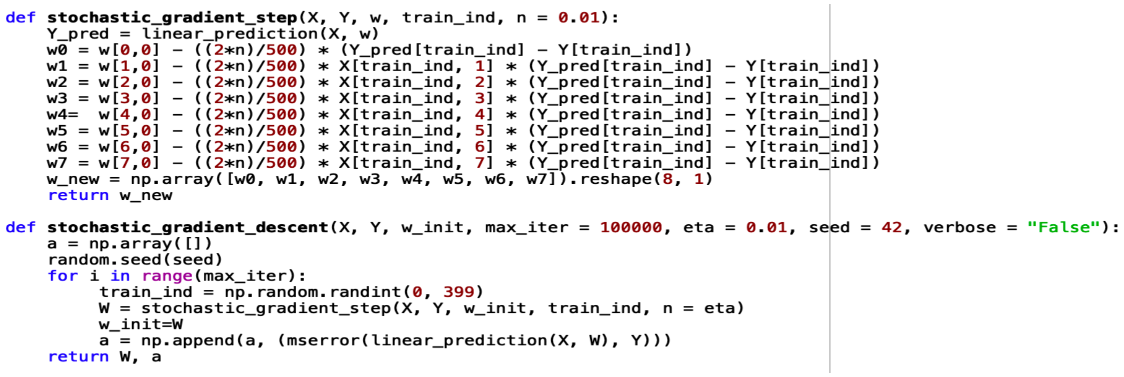


$$w_0 \leftarrow w_0 - \frac{2\eta}{\ell}((w_0 + w_1 x_{k1} + w_2 x_{k2} + w_3 x_{k3}) - y_k)$$

$$w_j \leftarrow w_j - \frac{2\eta}{\ell} x_{kj}((w_0 + w_1 x_{k1} + w_2 x_{k2} + w_3 x_{k3}) - y_k), \ j \in \{1, 2, 3\},$$

where k - random index, $k \in \{1, \dots, \ell\}$.

Python code:



```
def stochastic_gradient_step(X, Y, w, train_ind, n = 0.01):
    Y_pred = linear_prediction(X, w)
    w0 = w[0,0] - ((2*n)/500) * (Y_pred[train_ind] - Y[train_ind])
    w1 = w[1,0] - ((2*n)/500) * X[train_ind, 1] * (Y_pred[train_ind] - Y[train_ind])
    w2 = w[2,0] - ((2*n)/500) * X[train_ind, 2] * (Y_pred[train_ind] - Y[train_ind])
    w3 = w[3,0] - ((2*n)/500) * X[train_ind, 3] * (Y_pred[train_ind] - Y[train_ind])
    w4=  w[4,0] - ((2*n)/500) * X[train_ind, 4] * (Y_pred[train_ind] - Y[train_ind])
    w5 = w[5,0] - ((2*n)/500) * X[train_ind, 5] * (Y_pred[train_ind] - Y[train_ind])
    w6 = w[6,0] - ((2*n)/500) * X[train_ind, 6] * (Y_pred[train_ind] - Y[train_ind])
    w7 = w[7,0] - ((2*n)/500) * X[train_ind, 7] * (Y_pred[train_ind] - Y[train_ind])
    w_new = np.array([w0, w1, w2, w3, w4, w5, w6, w7]).reshape(8, 1)
    return w_new

def stochastic_gradient_descent(X, Y, w_init, max_iter = 100000, eta = 0.01, seed = 42, verbose = "False"):
    a = np.array([])
    random.seed(seed)
    for i in range(max_iter):
         train_ind = np.random.randint(0, 399)
         W = stochastic_gradient_step(X, Y, w_init, train_ind, n = eta)
         w_init=W
         a = np.append(a, (mserror(linear_prediction(X, W), Y)))
    return W, a
```

Result:

After 100000 iterations, we were able to get an accuracy of 83.63% on our test set, which is relatively more than previous two models.

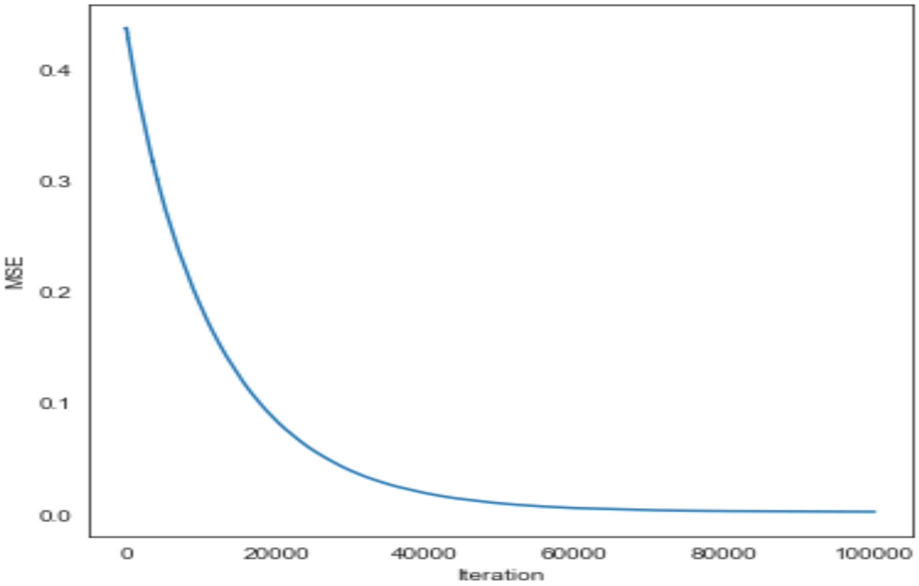


Figure 6. Number of Iterations vs MSE: Linear Regression Using Stochastic Gradient Descent

Conclusion

While implementing this project, we went through the materials which were told in the course of Big Data Analytics such as error estimation techniques, Linear

Regression, Ridge Regression, Regression Metrics to evaluate the performance of each model. The main innovative idea of this project was to implement a stochastic gradient approach [4], which could give a better accuracy result than previously implemented works such as Linear Regression and Ridge Regression. The reason why we added these models again is to try different error estimation techniques which were taught in the class and report their result.

References

[1] Global Flow of Tertiary-Level Students. http://uis.unesco.org/en/uis-student-flow

[2] Machine Learning: Concepts, Methodologies, Tools And Applications, 1st Edition. IGI Global. 2011

[3] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[4] Tong, L., 2004 "Solving large scale linear prediction problems using stochastic gradient descent algorithms", Banff, Alberta, Canada — July 04 - 08, 2004

Previous Works on the same Data:

1) https://towardsdatascience.com/predicting-ms-admission-afbad9c5c599
2) https://medium.com/@tkhan016/predicting-graduate-admissions-using-linear-regression-bab05b6988b5
3) https://github.com/sabeelahmad/Graduate-Admission-Predictor
4) https://rpubs.com/wkania/Graduate-linear

".

-