Data Mining project
Visualization, analysis and predictive modeling of a Graduate Admissions dataset

Grzegorz Rypeść

1. Introduction

The goal of the assignment is to perform basic statistical analysis on the graduation admission dataset. We are interested in exploring the association among features, applying and finding the best regression, classifications and clustering methods that will predict/classify values. We hope that by having created this analysis we can help students to estimate their chances when applying for master degrees.

The dataset we chose comes from: https://www.kaggle.com/mohansacharya/graduate-admissions
It contains several parameters which are considered important during the application for Masters Programs. It is is inspired by the UCLA Graduate Dataset. It estimates a chance for a student for admission to master studies. We mainly used Python programming language (numPy, pandas, matplotlib, seaborn libraries), GeoDa and Power-BI visualization tools to make the research.
You can find this projects' repository with all code sources and raports here: https://github.com/grypesc/graduateAdmissions

In this raport we firstly focused on data analysis, preparation and visualization. Then we applied machine and statistical learning methods to build regression, clustering, classification models and provided theirs comparison .

2. Dataset preparation

The parameters included in the dataset are :
1. Serial No. (1 to 400), int 64
2. GRE Scores (290 to 340), int64
3. TOEFL Scores (92 to 120), int64
4. University Rating (1 to 5, higher is better), int64
5. Statement of Purpose (1 to 5), int64
6. Letter of Recommendation Strength (1 to 5), int64
7. Undergraduate GPA (6.8 to 9.92), float64
8. Research Experience (either 0 or 1), float64
9. Chance of Admit (0.34 to 0.97), float64

There are no invalid nor null records. There are 500 records in total and 9 columns. We changed some of the labels to remove spaces at the end of the text as it may be misleading. The dataset weighs 16,2 kB.

3. Correlation between features

With use of Python programming language we managed to get a correlation table visible on the figure number 1. We can see that LOR, SOP, and Research have the lowest correlation with probability of admission. On the contrary, the highest influence on the admission have: CGPA, GRE score and TOEFL score. Serial No. is, as expected, extremely low and that table will be further not taken into account during the analysis.

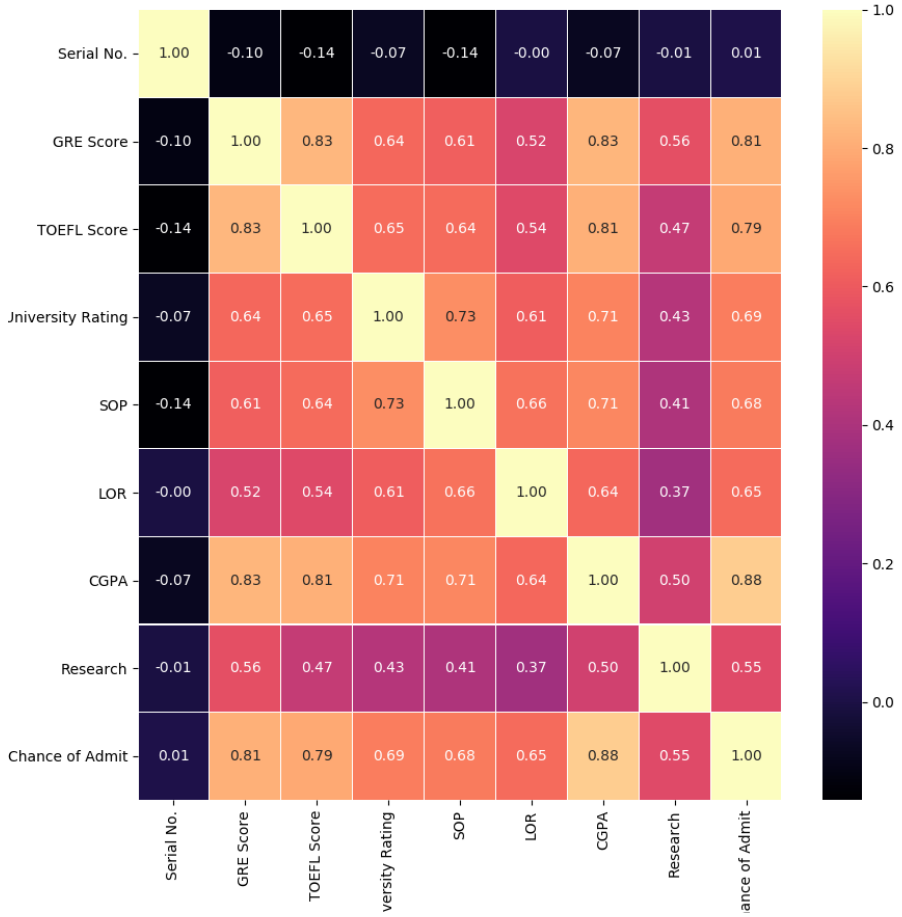

Fig. 1 Correlation between features

4. Visualization of the dataset

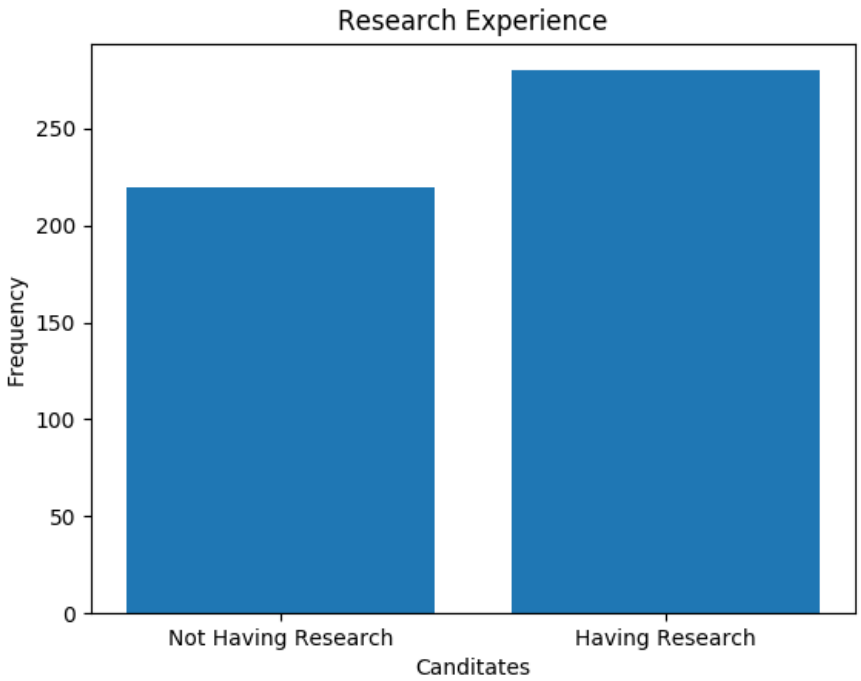

Fig. 2 Frequency of candidates by research

```
grzegorz@ubuntu:~/Projects/graduateAdmissionsAnalysis$ python visualization.py
('Not Having Research:', 220)
('Having Research:', 280)
```

Fig. 3 Number of candidates by research

Although research has the lowest correlation out
of all of the parameters we don’t think it can be skipped. 56% of candidates have a research.

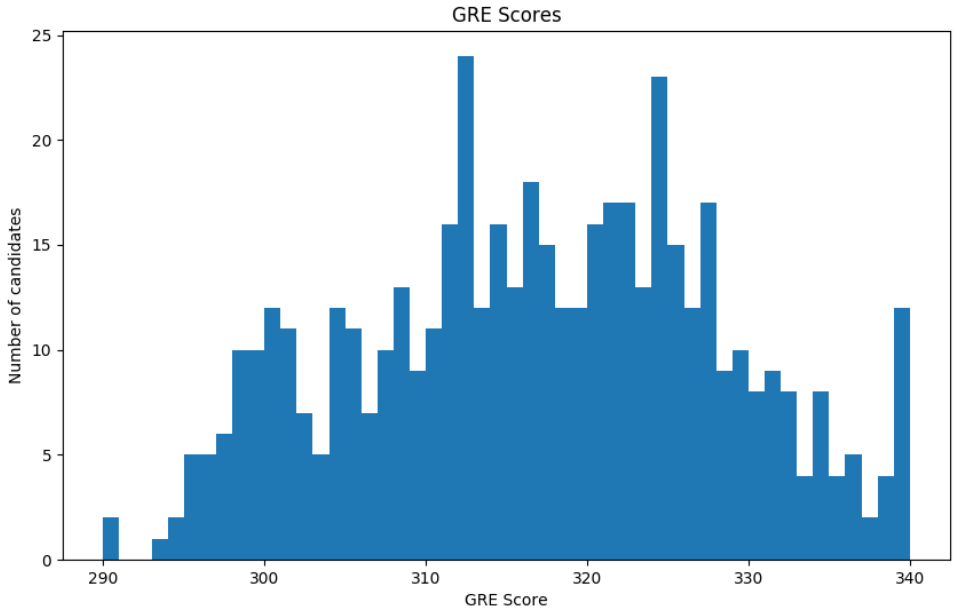

Fig. 4 No. candidates by GRE Score

Histogram from figure 4. shows the frequency for GRE scores. There is a density between 310 and 330. Surely having more than 330 points is a good aiming spot for a candidate.

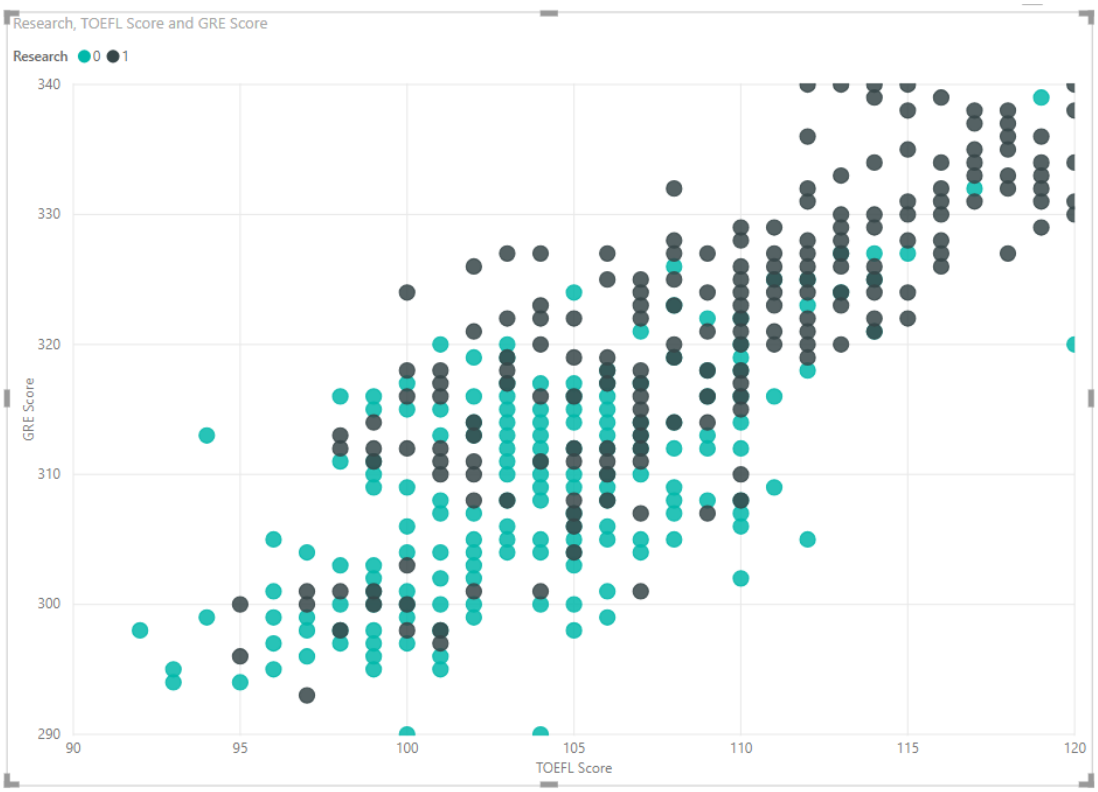

Fig. 5 GRE score by TOEFL score by having a research

Figure 5. shows a strong correlation between TOEFL score and GRE score. It is logical that on average a good student succeeded in both. We can also see that people with higher scores made a research.

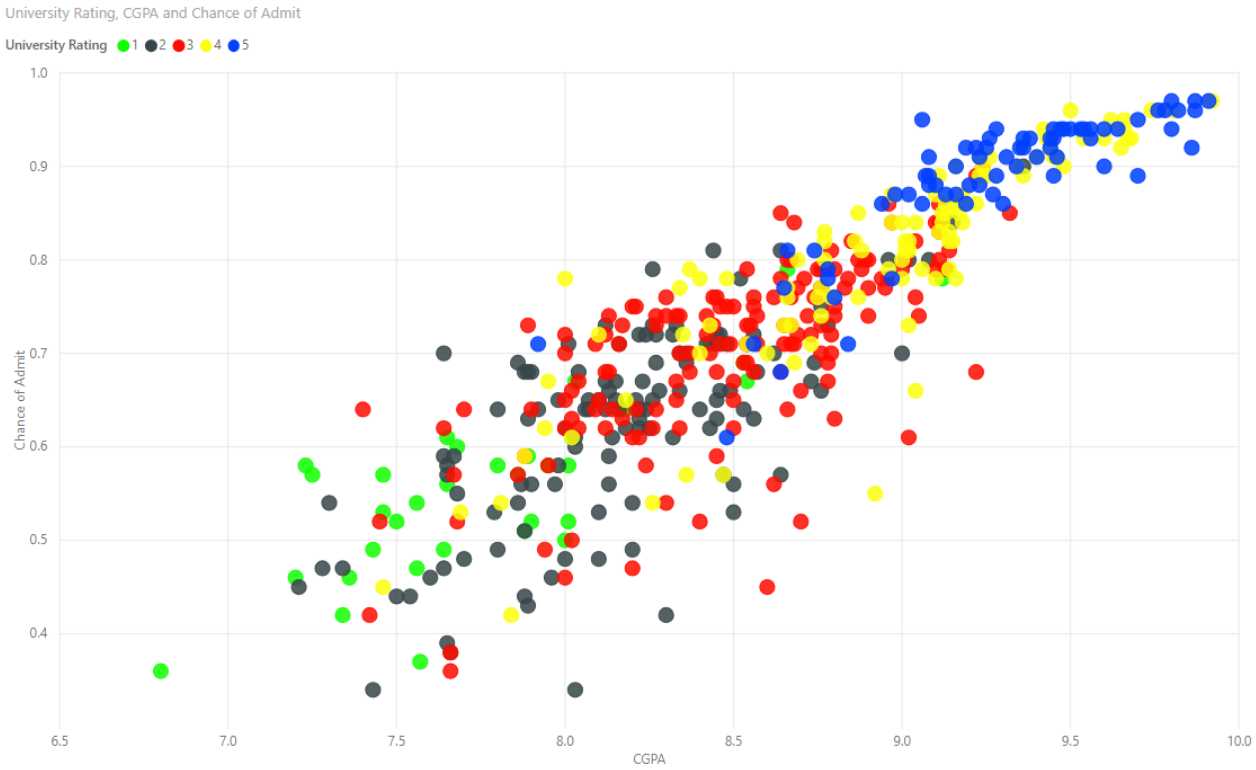

Fig. 6 Chance of admission by CGPA by university rating

Chart “Fig.6” shows that the better university is, its candidate gets more points and has a higher chance of admission.

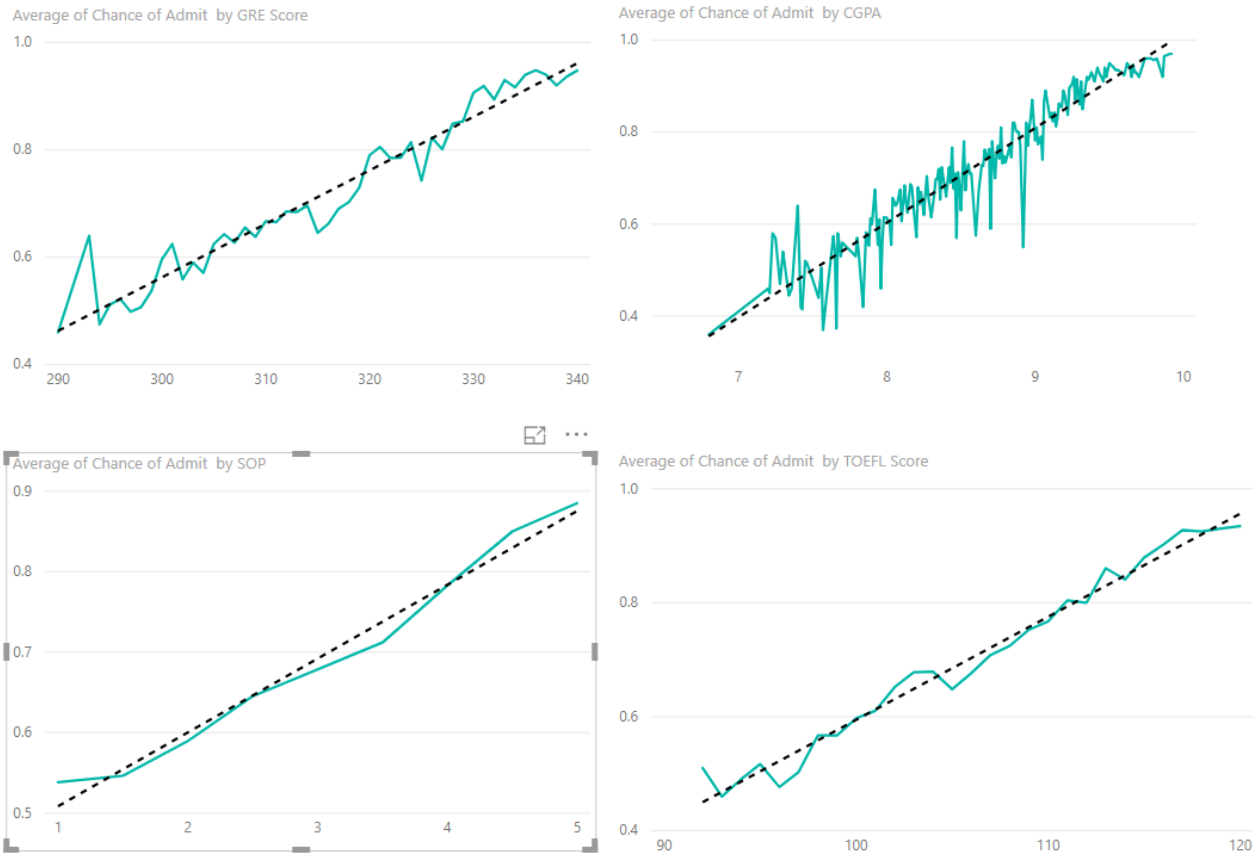

Fig. 7 Admission chance by different features and a trend line

Estimated regression lines indicate that the higher GRE, CGPA, SOP, TOEFL the higher admisson chance. We can see big variation on the CGPA/admisson chart. Linear trends are visible.

On the Fig.8 chart, modeled in program GeoDa we also observe a linear trend in admission chance.

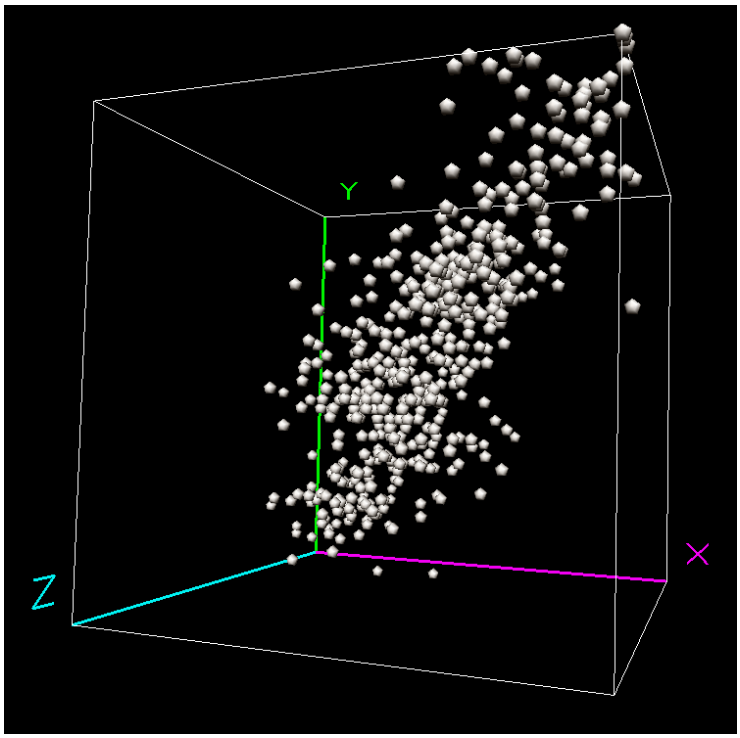

Fig. 8 3D visualization, X – TOEFL score, Y –GRE score, Z – admission chance

5. Applying regression algorithms

For a regression prediction we decided to build 7 models. We are going to compare results of each other. Regressors we decided to use come from scikit-learn library. These are:

- Linear Regression
- Ridge
- Random Forest Regressor
- Bayesian Ridge
- Linear Support Vector Regression
- Lasso
- Stochastic Gradient Descent

For accuracy comparison test we used a k-fold Cross-Validation to get more reliable results. Therefore for every model the test is firstly split into 5 subsets. Then regressor is fit into 4 of them and tested on the last one. Accuracy is calculated. This is repeated 5 times so every subset once becomes a test set. At the end we calculate an average accuracy from all 5 tries. It is also worth to mention that we applied a feature scaling (from 0 to 1) on the dataset before we ran this test. Here are the results:

```
grzegorz@ubuntu: ~/Projects/graduateAdmissions
File  Edit  View  Search  Terminal  Help
grzegorz@ubuntu:~/Projects/graduateAdmissions$ python -W ignore regression.py
### LinearRegression ###
Average accuracy on training sets: 0.8081460596390899

### Ridge ###
Average accuracy on training sets: 0.8084115596635855

### RandomForestRegressor ###
Average accuracy on training sets: 0.7337054492888929

### DecisionTreeRegressor ###
Average accuracy on training sets: 0.5874050576906417

### BayesianRidge ###
Average accuracy on training sets: 0.8083393587807187

### LinearSVR ###
Average accuracy on training sets: 0.8068942212187394

### Lasso ###
Average accuracy on training sets: 0.8082042247192074

### SGDRegressor ###
Average accuracy on training sets: 0.7440146900336941

grzegorz@ubuntu:~/Projects/graduateAdmissions$
```

Fig. 9 Regressors' accuracy with scaling

```
grzegorz@ubuntu: ~/Projects/graduateAdmissions
File  Edit  View  Search  Terminal  Help
grzegorz@ubuntu:~/Projects/graduateAdmissions$ python -W ignore regression.py
### LinearRegression ###
Average accuracy on training sets: 0.8081460596390901

### Ridge ###
Average accuracy on training sets: 0.8082401616972963

### RandomForestRegressor ###
Average accuracy on training sets: 0.7437942135483457

### DecisionTreeRegressor ###
Average accuracy on training sets: 0.6001677864713084

### BayesianRidge ###
Average accuracy on training sets: 0.8082181975059688

### LinearSVR ###
Average accuracy on training sets: 0.6354799688153534

### Lasso ###
Average accuracy on training sets: 0.8081894684217049

### SGDRegressor ###
Average accuracy on training sets: -3.376059556681034e+29

grzegorz@ubuntu:~/Projects/graduateAdmissions$
```

Fig. 10 Regressors' accuracy without scaling

As we see on the fig. 9 and fig. 10 Ridge and Bayesian Ridge regressors got the best accuracy. Only decision tree regressor got noticeably worse accuracy. We couldn't find better parameters (such as seed used by the random number generator or splitting strategy) that would increase the prediction accuracy. It is probably overfitted because its accuracy on the training set is equal to 100%. We also noticed that Linear Support Vector Regression performed worse without features scaling and Stochastic Gradient Descent became completely unreliable in that case.

To ensure that our choice of k to k-fold Cross-Validation was possibly the best, we calculated accuracy for k in range 2 to 100. The results can be seen on plots below:

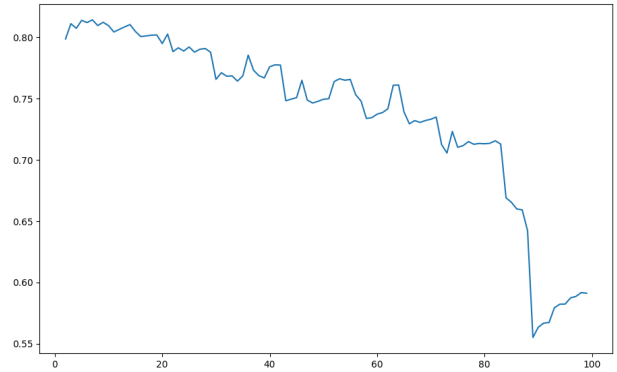

Fig. 10 Linear Regression - k by accuracy

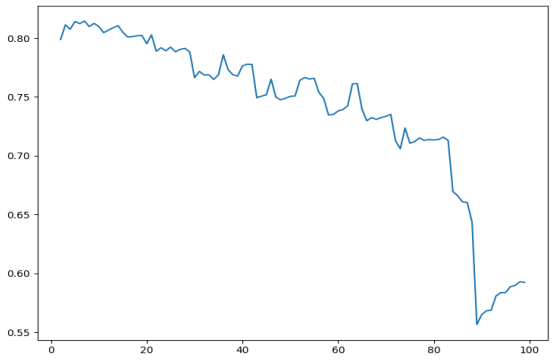

Fig. 11 Bayesian Ridge - k by accuracy

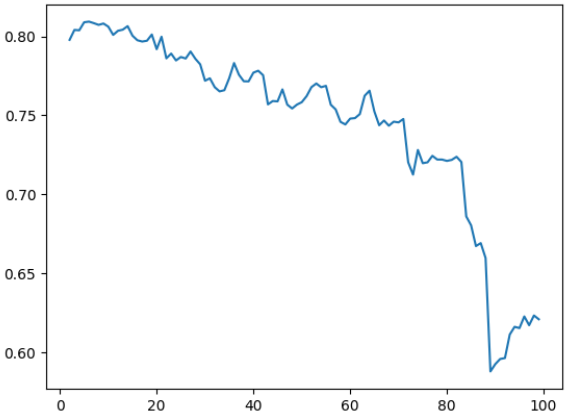

Fig 11. Linear SVR – k by accuracy

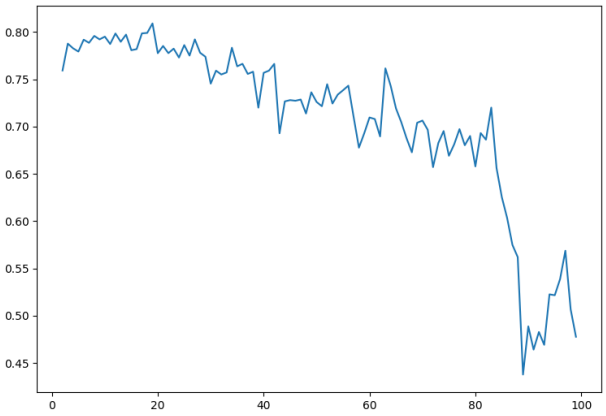

Fig 12. Random Forest – k by accuracy

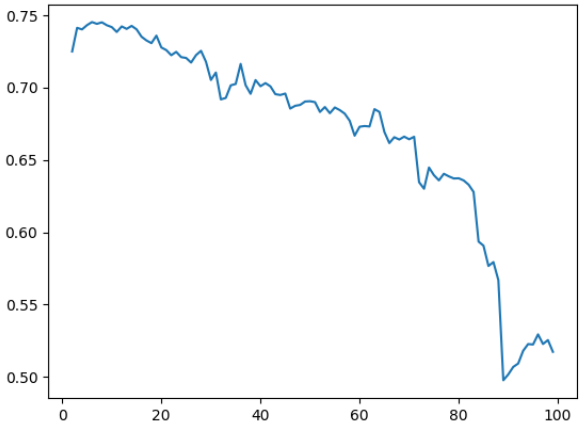

Fig 13. SGD Regressor – k by accuracy

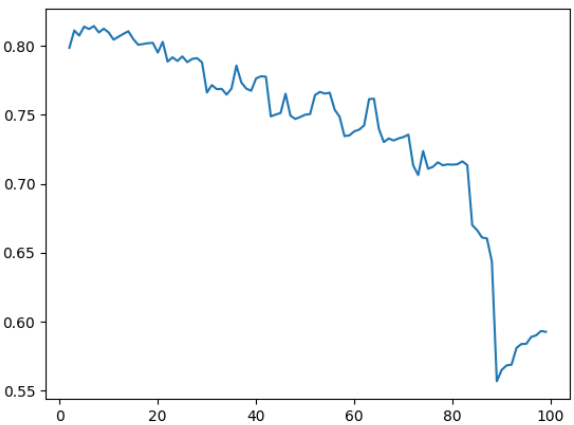

Fig 14. Lasso – k by accuracy

As we can see, the best accuracy is usually obtained for k = 5. It grows from 2 to 5 and than falls as k gets higher than 10.

Another part which we wanted to visualize was how our predicted probability differs from the real data. To do that, we drew plots for each regression model, putting predicted and real probabilities on axis. Below we can observe obtained results:

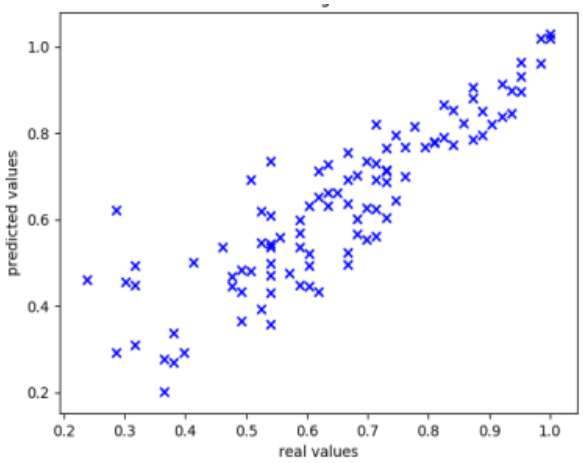

Fig 15. Linear Regression

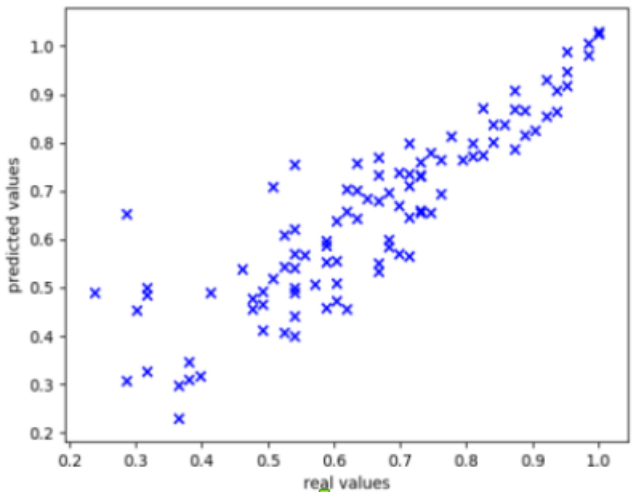

Fig 16. Linear SVR

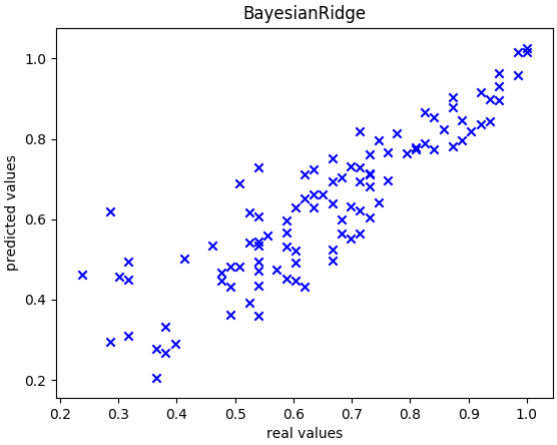

Fig 17. Bayesian Ridge

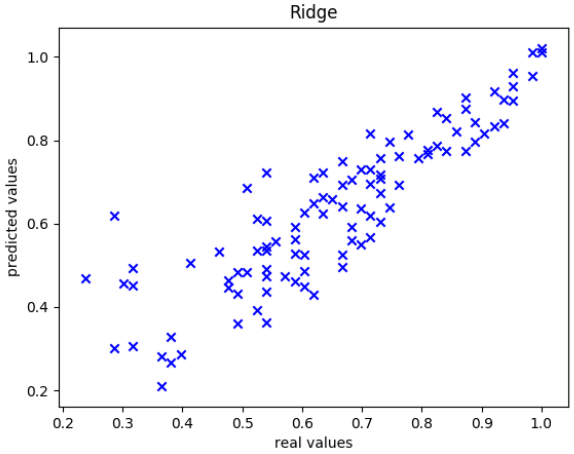

Fig 18. Ridge

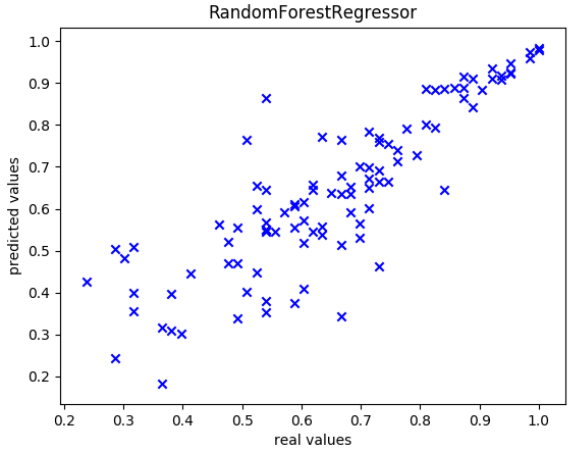

Fig 19. Random Forest

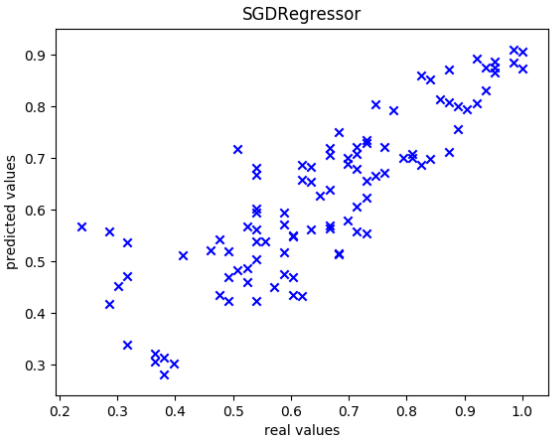

Fig 20. SGD Regressor

In a perfect situation we would like to get a linear function. Knowing it is impossible, its approximation is also satisfying for us. As we can notice from graphs, Bayesian Ridge and Ridge got the best approximation to a straight line. Linear Regression, Linear SVR and Lasso looks quite similar, but after a moment of consideration we can notice they are little bit worse from its predecessors. The two worst one even for the first look are SGD Regressor and Random Forest Regressor. The results agrees with our previous analysis.

6. Applying clustering algorithms

For a clustering prediction we decided to choose three different methods: K means, Gausian Mixture and Hierarchical Clustering. For each one we calculated accuracy by comparing our predictions with real values from the dataset.

As a predicting feature we picked “Researches”. We dropped this table from our data and tried to predict it once again. We decided that predicting this feature will make the biggest sense for our dataset, because it has only 2 possible results – 1 or 0.

To get better results we decided to use Principal Component Analysis. It maps all features to just one, making our calculations faster.

```
*** Errors ***

Average clustering accuracy for K means: 0.780000

Average clustering accuracy for Gausian Mixture: 0.782000

Average clustering accuracy for Hierarchical Clustering: 0.766000
```

Fig 21. Accuracy while using PCA

```
*** Errors ***

Average clustering accuracy for K means: 0.780000

Average clustering accuracy for Gausian Mixture: 0.780000

Average clustering accuracy for Hierarchical Clustering: 0.788000
```

Fig 22. Accuracy without PCA

Results while using PCA and without it were similar, it was just a little bit worse for Hierarchical Clustering. When we clustered without PCA, we obtained the best result for Hierarchical Clustering, for K means and Gausian Mixture it was the same.

We also plot scatters of how classification should looks like (on the picture with title “Research) and how it looked after our predictions. The results are shown below.

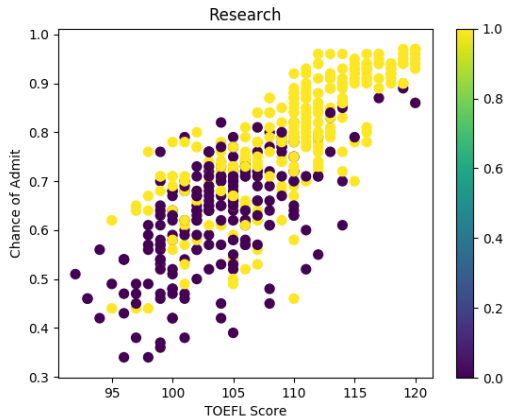

Fig 23. Original values for research

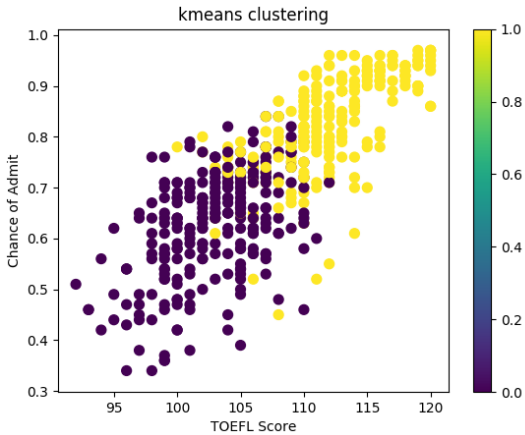

Fig 24. Research after K means clustering

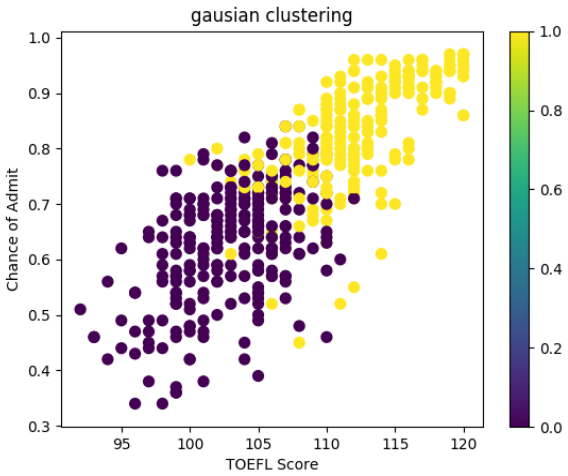

Fig 25. Values of research after gausian clustering

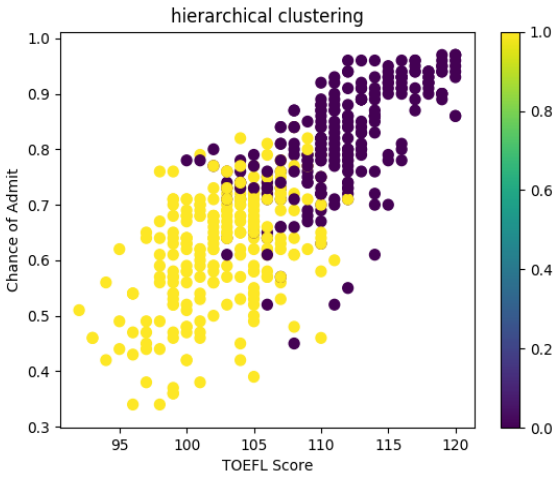

Fig 26. Values of research after hierarchical clustering

Note: Colors for hierarchical clustering are reversed, because clustering methods chooses labels of the classes randomly from 0 and 1.

At the end we wanted to find out which features are correlated the most with “Researches” and than check what happens if we make clustering taking into account only the most important features.

From the picture below we can find out that two most important features for our clustering model are: GRE Score, Chance of Admit and CGPA.

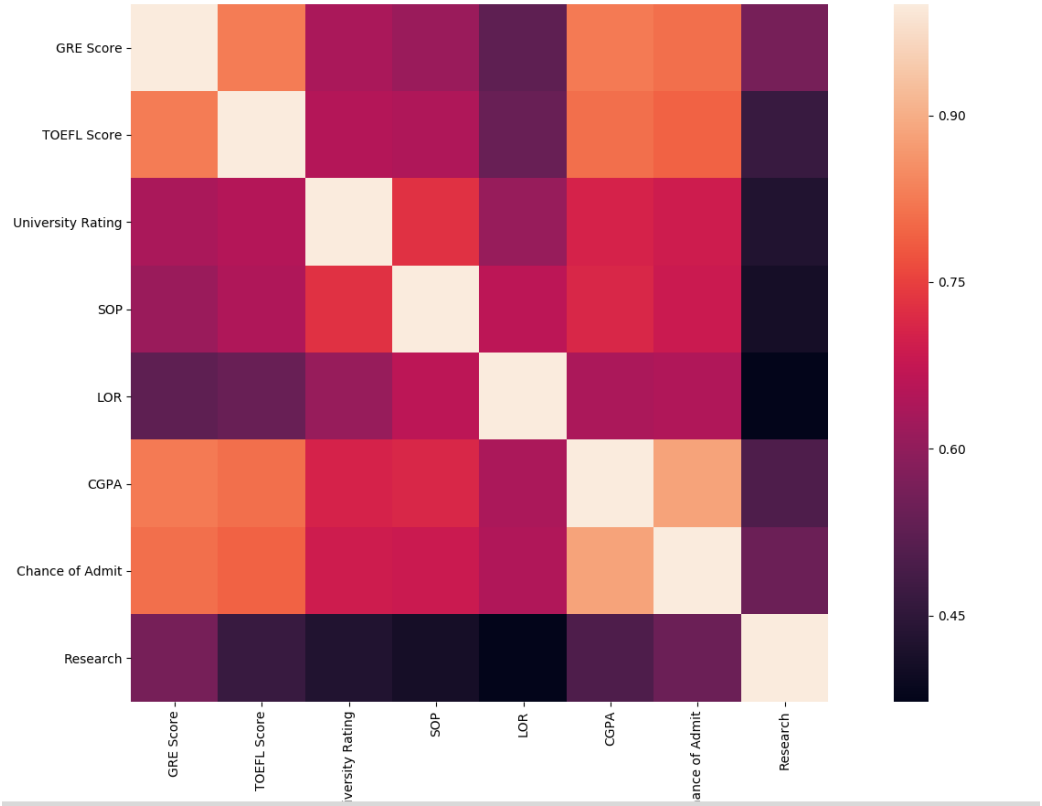

Fig 27. Corelation between features

After making clustering for GRE Score, Chance of Admit and CGPA and than only for GRE Score and Chance of Admit we got those results:

```
*** Errors ***

Average clustering accuracy for K means: 0.772000

Average clustering accuracy for Gausian Mixture: 0.772000

Average clustering accuracy for Hierarchical Clustering: 0.732000
```

Fig 28. GRE Score, Chance of Admit, CGPA

```
*** Errors ***

Average clustering accuracy for K means: 0.780000

Average clustering accuracy for Gausian Mixture: 0.780000

Average clustering accuracy for Hierarchical Clustering: 0.782000
```

Fig 29. GRE Score, Chance of Admit

While clustering with 3 features we got slightly worse results than before. However, interesting fact is that for clustering with 2 features we got better results than for clustering with 3 features. They were almost similar to results with all of the factors.

7. Applying classification

To consider classification we needed to map one of the features into classes. We chose to turn chance of admission into 4 classes as follows:

- 0.90 – 1.00 Class 0
- 0.75 – 0.89 Class 1
- 0.50 – 0.74 Class 2
- 0.00 – 0.49 Class 3

We fit classifiers to random 400 rows and tested them on other 100. Normalization of features was applied. We used following classification algorithms:

- Logistic Regression
- Support Vector Machines,
- Multi-layer Perceptron (4 hidden layers, each one has 20 nodes)
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Decision Tree Classifier
- Gaussian Naive Bayes
- K-Neighbors Classifier

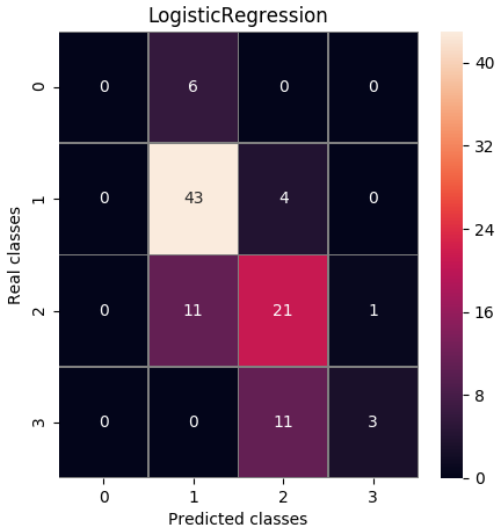

Fig. 30

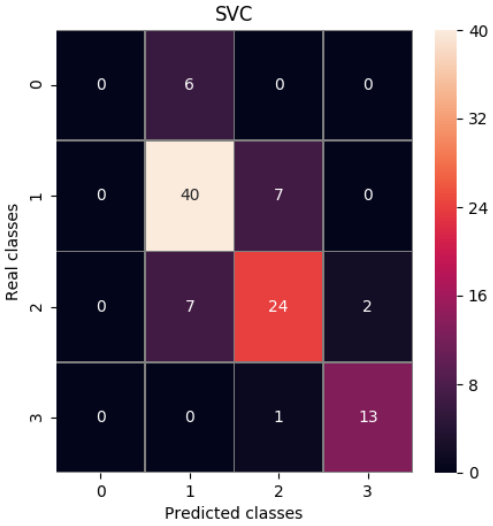

Fig. 31

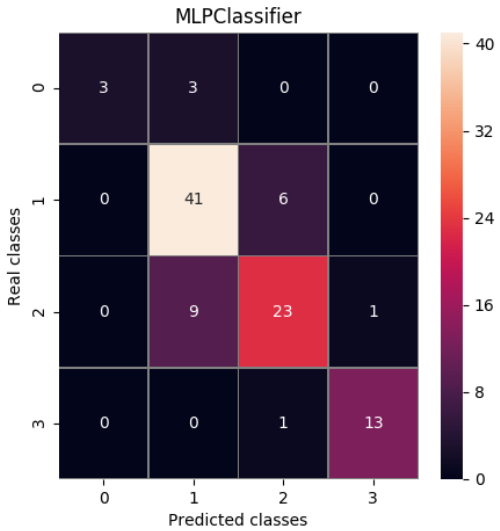

Fig. 32

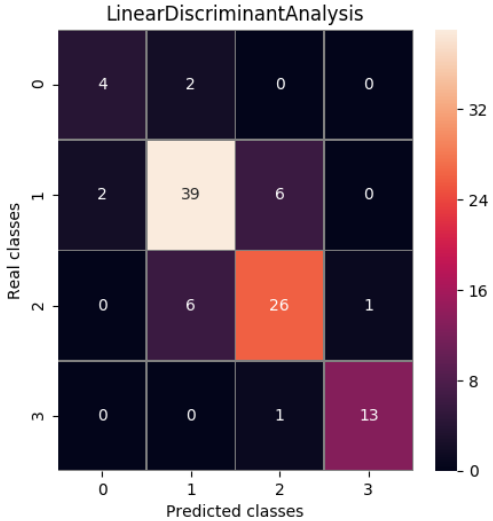

Fig. 33

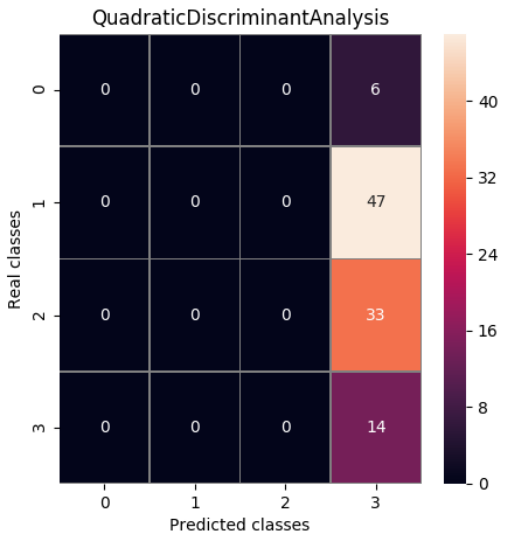

Fig. 34

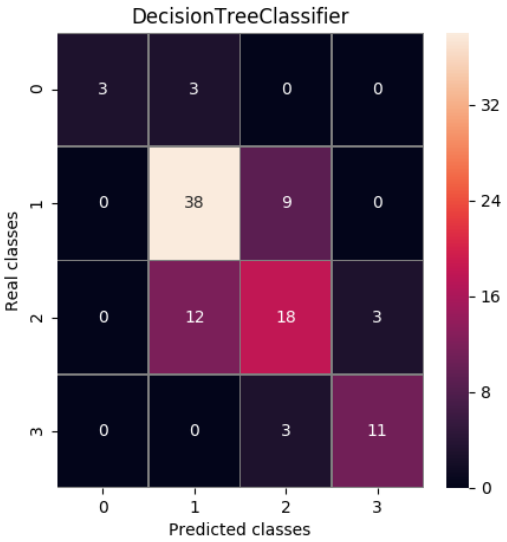

Fig. 35

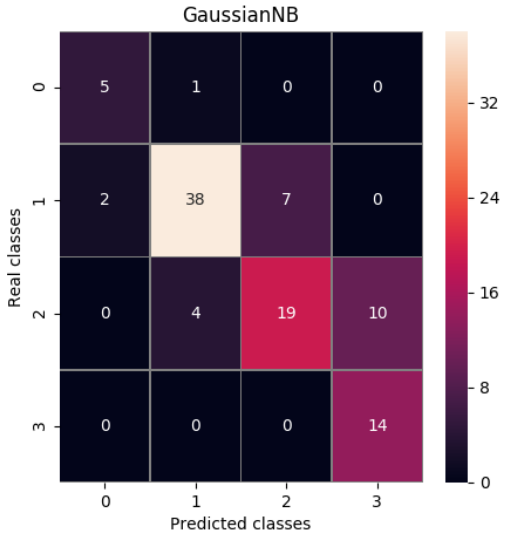

Fig. 36

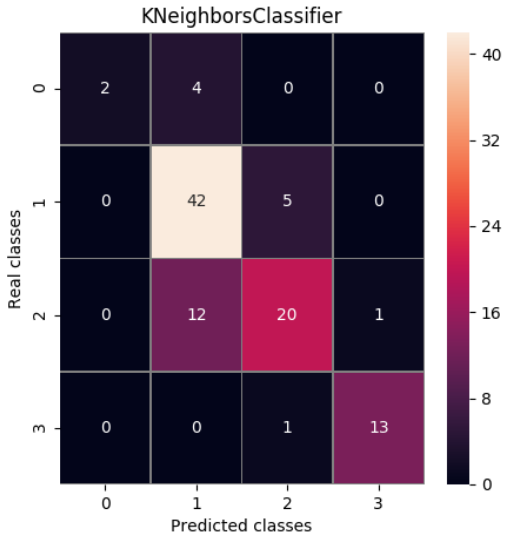

Fig. 37

Logistic regression, SVC and QDA classifiers classified 0 data points to class 0. For reasons we don't know QDA classified all points to class number 4. Maybe that's because the classes are linearly correlated to features. For this reason LDA classifier got great accuracy of 82% being also the best one. Logistic regression bad accuracy can be explained by a not binary number of class, it performs the best when it is fit to classify only 2 classes.

```
### LogisticRegression ###
Classification accuracy: 0.67

### SVC ###
Classification accuracy: 0.77

### MLPClassifier ###
Classification accuracy: 0.8

### LinearDiscriminantAnalysis ###
Classification accuracy: 0.82

### QuadraticDiscriminantAnalysis ###
Classification accuracy: 0.14

### DecisionTreeClassifier ###
Classification accuracy: 0.7

### GaussianNB ###
Classification accuracy: 0.76

### KNeighborsClassifier ###
Classification accuracy: 0.77
```

Fig 38 Accuracy of classifiers

8. Key findings

After analyzing the data set we could see that, with exception of index number, all features are correlated with each other. That means that, on average, the better the student is the better are all his grades, test results, university and chance of admission. Chance of admission is linearly proportional to non binary features.

The best regression models were ones based on Ridge and Bayesian Ridge algorithms. If a student would like to estimate his chances of admission he can do it with an accuracy of at least 80.3 %. At least because the final regressor would be trained on all 500 data points and it would perform better. The best k values for k-fold cross-validation ranged from 5 to 10.

Clustering analysis proved that researches do really affect chances of admission. It didn't show us differences in used clustering algorithms nor when we used Principal component analysis.

Classification analysis showed that for predicting classification of graduation admissions in the dataset it is the best to use linear discriminant analysis or a neural network. When it comes to classification problems with more than 2 classes accuracy of logistic regression is low compared to other classifiers.

If you have any other conclusions feel free to open an issue on the github repository. We are curious about what happened for example with QDA in the classification test.

9. References

- scikit-learn documentation - https://scikit-learn.org/stable/
- https://www.kaggle.com/mohansacharya/graduate-admissions
- An Introduction to Statistical Learning 7th edition book
- http://uc-r.github.io/discriminant_analysis