COMP 5411 FA / FB

Fall 2019

Final Project Report

Project Title: Student Mobility Pattern in the EU

Group Members:

Name	Student No.	Section
Anna Dieckvoss	1127324	FB
Borys Komarov	0889421	FB

Student Mobility Pattern in the EU

Anna Dieckvoss (1227324)
Department of Computer Science
Lakehead University
adieckvo@lakeheadu.ca

Borys Komarov (0889421)
Department of Computer Science
Lakehead University
bkomarov@lakeheadu.ca

ABSTRACT

Background: The analysis of human mobility patterns helps to predict the destination choices. Prediction of the most suitable destinations countries improves the overall experience for students in the Erasmus program.
Objective: This study aimed to use Descriptive analytic and machine learning methods to (1) predict suitable destinations countries for future students and (2) measure relative performance of Universities to help the Erasmus coordinator to make informed finical decisions.
Methods: The destination choices are defined as the 33 countries, which are participating as host countries in the Erasmus program. Prediction models were developed using 2 methods: (1) Naive Bayes (2) Random Forests. It was applied on the imbalanced an balanced dataset. A class imbalance issue was addressed using a random over and under sampling of the training dataset.
The relative performance of Universities defined by comparing different success steps and grouping with the method k-medoids.
Results: The prediction model which gave us the best average result was Naive Bayes. In this class-imbalanced dataset for Naive Bayes top-3 accuracy was 74.95% and random Forest top-3 accuracy: 66.64%. In this balanced dataset, for Naive Bayes top-3 accuracy: 68.547705% and random Forest top-3 accuracy: 67.39%. The Clustering analysis shows that there is clear division on bigger and smaller in terms of participating universities in the ERASMUS program.
Conclusions: Classification analysis was applied to predict the top 3 destination for students in the EU. Clustering analysis was applied to group Universities based on their relative performance.

Keywords: Erasmus Dataset, classification analysis, cluster analysis, Student Mobility Pattern

I. INTRODUCTION

A. Problem definition

The topic of this paper is the Human Mobility Pattern Clustering. Based on an Erasmus data sets we analyzed the student mobility trends in 2012-2013.

Erasmus exchange program operates since the late 1980 's and have sent 3.000.000 students to another country. It stands for EuRopean Community Action Scheme for the Mobility of University Students. With Erasmus students get the opportunity to study at universities in the EU member states for set periods of time. The generated credit points can be transferred to the home university [1]. The Bologna reform defines a standardization of study in the European Union. The Erasmus program is also extended for staff working in education non-necessarily on teaching positions. Participants of the project describe that it was very hard to organize the process although there is a general standardization and it was definitely worth going for it and changed their life for good.

The mobility of young people is interesting to know because based on analyzing human mobility patterns we can predict destination choices for the following year and improve the overall experience for students on the Erasmus program by providing them with a recommendation on the most suitable destinations countries. More and more people want to try gaining knowledge abroad every year in our study we try to help students enrolled in the Erasmus program to make more informed decisions.

B. Objectives

This study aimed to use Descriptive analytic and machine learning methods to predict suitable destinations countries for future students by using classification. Additionally, the relative performance of Universities was measured by comparing different success steps to help the Erasmus coordinator to make informed finical decisions based on the clustered university.

C. Significance of the problem

The number of students who are going abroad every year is highly increasing. The broad social, economic and political factors of this mobility are diverse and not always known. It was very nicely put in [5] “Experiences and meanings gained in a mobility program like Erasmus have implications beyond academic achievements”. Globalization impacts the world and more and more companies become worldwide having offices in different locations with different cultures and one should be prepared for it after graduation. Academic mobility is an opportunity for students to gain new knowledge, experience different cultures, and build new connections.

The more people go on the Erasmus and similar programs the more data is generated and it now becomes even more important to provide accurate and automated recommendations. The recommendation will be based on various standard features including destination country, origin country, study program, program length, credit points. Potentially, our findings could help hundreds of thousands of exchange or potential-exchange students in Europe. Many parties can benefit from our study. Countries and universities might get a detailed analysis of how they are performing in terms of international student recruitment and what would be predictions for them for the nearest future. Students can get a chance to maximize their experience by adopting the power of data on their side. Erasmus program may get deeper insights on which countries are better encouraging incoming traffic of international students and what affects that.

D. Literature review

We were able to find several articles exploring various aspects of the Erasmus program and attempting to produce insights. The paper [3] analyze the mobility network of Erasmus staff and students within a big European project FETCH (Future Education and Training in Computing). The goal was to evaluate the current state of the Erasmus mobility network among institutions. The authors applied Techniques that are commonly used in social network analysis to academic mobility flow. "The structure of the network is investigated using connected component analysis and k-core decomposition."

At the same time authors in another paper [4], which uses the same data as we are using for our study, but for a different time period. They were trying to cluster countries based on the number of students coming into and out of the country and then make a conclusion if there is a finite number of classes as well as whether every country falls into this set of identified classes. In their results, they present that it is possible to represent all countries using three classes: good importers and exporters, good importers only and good exporters only. Their results are quite interesting, however, they were not taking into account more than a couple of features from the entire dataset which gave them results "consistent with previous studies"[4], but almost no interesting insights or breakthroughs.

E. How the proposed project matches with previous work

There are several things in common between the research we found and our proposal as most of the research we found is based on custom collected data or the open data sets similar to the ones we are going to use, however, they either take only one aspect of the data (for example only students mobility) [4] whereas our proposal is to go above just looking at the problem from the student's perspective and that is why we decided to extend our study to propose a view from ERASMUS organisation perspective. The topic itself is not extremely popular and therefore the research on the topic is sparse and spread along the last 20 years and as a result, there is no actual research on the up-to-date data. Some papers focus on a broader range of countries, not only within the European Union [3]. In this paper, we study the Erasmus student mobility and not especially computer science programs focus like the paper [3].

During our investigation of the existing literature we often faced two extremums where most of the papers are presenting qualitative, descriptive research, but, on the other hand, others which we found were focused only on computer science aspects of the data [3]. That is why we aim to make our research using both descriptive analytics and computer science methods so it could fill in the gap between descriptive and purely technical research on the topic.

II. DATA

A. Data source

The used datasets are the official Erasmus mobility statistics 2012-13, which has been published in 2015 on the open dataportal of the EU [2]. It is delivered as a csv file.

B. Data description

Dataset representing students mobility includes 34 Features such as the student ID, home institution and ECTS Credits and 248153 Instances represent the exchange students. It has both categorical (student subject area, gender) and continuous (length of the study period, number of ECTS credits) features. The file contains missing values, which are marked witch '?' or Unknown.

The dataset students mobility dataset contains 3 types of mobility:

- Study mobility
- Placement mobility
- Study + Placement mobility

Not all of the attributes are present for all 3 mobility types. The distribution of instances are following. Study mobility - 211,519 instances, Placement mobility - 55,552 instances, Study with placement mobility - 476. During this project we primarily focused on Study mobility type combined on the study part with the study with placement mobility type.

III. METHODS & TOOLS

A. Initial data prepossessing

While studying our dataset we realised that there is a need for clean up of the data. As we decided to focus on study placements we first had to filter those from the whole dataset. Then some adjustments had to be made to remove unnecessary and meaningless attributes. The following actions were taken

1) Garbage attributes reduction: Most of the attributes in our dataset do not have self-explanatory names, therefore there is an additional dictionary supplied along with the dataset. While studying the dictionary we realised that the following attributes are unnecessary and do not play important role for further analysis.

- ID_MOBILITY_CDE - ID of a placement

- CONSORTIUM_AGREEMENT_NUMBER - Abstract agreement number not relevant for our objectives
- SPECIAL_NEEDS_SUPPLEMENT_VALUE - As we do not separate students with / without special this attribute is not useful for us
- SHORT_DURATION_CDE - This attribute shows whether student returned home due to "force majeure". We do not consider this special case
- QUALIFICATION_AT_HOST_CDE - Whether student will receive any kind of qualification (degree) at the host university. This attribute is not relevant to our objectives as it depends not on a university, but mostly on time student is participating in the program.

2) Work placements attributes reduction: As we consider only students with study placements we need to get rid of all attributes related to work placement information.

- MOBILITY_TYPE_CDE - Flag used to separate study and work placements. We no longer need it because we have already selected only study placements.
- PLACEMENT_ENTERPRISE_VALUE
- PLACEMENT_ENTERPRISE_CTRY_CDE
- PLACEMENT_ENTERPRISE_SIZE_CDE
- TYPE_PLACEMENT_SECTOR_VALUE
- LENGTH_PLACEMENT_VALUE
- PLACEMENT_START_DATE
- ECTS_CREDITS_PLACEMENT_AMT
- PLACEMENT_GRANT_AMT

3) Missing values in our dataset: Missing values in our dataset are represented as 0 if attribute is continuous and as one of the following strings for categorical attributes.

- "? Unknown ?"
- "???"
- "?"

As most of the remaining attributes in our dataset are categorical we manually checked all the continuous ones and wrote a procedure to programmatically check all the categorical ones. If any missing values were detected the warning would be printed to the console. We were not able to find any missing values in the selected subset of attributes. Most missing values in the dataset are caused by the fact that it contains 3 types of placement instances and there are some attributes which are not shared among all the 3 instance types.

4) Attributes clean up: Finally, we noticed a few attributes which needed some cleaning. First of all, there was one country in the dataset represented by three different country codes and so we grouped all three of them under one code. It was Belgium which was initially split into:

- "BEDE"
- "BEFR"
- "BENL"

We grouped them under "BE" code. Secondly, many languages were represented by multiple spellings of the language code, for example initially we could have the following options for English language "En", "EN", "en". This was fixed by making all language code values upper case. Lastly, the "STUDY_GRANT_AMT" attribute, which represents total amount of scholarship support received by student, was initially stored as character string representing an integer number. We had to convert it to numeric field taking into consideration two possible decimal positions.

B. Classification

One of the primary goals we tried to achieve was to improve overall user experience with the ERASMUS program and especially first steps when student needs to select destination country and university by trying to suggest best options based on the similar student profiles from the previous year. This hypothetical program can be used both by students themselves or by ERASMUS program coordinators at each university participating in the program. Our task was to predict "HOST_INSTITUTION_COUNTRY_CDE" attribute values.

1) Preprocessing: For classification task we did several things at data preprocessing stage such as removing attributes directly correlated with the target variable, features selection and trying to address class imbalance. We found out that "HOST_INSTITUTION_CDE" is directly correlated with the target variable and therefore we had to remove it from the list of used features.

2) Features selection: For both classification methods described below we used the same features selection algorithm: Sequential Forward Selection[6] (SFS). SFS is a simple features selection method which starts with an empty set of features at each iteration it tries to find the best feature and add it to the set. We repeat process until there is not performance gain. To validate features selection method performance we used Objective Function[6] with wrapper approach using Naive Bayes as validation Machine Learning method. Additionally, 5-fold cross validation is used inside features selection to ensure the best accuracy.

3) Addressing class imbalance: During the preprocessing stage we decided to investigate our classes in details. The first thing we could see was that there exist a significant class imbalance (see Figure 1). As most of our attributes are categorical the standard SMOTE method would not work for us well and therefore we decided to implement custom under and over sampling method ourselves to restore balance in our classes.

Our method works in the following way:

- Select 10 target variable values which have the lowest number on instances in the training set (bottom10)
- Select 10 target variable values which have the highest number on instances in the training set (top10)
- For each target value from the bottom10 list randomly with replacement double the number of instances per value.
- Add these duplicated instances to the resulting dataset
- Take randomly 30% of the instances which have value for the target variable equal to one from top10
- Remove instances taken in the previous step from the target dataset
- Return the target dataset

To ensure the best results we repeat this method couple of times. Class balancing is optional and can be configured using parameters.

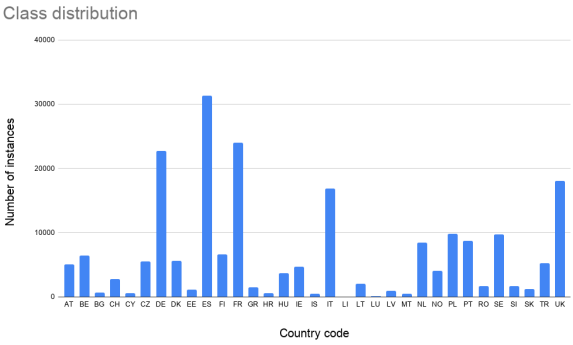

Fig. 1. Class Distribution

4) Naive Bayes: The first classification method we applied to our problem was Naive Bayes. Naive Bayes is a supervised machine learning algorithm which uses Naive version of Bayes Theorem [7]. I counts conditional probability for each possible value of the target variable and then selects the value with highest probability as a result. We decided to use it, because it works well with categorical data and is easy to understand and implement.

5) Random Forest: The second classification method we selected was Random Forest [9]. Random Forest combines Begging and Decision Tree into single supervised Machine Learning method. It is achieved my building some amount of trees using training dataset and then running test instances down those trees in order to classify them.

6) Validation methods: We use 5-fold cross validation for the Naive Bayes method. Due to significant time needed to build the trees for the Random Forest we could not use cross validation for it and therefore we are just using training and validation set split to test Random Forest accuracy. The split parameters can be adjusted, the default ones are 70% for training and 30% for testing.

Other than that, we could previously see that our dataset is imbalanced and then we realised that accuracy cannot be the only performance measure for such an imbalanced dataset. Therefore, we introduced f-score. F-score measures performance of the model using precision and recall measurements which works better for imbalanced datasets.

C. Clustering

K-means clustering set a data set into k groups or clusters. The different between K-means and k-medoids clustering is that in k-medoids each cluster is represented by one of the data point in the cluster. It is a robust alternative to k-means clustering. The reason is that the algorithm is less sensitive to noise and outliers, compared to k-means. The K-medoids version we use is the Partitioning Around Medoids algorithm (PAM) which is originally from Kaufman and Rousseeuw 1990.

For clustering we create a new table of universities using the information out of the given Erasmus data set. The following list shows the used features:

- Number of universities which have connection to this university HOME_INSTITUTION_CDE
- Number of students which came to this university
- Average/Median age of student (We can use median to overcome outliers) per university STUDENT_AGE_VALUE
- Number of programs per university STUDENT_SUBJECT_AREA_VALU
- Average grant amount per university STUDY_GRANT_AMT
- Number of languages per university LANGUAGE_TAUGHT_CDE
- Average number of ECTS points for student per university ECTS_CREDITS_STUDY_AMT
- Average/median (TO DECIDE) length of the program per university LENGTH_STUDY_PERIOD_VALUE
- Percentage of people who failed the program, proportion (failed / total) per university

The data set has to be normalized before applying clustering.

D. Tools & Libraries

We decided to use R programming language and R-studio IDEA. R comes with a good number of methods embedded within its standard library, however, we still needed to include some libraries.

We use the following libraries in our project:

- naivebayes - Naive Bayes implementation
- randomForest - Random Forest implementation
- cluster, factoextra - Silhouette width and Clustering

Most of the data preprocessing and validation techniques such as fixing class imbalance, cross validation and features selection was implemented by us ourselves in order to practice implementing those methods and effort to understand them better.

IV. Results

A. Descriptive analytic

Descriptive analytic is a way to interpret data using basic statistics tools to get to know the domain. The distribution of the original data set is 55552 Placement, 211519 Studies and the amount of 476 combined. In the following we will only focus on the student placements. The Top sending countries are Spain (39249 students), France(35311), Germany(34891), Italy and Poland. The other way around the top receiving countries or institution are mostly the same countries with Spain(31360), France(23970), Germany(22728), UK and Italy. Moreover the average Duration is 6.2 Month for Students. like expected the top thought Languages are English (117260) Spanish(23419) and French (22437).

B. Classification Results

Before starting a discussion about numeric results we should first discuss the parameters used for each method. Naive Bayes implementation uses only two hyperparameters:

- Laplace smoothing coefficient - used to give Naive Bayes ability to work with 0 probabilities in some cases we are using value 0.5
- k - number of folds to use in cross validation. k is set to 5 and cannot be changed

Random Forest uses slightly more hyperparameters:

- Number of trees - number of trees to construct during training process the value we are using is 300
- mtry - number of attributes randomly sampled at each split we are using 3
- percentage_train - percentage of the dataset which should be takes as a training data

Additionally, as we have 33 possible values for the target variable we use top-1, top-2 and top-3 accuracies to measure performance of the proposed models, because out main goal is not to suggest 1 best choice for a student, but rather to give him a list of options to select from on one hand, but at the same time narrowing it down from 33 countries to a smaller number to make their choice easier and overall experience better.

The tables below show the results for various runs of Naive Bayes and Random Forest classification methods. The first table I represents Naive Bayes performance on the imbalanced dataset. The second table II presents results of Naive Bayes after our class balancing method was used. The third table III shows Random Forest results for an imbalanced dataset. The final table IV in classification results represents Random Forest results for a balanced dataset. Figure 2) visualises a comparison between average Naive Bayes and Random Forest performance for the problem under study.

The table I shows the results of the classification with an imbalanced data set.

Overall, we can see that the implemented method for class balancing does not work very well. The average performance of the Naive Bayes algorithm is around 58 - 59% for the top-1 accuracy, slightly more than 68% for top-2 accuracy and around 75% for the top-3 accuracy. In case of balanced version both accuracy and f-score are lower compared to the unbalanced version. All average results are calculated on at least 3 runs of the program. The average performance for Random Forest was very similar to the one of Naive Bayes for the f-score and top-1 accuracy. However, for top-2 and top-3 accuracy the average performance dropped significantly by around 2 and 5% respectively. There was one very successful run for the Random Forest where we got the all time high on all three accuracies and f-score the top-3 accuracy was 75.33% whereas top f-score was 0.362. Finally, we tested our Random Forest model on the balanced dataset and similarly to the Naive Bayes got worse results. The drop of average accuracy for the top-1 accuracy was 6%, for top-2 3%, for top-3 2% and for the f-score 0.02.

C. Clustering Results

Looking at instances in cluster we see that some features are unnecessary to take into consideration, because all of

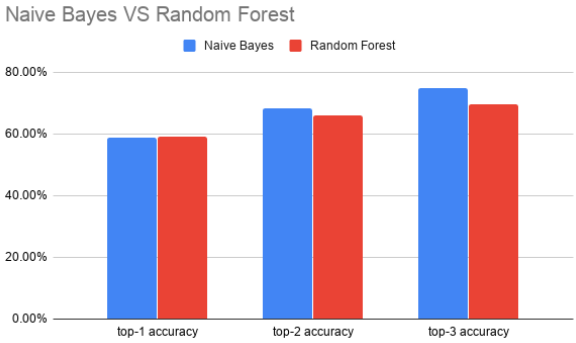

Fig. 2. Performance Comparison

	Accuracy [%]			F1 score
	top-1	**top-2**	**top-3**	**top-1**
Run 1	58.7	69.32	74.97	0.342
Run 2	58.73	69.23	74.93	0.334
Run 3	58.74	69.27	74.97	0.335
Average	58.72	68.27	74.95	0.337

TABLE I
CLASSIFICATION RESULTS NAIVE BAYES - IMBALANCED DATA

	Accuracy [%]			F1 score
	top-1	**top-2**	**top-3**	**top-1**
Run 1	51.34	62.44	68.57	0.291
Run 2	51.41	62.58	68.73	0.295
Run 3	58.38	62.62	68.76	0.298
Average	51.37	62.54	68.68	0.294

TABLE II
CLASSIFICATION RESULTS NAIVE BAYES - BALANCED DATA

	Accuracy [%]			F1 score
	top-1	**top-2**	**top-3**	**top-1**
Run 1	58.12	63.41	66.33	0.31
Run 2	58.51	63.92	66.86	0.315
Run 3	61	70.48	75.33	0.362
Average	59.21	65.93	69.5	0.329

TABLE III
CLASSIFICATION RESULTS - RANDOM FOREST IMBALANCED DATA

	Accuracy [%]			F1 score
	top-1	**top-2**	**top-3**	**top-1**
Run 1	52.38	62.17	66.91	0.302
Run 2	52.59	62.23	67.06	0.3
Run 3	54.17	63.83	68.21	0.309
Average	53.04	62.74	67.39	0.303

TABLE IV
CLASSIFICATION RESULTS - RANDOM FOREST BALANCED DATA

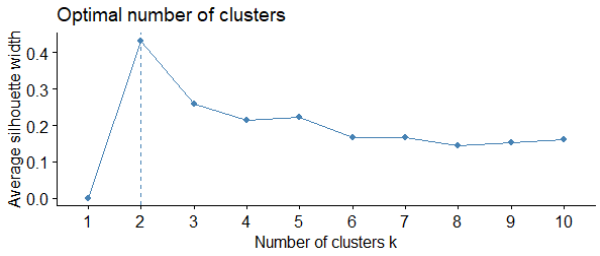

Fig. 3. Silhouette width

the instances under study have very similar values in those features. The features that do not differ much among universities are median age, grand amount and the number of ECTS points. Additionally, we can conclude that during 2012/2013 year the average amount of funding per student did not depend on the size of the university.

When we used Silhouette width to determine the best number of clusters we were initially surprised that it returned best k = 2 (figure 3) for the 2400+ instances we have. After analysing results we realised that the best k which Silhouette width analysis returned works as a filter between bigger and smaller universities. Moreover, in our dataset bigger and smaller universities can be easily distinguished by the following parameters: number of students, study length, number of connected universities, number of subjects students came to study on and number of languages of instruction at each university. Our clustering solution show that there is a gap between universities of different size and that the number of intermediate sized universities is rather small. Additionally, looking at the number if instance in each cluster we can conclude that the number of big universities is almost 4 times less than the number of small ones on the ERASMUS program.

V. Conclusion

A. What has been done

Classification analysis was applied to predict the top 1 up to top 3 destination for students in the EU. It was done using Naive Bayes and Random Forest. The most suitable destinations countries aims to improve the overall experience for the students participating in the Erasmus program.
Clustering analysis was applied to group universities based on their relative performance. We used k-medoids algorithm to perform clustering on the aggregated data taken from the original dataset to try to see patterns in the universities participating in the program and their relative performance to help the Erasmus coordinators to make more informed decisions.

B. Summary of the results

We were able to apply two methods to solve the classification problem. The first method was Naive Bayes. In this class-imbalanced dataset for Naive Bayes top-3 accuracy was 74.95% and random Forest top-3 accuracy: 66.64%. In this balanced dataset, for Naive Bayes top-3 accuracy: 68.547705% and random Forest top-3 accuracy: 67.39%.

Overall, we consider the achieved results in classification to be good, as we tried to solve a new problems and learned a lot while approaching it from different angles trying to improve the results. Among the two methods we applied the performance is quite similar whereas Naive Bayes gives us better average results and Random Forest performed best in single run comparison.

The Clustering analysis shows that there is clear division on bigger and smaller in terms of participating universities in the ERASMUS program and that one way to handle them would be to divide into 2 groups and approach them separately.

C. Future Work

In future, we shall focus on the comparison between student, staff, and teachers mobility placements. All those datasets are in open access. There is also the opportunity to compare the results of 2012-2013 to other years.

Acknowledgment

We would like to thank Dr. Quazi Abidur Rahman for guiding us through this research study and enlightening us with his in depth knowledge.

Appendix I
Reproduction of our work

During the development of the project we were using R Studio and would suggest using R Studio for verifying it would be a recommended choice. Our code structure looks as follows:

- main.R - main file of the project code containing core function **run_program()**
- classification.R - R file containing functions related to the classification task
- clustering.R - R file containing functions related to the clustering task

Before starting you have to install the following packages:

- naivebayes [8]
- randomForest [10]
- factoextra (install.packages(c("cluster", "factoextra")))

To run the code you should open the main.R script, source it into R and execute run_program() function. The program should start in its default mode: Classification task using Naive Bayes method.

There are various parameters available for run_program() function, for example the program mode, which is 0 for Classification and 1 for Clustering. The second parameters is for the classification run. With choosing a 0 it will use the classification_method Naive Bayes and with 1 Random Forest. So, for example when you decide to run classification with Random forest just execute run_program(0,1). The third parameter is for the mode Clustering. It represents k and will define the value of clusters. For clustering results please run run_program(1). For more details on run_program function please check the documentation in the code.

The code may produce intermediate results during clustering task, all of them would be stored in the 'data' directory.

REFERENCES

[1] https://www.erasmusprogramme.com/post/what-is-the-erasmus-programme

[2] https://data.europa.eu/euodp/en/data/dataset/erasmus-mobility-statistics-2012-13

[3] Analysis of Staff and Student Mobility Network within a Big European Project, Miloš Savić, Mirjana Ivanović, Zoran Putnik, Kemal Tütüncü, Zoran Budimac, Stoyanka Smrikarova, Angel Smrikarov,MIPRO 2017, May 22- 26, 2017, Opatija, Croatia

[4] Breznik, Kristijan & Skrbinjek, Vesna & Law, Kris & Đaković, Goran. (2013). ON THE ERASMUS STUDENT MOBILITY FOR STUDIES.

[5] Mizikaci, Fatma; Arslan, Zülal Uğur. A European Perspective in Academic Mobility: A Case of Erasmus Program. Journal of International Students . 2019, Vol. 9 Issue 2, p705-725. 21p.

[6] http://research.cs.tamu.edu/prism/lectures/pr/pr_l11.pdf

[7] https://plato.stanford.edu/entries/bayes-theorem/

[8] https://github.com/majkamichal/naivebayes

[9] https://www.stat.berkeley.edu/ breiman/RandomForests/cc_home.htm

[10] https://cran.r-project.org/web/packages/randomForest/randomForest.pdf