What are the individuals more likely to have an income below or equal to the average?

Ana Sofia Silva
m20200220@novaims.unl.com
Bussiness Analytics

José Francisco Alves
m20200653@novaims.unl.com
Business Analytics

Miguel Nunes
m20200615@novaims.unl.com
Business Analytics

Mohammadali Gharghi
m20200997@novaims.unl.pt
Business Analytics

Abstract

This project aims to implement supervised models in order to predict what are the more likely people to have an income below or equal to the average. The dataframe given has informations relative to a sample of selected individuals. In order to predict their tax rate based on their income a set of steps were handled in order to extract, transform, load and analyse the dataframe. After the data cleaning and analysis, a group of models where selected to be tested with suitable parameters. After several attempts of testing in Kaggle the model with higher score was Voting Classifier, with a final score of 0.86567. The structure of this report is divided in Introduction, Backgroud, Methodology, Results and Conclusion.

Keywords: Project, Machine Learning, Feature selection, Predictive Models, Voting Classifier.

1. Introduction

A new planet has been discovered in the galaxy and there is a project to move people there and make a new land. Government wants to take tax from people based on their income. There is a group of people who took their salary for the first month. We need to create a

model to match new people with the first people's group to find out their tax ratio for the first month.

For our project we selected a total of 46 variables, that included ordinal, numerical and binary types. The prediction variable - *Income* - is categorical and binary indicating 0 for "lower or equal to the average" and 1 for "higher than the average". The applied models were selected according to the variables input and output types. Most of the theoretical support for the used classification models was presented in the Machine Learning class, although, it was explored the Voting Classifier and K-Nearest Centroid.

2. Background

This section presents models that have not been explored during the practical classes and that are applied in the project.

2.1 K-Nearest Centroid (KNC)

The KNC classifier is a simple algorithm that represents each class by the centroid of its members. It has no parameters to choose, making it a good baseline classifier.

The distance in the feature space between the query instance and each K centroid is computed and the prediction returned by the model is the target feature level of the centroid that is nearest to the query in the feature space.

2.2 Voting Classifier

A collection of several models working together on a single set is called an ensemble. The method is called Ensemble Learning. It is much more useful to use all different models at the same time as it provides lower error and less over-fitting [4] [1].

A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output based on their highest probability of chosen class as the output. The idea is instead of creating separate dedicated models and finding the accuracy for each them, we create a single model which trains by these models and predicts output based on their combined majority of voting for each output class [4] [1].

Voting Classifier has two types of votings: Hard Voting and Soft Voting. In Hard Voting, the predicted output class is a class with the highest majority of votes On the other hand, in Soft Voting, the output class is the prediction based on the average of probability given to that class [4] [1].

3. Methodology

First, we start by inserting the essential packages to initialize the work project. Therefore, we started to explore the data in order to understand its behaviours and dimensions. The univariate and multivariate analysis was previously assessed to study the distribution and shape of the dataframe.

In the next step, the features were transformed and new variables were created according to its measures.

To perform this work project, the Jupyter Notebook platform belonging to the Anaconda Navigator was used. We applied python programming language to build codes and models. The work project report was made through an online editor Overleaf. Machine learning books and a free software machine learning library for the Python programming language named Scikit-learn were also used.

4. Results

4.1 Sample

In the first phase of our project, we have focus on importing the needed libraries, train and test dataframes. In order to extract and transform the dataframe it was imported Pandas, Scipy and Numpy. Another set of libraries were imported with the aim of dataframe visualization and preparation (Numpy, Scipy, Matplotlib and Seaborn). Sklearn was used in order to implement the set of predictive models, feature selection, train our models and assess their results.

4.2 Dataframe Exploration

The presented dataframe has a total of fifteen features which describe a sample of selected individuals chosen for the new planet. In this section it was handled a set of steps which gave an in-depth picture of the data, in order to get more information about it [2]. According to this, this sections analyse typical data quality issues as missing values, outliers treatment and inappropriate level for a feature.

The first phase of this process is to identify the people who belong to each class, according to their income.

The initial dataframe included qualitative and categorical features. A table of the data exploration (Table 2) was assessed in order to describe the characteristics of a portion of features using standard statistical measures of central tendency and variation. The complete table is presented on the Jupyter file.

Group	Description	Value
A	Selected by diversity (Group A)	1
B	Payed to participate (Group B)	2
C	Rejected but payed (Group C)	3

Table 1: Group and the correspondent description and value associated in the further analysis.

Lives with	Base Area	Education Level	Years of Education	Employment Sector	Role	Working Hours per week	Money Received	Ticket Price	Income
22400	22400	22400	22400.000000	22400	22400	22400.000000	22400.000000	22400.000000	22400.000000
6	40	16	NaN	9	15	NaN	NaN	NaN	NaN
Wife	Northbury	Professional School	NaN	Private Sector - Services	Professor	NaN	NaN	NaN	NaN
9012	20074	7232	NaN	15599	2849	NaN	NaN	NaN	NaN
NaN	NaN	NaN	13.173884	NaN	NaN	40.483795	1324.915357	109.145313	0.237098
NaN	NaN	NaN	2.512451	NaN	NaN	12.370921	9227.771813	500.208904	0.425313
NaN	NaN	NaN	2.000000	NaN	NaN	1.000000	0.000000	0.000000	0.000000
NaN	NaN	NaN	12.000000	NaN	NaN	40.000000	0.000000	0.00	0.000000
NaN	NaN	NaN	13.000000	NaN	NaN	40.000000	0.000000	0.000000	0.000000
NaN	NaN	NaN	15.000000	NaN	NaN	45.000000	0.000000	0.000000	0.000000
NaN	NaN	NaN	21.000000	NaN	NaN	99.000000	122999.000000	5358.000000	1.000000

Table 2: Descriptive statistics of the initial dataframe.

As the Table 2 shows, the dataframe needed some cleaning and modification. There are approximately 12% of missing values and different metrics on the features. There were characters has '?' that has been replaced with NaN.

4.3 The structure of a data quality plan

Feature	Data Quality Issue
Citizen_ID	Irrelevant information for prediction
Years of Education	Outliers
Education Level	Outliers
Name	Irrelevant information for prediction
Birthday	Need of feature transformation
Native Continent	Need of feature transformation
Marital Status	Need of feature transformation
Lives With	Need of feature transformation
Base Area	High cardinality and Missing values
Employment Sector	Missing values
Role	Missing values and High cardinality
Total Price	Outliers
Working Hours per Week	Outliers
Money Received	Outliers

Table 3: Data quality report.

4.4 Transform and Create variables

The main purpose of this section is to prepare the dataframe for the predictive models implementation. A set of transformations and encoding were made to create ordinal, nominal and binary features.

It was manually created the following variables:

- To have in consideration the gender of each sample, it was created a binary variable *Gender* with the integer 0 associated with women and 1 with men. The name of each individual was not considered relevant to our study case.
- As for *Birthday* the numerical variable *Age* was created from the year information in *Birthday*.
- *Education Level* was transformed into a ordinal variable with the range 1-16 indicating the low level of education to 1 and there successively.
- *Group* was created according to the project presentation rules (Table 3).

The variable *Citizen_ID* which corresponds to the unique identifier of each citizen and, for that is deleted, due to its irrelevancy in the predictive models.

Dummy variables were created for *Native Continent*, *Marital Status*, *Lives With*, *Employment Sector* and *Role* with the function "set_values_as_columns". This new variables

are binary and indicate the logical value for each observation with true associated to 1 and 0 to false.

4.5 Univariate Analysis of the original features

4.5.1 Continuous Features

For each feature, we should examine the central tendency and variation to understand the types of values that each feature can take. The Figure 1 shows the histograms of *Total Price*, *Years of Education*, *Working Hours per week* and *Money Received*.

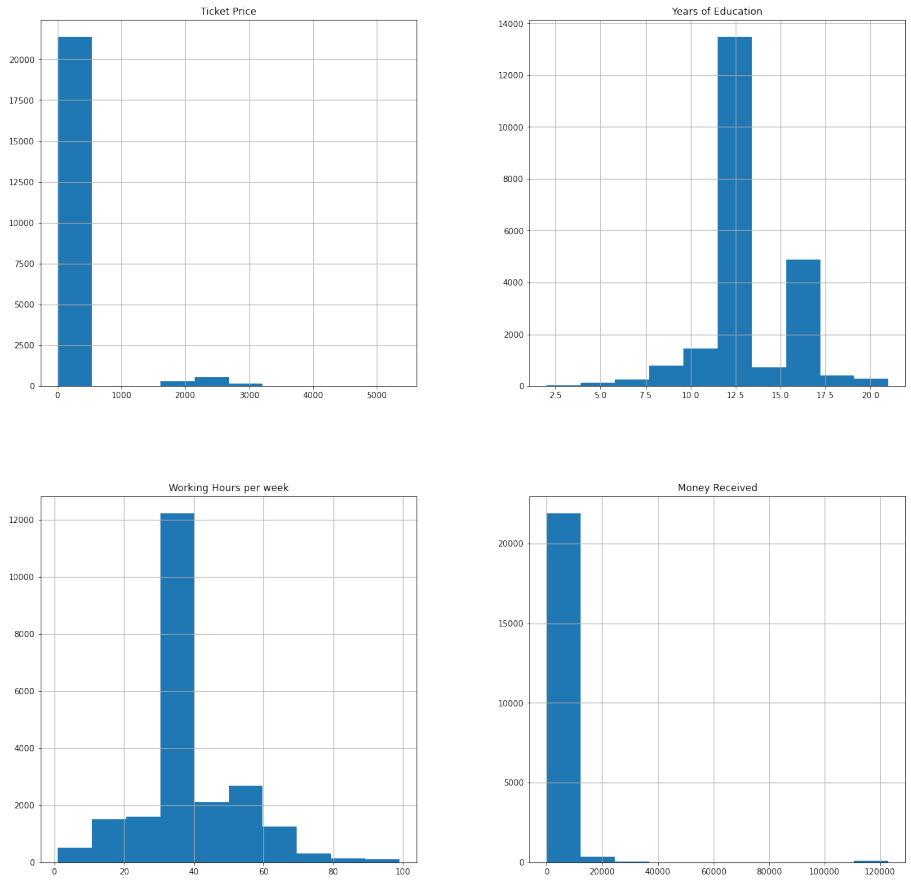

Figure 1: Histograms for *Ticket Price*, *Years of Education*, *Working Hours per Week* and *Money Received* variables.

The Figure 2 shows the histogram of *Education Level* regarding the transformation of the feature type.

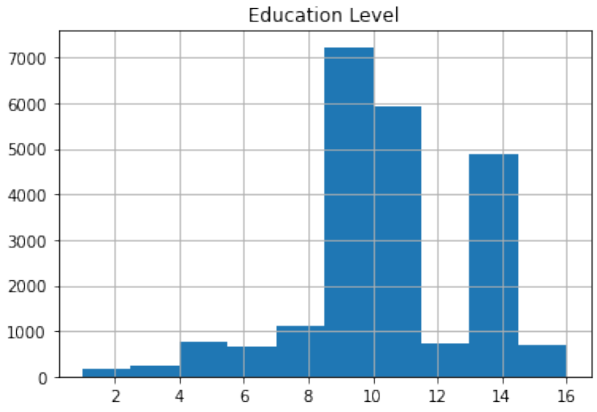

Figure 2: Histogram for *Education Level*.

The boxplots visualization (Figure 3) were assessed to analyse the previous distributions and the outliers presence.

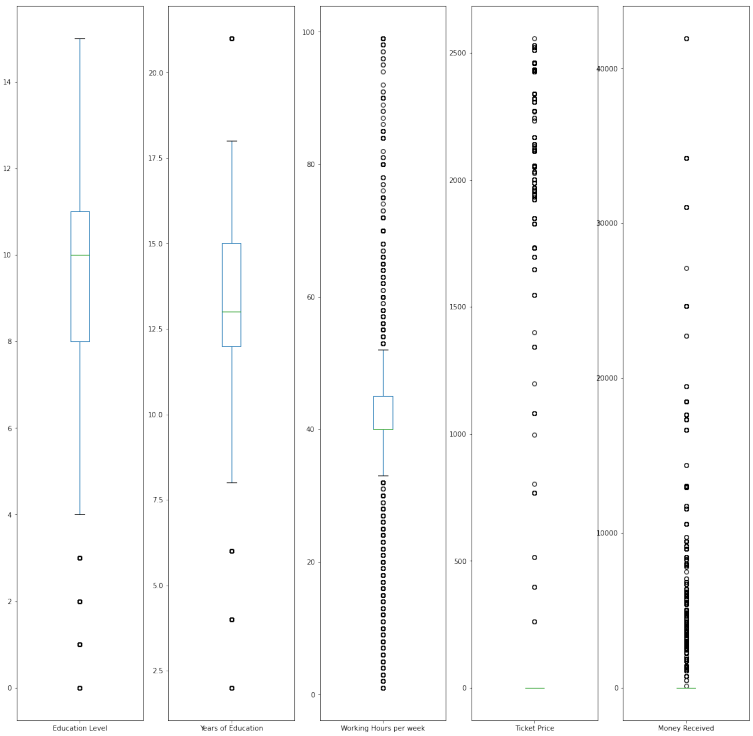

Figure 3: Boxplots for *Ticket Price*, *Years of Education*, *Working Hours per Week* and *Money Received*features.

From the histograms in Figures 1 and 2, we see that the continuous features *Years of Education* AND *Age* seem to follow a normal distribution. A bi-modal distribution with

two clear peaks is clear in *Years of Education* and *Education Level*. Also, according to Figure 3 the median in *Money Received* and *Ticket Price* it's near zero presenting in both cases (positive skewness), a high presence of outliers and a low dispersion.

4.6 Multivariate Analysis of Continuous Variables of the original features

In this section, we concluded that there's no linear relationship between the continuous variables.

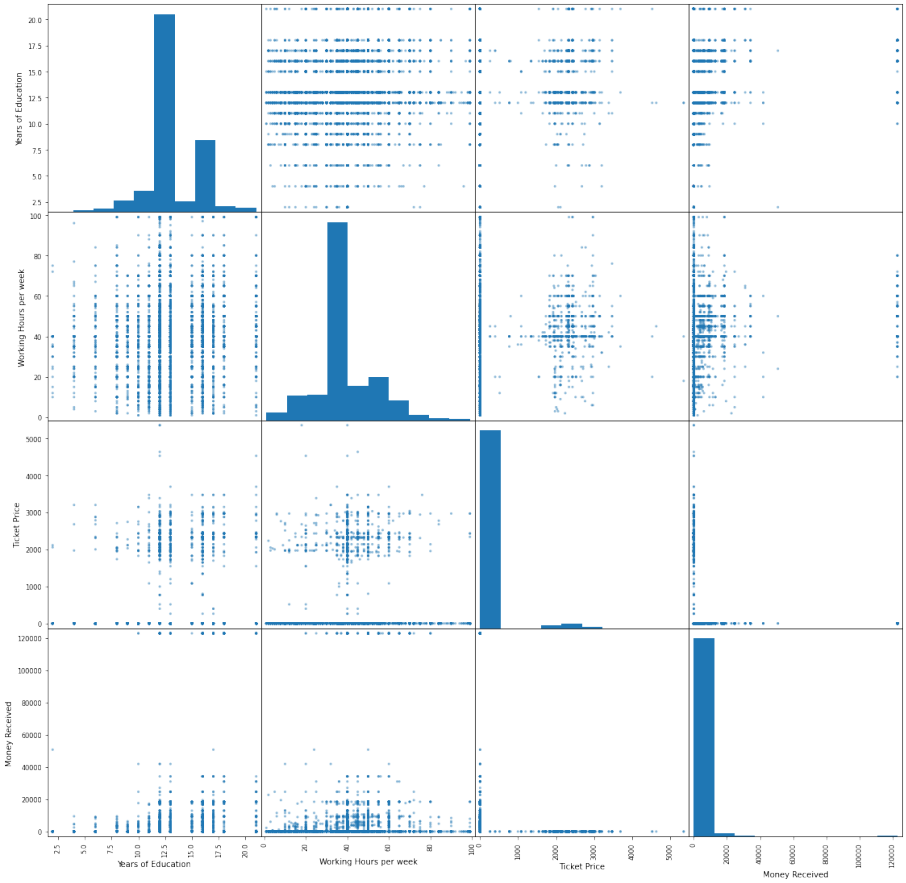

Figure 4: Boxplots for *Ticket Price*, *Years of Education*, *Working hours per week* and *Money Received*.

4.7 Outliers Analysis

In this section our first step was to choose non categorical columns and then identified outliers through boxplot and histogram visualizations, where we compare odd values against

the pattern of the data. According to the boxplots, there are a lot of observations outside the minimum and maximum values, which may be considered as outliers. There were 5 non categorical columns:

- *Money Received*
- *Ticket Price*
- *Working Hours per Week*
- *Years of Education*
- *Age*

We knew that *Money Received* and *Ticket Price* are applied to a group of observations, so the rest of observation would be zero on that fields and counted as an outlier. So, we decided not to take them for consideration.

A commonly used approach to setting the upper and lower thresholds is to use the mean value of a feature plus or minus 1.5 times the standard deviation. To decide a more robust threshold to set the cut of the outliers, otherwise there would be a high percentage of data included as it, it was set 3 as the threshold.

According to the chosen method, no outlier detected in *Age* and *Years of Education* fields and the only field with outlier was *Working Hours per Week*. We took three approaches for this field: Removing the fields, marking them out in a separated column named *outlier* and do nothing about them and checked the final score for all of them. The best result has been with taking no action with those outliers.

4.8 Fill Missing Values

We started by checking the existence of missing values as can be seen in the notebook.

We were able to detect 3 variables - *Base Area*, *Employment Sector*, *Role* - where this happened. The variable *Base Area* was not selected for the model due to its high cardinality and irrelevancy.

For the rest of fields, we counted the unique values and made a boolean column for each group and set it to 1 if the observation was set to this value. Furthermore, the missing values got weight of 0 on those expanded fields.

4.9 Correlation Analysis

The first step in trying to reduce the number of predictors is using the correlation matrix and measure the relevance for predicting the outcome variable. With this knowledge, the

set of predictors should be reduced to a sensible set that reflects the problem at hand in order to check if there are two or more variables explaining the same or almost the same information, we performed the correlation analysis. [3] Some practical reasons for predictor elimination are high correlation with another predictors with the aid of correlation matrix. In our case study, the chosen correlation method was Spearman due to its higher correlation and because of a non linear relationship between the variables.

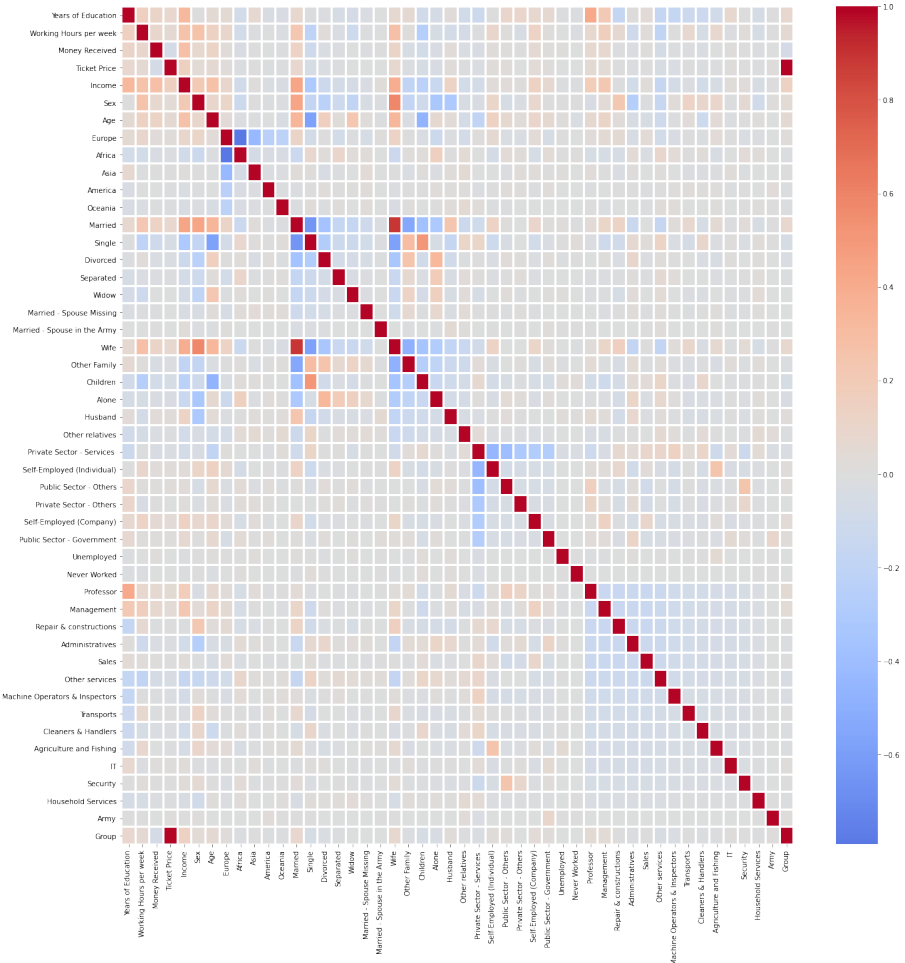

Figure 5: Correlation matrix with Spearman correlation.

From the correlation analysis in Figure 5 some conclusions can be learned in what concerns feature selection, once this value suggest that those variables may retain the same information. As takeaways, from the visualisation presented, some attempts in the final models were done with these decisions:

- Include the *Ticket Price* or the *Group* due to the high correlation;
- Exclude *Europe*, *Africa*, *America*, *Asia* and *Oceania* due to the low correlation.

In the final model, we decide to not exclude any variables due to higher scores with the test dataframe.

4.10 Train Validation Partition

Train validation partition provides a set of data where we can train the model and validate the results in another set. A training set is the subsection of a dataframe from which the machine learning algorithm uncovers relationships between the features and the target variable. A validation set is another subset of the input data to which we apply the machine learning algorithm to see how accurately it identifies relationships between the known outcomes for the target variable and the dataframe's other features. Partitioning data into training and validation sets allow us to develop highly accurate models that are relevant. We define 25% of the data as validation set where the distribution of the dependent variable was maintained in both partitions.

In our final model the only excluded variable was Base Area. This decision was complemented with Lasso and Ridge Regression graphs analyses.

4.11 Data Standardization

Having continuous features in a dataframe that cover very different ranges can cause difficulty for some machine learning algorithms due to the different contribution by each model. In order to avoid these difficulties, we standardized the data for our features to have the same impact.

We tested two scaling techniques, Standard Scaler and Robust Scaler, however we chose only standard scaler because it gave better results and because Robust Scaler doesn't take the median into account and only focuses on the parts where the bulk data is.

4.12 Feature Selection

A group of features were tested according to LASSO and RIDGE analysis during the project elaboration. Initially we choose all variables to do the split and each time feature selection was accomplished we returned to Train Validation Partition step in order to select the most appropriate features. The final model ended with the exclusion of Base Area.

4.13 Tested Models

In this section were tested a total of 17 models and ensembles. The accuracy was, in general, registered for each of them, and it was implemented to find the best parameters and to run the best score in train and validation with the aim of avoiding overfitting. It was considered

a list of possible parameters per each model and the function *GridSeachCV* had the aim of search a estimator to fit suitable parameters. The best score was 0.86567 with Voting Classifier.

5. Conclusion

In the elaboration of this project it was possible for our team to have an overview of the Machine Learning subject, including exploring new tools of Sklearn.

The data preparation and preprocessing were an essential phase of the project because the quality of the final dataset will influence all the following analysis. The treatment of outliers were crucial in order to maintain the consistency and coherence of the original data without losing too much information. Furthermore, the data partition was tested repeatedly with different sets of variables based on the analysis of the feature selection methods and the accuracy obtained, not only, in the validation set, but also in the test set.

Subsequently we started by training our model by using algorithms such as Decision Trees, Logistic Regressions, Support Vector Machines, Voting Classifier, Gradient Boost, different ensemble methods, and others, until we reach a final prediction based on the 25% of the test dataset, we can assess on kaggle.

References

[1] Geeks for Geeks. *ML — Voting Classifier using Sklearn*. URL: `https://www.geeksforgeeks.org/ml-voting-classifier-using-sklearn/`.

[2] Jiawei Han and Micheline Kamber. *Data mining : concepts and techniques*. San Francisco [u.a.]: Kaufmann, 2005. ISBN: 1558604898 9781558604896. URL: `http://www.amazon.com/Data-Mining-Concepts-Techniques-Management/dp/1558604898`.

[3] John D. Kelleher, Brian Mac Namee, and Aoife D'Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. The MIT Press, 2015. ISBN: 0262029448.

[4] Sanchita Mangale. *Voting Classifier*. URL: `https://medium.com/@sanchitamangale12/voting-classifier-1be10db6d7a5`.