Exploratory Data Analysis

**Beer Profile and Ratings**

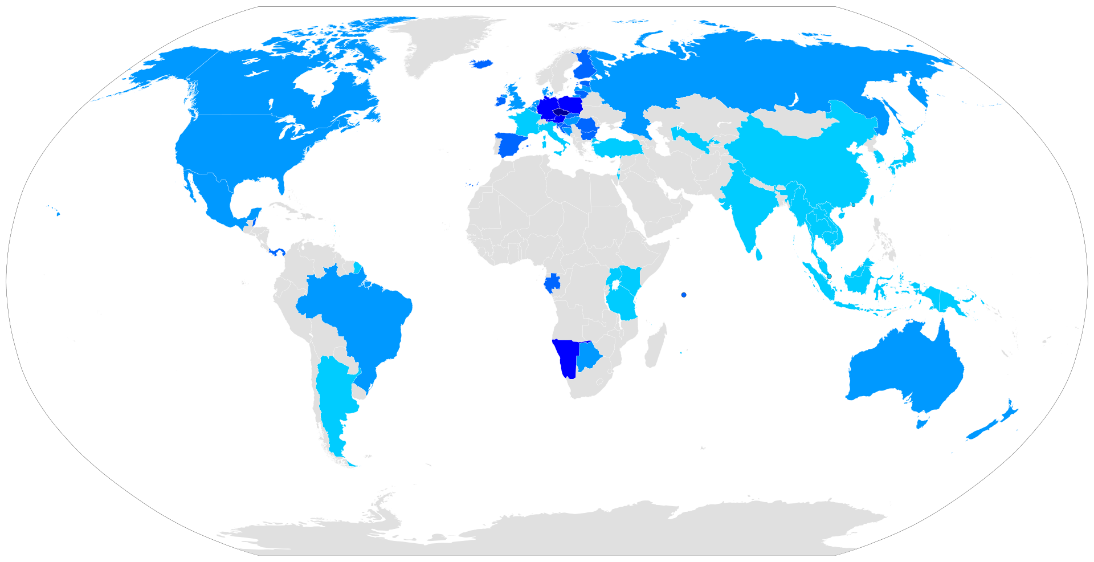
Students:

|  |  |
| --- | --- |
| Istrati Lucian – Master 411 | Patilea Catalina Camelia – Master 411 |
| Popescu Ionut-Alexandru – Master 411 | Zaharia Stefan Tudor – Master 411 |

1. Introduction

This documentation is intended to describe the process we went through while developing the project for the Exploratory Data Analysis Lecture.

For the beginning, the project consists in ***2 parts***: an exploratory data analysis, made in R, and machine learning algorithms, developed in Python.



|  |
| --- |
| Beer consumption per capita by country (2018) |
| ≥ 125 litres    100–125 litres    75–100 litres    50–75 litres    < 50 litres |
| 1st place: Cehia (188 l/capita)  3rd place: Romania (100 l/capita) |

1. Dataset

The dataset – “*Beer Profile and Ratings Data Set*”- is a public one, taken from Kaggle (<https://www.kaggle.com/ruthgn/beer-profile-and-ratings-data-set>) and concerns ratings and consumer reviews about beers worldwide. There are **3197** unique beers from **934** different breweries, translating to **3197** unique rows and **25** columns.

Columns:

|  |  |  |
| --- | --- | --- |
| * Name * Style * Brewery * Beer Name Full * Description * ABV (alcohol by volume) * Min IBU * Max IBU * Astringency | * Body * Alcohol * Bitter * Sweet * Sour * Salty * Fruits * Hoppy * Spices | * Malty * Review aroma * Review appearance * Review palate * Review taste * Review Overall * Number of reviews |

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedBrief summary of the dataset, undergone in the exploratory data analysis made with Shiny R:

Quick overview of the dataset, concerning only the first few columns:

Table

Description automatically generated

There are some particularities about beers. The first 8 columns describe more the beers and breweries and some technical details (name, style, brewery, description, beer full name, ABV, IBU). The following 3 concern the mouthfeel (astringency, body, alcohol), 4 describe the taste (bitter, sweet, sour, salty) and other 4 are about flavour and aroma (fruits, hoppy, spices, malty). The last 6 columns contain more information about beer reviews, including the number of consumer reviews, overall rating score and more ( review on aroma, review on palate ,review on appearance, review on taste, review overall and number of reviews).

1. Beer Analysis

In order to choose what we wanted to predict, we had to go through most features and see how the data looks like. We started with a wordcloud for the beer naming and observed which words are associated with beers.

A picture containing logo

Description automatically generated

We next went for an analysis of the top 10 Breweries that appear in the given dataset.

As we could see that there are more than 500 different breweries, some of them appearing more or less in the dataset, we had to choose the first 10 most popular. As we can see, “Boston Beer Company (Samuel Adams)” appears 40 times, followed closely by “Dogfish Head Brewery” with 31 beers and “Anheuser Busch” with 30.

Chart, bar chart

Description automatically generated

Chart, bar chart, histogram

Description automatically generatedWe looked next on how the styles are distributed in the dataset.

We examined the first most popular 20 styles, observing again that “Lager” - “Adjunct” and “European Pale”- are the most used styles, with almost 50 beers each in the dataset, followed closely by “Wheat Beer” with almost 45.

Chart, bar chart, histogram

Description automatically generated

In order to understand things better, we also looked which are the least preferred 10 styles by our consumers, and we discovered that there are less than 5 beers each with styles “Brett Beer taste”, “Sour – Gose” and “IPA-New England”. Those are some special types of beverage that are not distributed all over the world and are less consumed due to their specific taste or content.

Chart, bar chart

Description automatically generatedWe continued with the first numerical column, ABV. In translation, ABV stands for “Alcohol by volume” and is a standard measure of how much alcohol (ethanol) is contained in a given volume of an alcoholic beverage, expressing everything as a volume percent.

Graphical user interface, text

Description automatically generated The figure above represents distribution of beers with a certain ABV, expressed in percentages (the y axis). Because as we could see in the first table, there are continuous values, we needed to create a standard according to the information we found on Wikipedia concerning the concentration of alcohol among beverages. So, the legend for the buckets is listed as it follows:

We can see that most beers have their ABV between 4 and 10, which is normal according to the legend, because that is the interval for normal beers and cidres. There are also some outliers, for example the alcohol free or low alcohol drinks, as well as the ones that have a very high volume of alcohol more than 10, similar to wine and spirits (vodka, whisky).

Chart, line chart

Description automatically generatedLast but not least, we needed to look at the reviews, both overall and particular.

Let’s explain the figure above. The x axis represents the buckets for overall reviews. The evaluations were continuos and we could not show every possible value, because that could have led to a chaos and therefor we created buckets as seen. So, we considered that the buckets shown above are the most accurate in what concerns the overall reviews given for beers from consumers. Must be mentioned that the reviews, overall or for taste, appearance, aroma and palate were given as they were, we did not calculate any mean in order to obtain them.

The y axis represents the percentages of reviews (number of reviews for a certain bucket over the whole number of reviews, multiplied by 100).

It is easy to remark that columns represent the buckets, meaning the overall review for a beer. The lines are reviews for taste, aroma, appearance and palate, the points being the percentage of reviews for a certain feature that is contained in the respective bucket.

As a conclusion of the figure, we can see that most reviews are between 3.75 and 4.5, summing up almost 60% of the reviews for beers, which denotes that the consumers are satisfied and would also recommend that beverages for other participants in this study. As in every other dataset, there are outliers, for example, very few, less than 5% would fully recommend the beers and less than 1% would never offer these drinks to someone else.

1. Model results

For this dataset we used 4 different types of algorithms for every task type, in our case classification and regression. We chose a variety of models that work on both tasks, described as follows:

1. SVM (Support Vector Machine)

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. Simply put, it does some extremely complex data transformations, then figures out how to separate your data based on the labels or outputs you've defined.

1. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set.

1. XGBoost

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. It’s vital to an understanding of XGBoost to first grasp the machine learning concepts and algorithms that XGBoost builds upon: supervised machine learning, decision trees, ensemble learning, and gradient boosting. Decision trees create a model that predicts the label by evaluating a tree of if-then-else true/false feature questions, and estimating the minimum number of questions needed to assess the probability of making a correct decision. Decision trees can be used for classification to predict a category, or regression to predict a continuous numeric value. A Gradient Boosting Decision Trees (GBDT) is a decision tree ensemble learning algorithm similar to random forest, for classification and regression. Ensemble learning algorithms combine multiple machine learning algorithms to obtain a better model.

1. Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine learning and are at the heart of deep learning algorithms. Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

These 4 types of models had the goal to predict the target value that we chose from our dataset and then after some hyperparameter optimization to see what model fit the best for our problem. The target we designated is the ABV (Alcohol content of beer (% by volume)) a float value from 0 to 100(technically never achieved in every beverage, in our case the maximum value is 58). As stated the regression problem is to predict ABV value, but for classification we had to create some classes based on our ABV column, thusly:

* The interval [0, 4) appointed as 0;
* The interval [4, 5) appointed as 1;
* The interval [5, 7.5) appointed as 2;
* The interval [7.5,100] appointed as 3.

4.1 Classification

Classification is a task that requires the use of machine learning algorithms that learn how to assign a class label to examples from the problem domain

1. SVM

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 0.5 | 0.24 | 0.32 | 17 | | 1 | 0.88 | 0.07 | 0.13 | 101 | | 2 | 0.66 | 0.91 | 0.77 | 326 | | 3 | 0.84 | 0.77 | 0.8 | 196 | | accuracy |  |  | 0.71 | 640 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 1 | 0.04 | 0.07 | 26 | | 1 | 0.8 | 0.08 | 0.15 | 98 | | 2 | 0.66 | 0.95 | 0.78 | 337 | | 3 | 0.89 | 0.73 | 0.8 | 179 | | accuracy |  |  | 0.72 | 640 | |

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
|  |  |

|  |
| --- |
| Top 3 hyperparameters combinations |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | params | split0\_test\_score | split1\_test\_score | split2\_test\_score | split3\_test\_score | split4\_test\_score | mean\_test\_score | | {'C': 1, 'gamma': 'auto', 'tol': 0.001} | 0.685547 | 0.691406 | 0.722114 | 0.716243 | 0.671233 | 0.697308 | | {'C': 1, 'gamma': 'auto', 'tol': 0.01} | 0.685547 | 0.691406 | 0.722114 | 0.716243 | 0.671233 | 0.697308 | | {'C': 1, 'gamma': 'auto', 'tol': 0.1} | 0.685547 | 0.693359 | 0.72407 | 0.714286 | 0.667319 | 0.696916 | |

1. Random Forest

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 0.86 | 0.29 | 0.43 | 21 | | 1 | 0.59 | 0.27 | 0.37 | 98 | | 2 | 0.71 | 0.85 | 0.77 | 338 | | 3 | 0.79 | 0.79 | 0.79 | 183 | | accuracy |  |  | 0.72 | 640 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 1 | 0.33 | 0.5 | 30 | | 1 | 0.49 | 0.18 | 0.27 | 104 | | 2 | 0.68 | 0.91 | 0.78 | 316 | | 3 | 0.9 | 0.82 | 0.86 | 190 | | accuracy |  |  | 0.74 | 640 | |

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
|  |  |

|  |
| --- |
| Top 3 hyperparameters combinations |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | params | split0\_test\_score | split1\_test\_score | split2\_test\_score | split3\_test\_score | split4\_test\_score | mean\_test\_score | | {'max\_depth': 50, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200} | 0.751953 | 0.707031 | 0.7182 | 0.712329 | 0.708415 | 0.719586 | | {'max\_depth': 90, 'max\_features': 'auto', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200} | 0.751953 | 0.707031 | 0.7182 | 0.712329 | 0.708415 | 0.719586 | | {'max\_depth': None, 'max\_features': 'auto', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200} | 0.751953 | 0.707031 | 0.7182 | 0.712329 | 0.708415 | 0.719586 | |

1. XGBoost

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 1 | 0.33 | 0.5 | 30 | | 1 | 0.48 | 0.3 | 0.37 | 99 | | 2 | 0.71 | 0.82 | 0.76 | 340 | | 3 | 0.78 | 0.8 | 0.79 | 171 | | accuracy |  |  | 0.71 | 640 | | |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 0.91 | 0.53 | 0.67 | 19 | | 1 | 0.47 | 0.34 | 0.4 | 106 | | 2 | 0.71 | 0.78 | 0.74 | 328 | | 3 | 0.82 | 0.86 | 0.84 | 187 | | accuracy |  |  | 0.72 | 640 | |

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
|  |  |

|  |
| --- |
| Top 3 hyperparameters combinations |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | params | split0\_test\_score | split1\_test\_score | split2\_test\_score | split3\_test\_score | split4\_test\_score | mean\_test\_score | | {'learning\_rate': 0.05, 'max\_depth': 7, 'n\_estimators': 100} | 0.697266 | 0.736328 | 0.712329 | 0.72407 | 0.747554 | 0.723509 | | {'learning\_rate': 0.05, 'max\_depth': 7, 'n\_estimators': 50} | 0.705078 | 0.730469 | 0.714286 | 0.722114 | 0.74364 | 0.723117 | | {'learning\_rate': 0.01, 'max\_depth': 7, 'n\_estimators': 200} | 0.705078 | 0.724609 | 0.716243 | 0.716243 | 0.747554 | 0.721945 | |

1. Neural Network

|  |
| --- |
| Confusion matrix |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | precision | recall | f1-score | support | | 0 | 0.39 | 0.32 | 0.35 | 22 | | 1 | 0.47 | 0.38 | 0.42 | 93 | | 2 | 0.69 | 0.78 | 0.73 | 337 | | 3 | 0.82 | 0.72 | 0.76 | 188 | | accuracy |  |  | 0.69 | 640 | |

|  |  |
| --- | --- |
| Confusion matrix heatmap | Neural Network loss plot |
|  |  |

|  |
| --- |
| Neural Network architecture: |
| Layer (type) Output Shape Param #  ===========================================================  dense\_3 (Dense) (None, 256) 3840  dense\_4 (Dense) (None, 256) 65792  dense\_5 (Dense) (None, 3197) 821629  ===========================================================  Total params: 891,261  Trainable params: 891,261  Non-trainable params: 0 |

4.2 Regression

Regression analysis is a fundamental concept in the field of machine learning. It falls under supervised learning wherein the algorithm is trained with both input features and output labels. It helps in establishing a relationship among the variables by estimating how one variable affects the other.

1. SVM

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
| MSE: 0.2000  MAE: 0.3028  R2: 0.5467  MAPE: 2.4137 | MSE: 0.1596  MAE: 0.2851  R2: 0.6691  MAPE: 3.5857 |

|  |
| --- |
| Top 5 hyperparameters combinations |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | params | split0\_test\_score | split1\_test\_score | split2\_test\_score | split3\_test\_score | split4\_test\_score | mean\_test\_score | | {'C': 1, 'epsilon': 0.1, 'gamma': 'auto', 'tol': 0.1} | 0.556326 | 0.746895 | 0.667088 | 0.648127 | 0.736148 | 0.670917 | | {'C': 1, 'epsilon': 0.1, 'gamma': 'auto', 'tol': 0.01} | 0.554045 | 0.747451 | 0.669248 | 0.646479 | 0.737008 | 0.670846 | | {'C': 1, 'epsilon': 0.1, 'gamma': 'auto', 'tol': 0.001} | 0.554011 | 0.747364 | 0.669257 | 0.646314 | 0.736881 | 0.670765 | | {'C': 1, 'epsilon': 0.01, 'gamma': 'auto', 'tol': 0.1} | 0.550512 | 0.744827 | 0.664819 | 0.645096 | 0.73627 | 0.668305 | | {'C': 1, 'epsilon': 0.01, 'gamma': 'auto', 'tol': 0.01} | 0.548639 | 0.744918 | 0.667664 | 0.644644 | 0.73558 | 0.668289 | |

1. Random Forest

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
| MSE: 0.1957  MAE: 0.3071  R2: 0.7165  MAPE: 3.3821 | MSE: 0.2010  MAE: 0.3035  R2: 0.6214  MAPE: 2.413 |

|  |
| --- |
| Top 5 hyperparameters combinations |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | params | split0\_test\_score | split1\_test\_score | split2\_test\_score | split3\_test\_score | split4\_test\_score | mean\_test\_score | | {'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 100} | 0.414021 | 0.745537 | 0.715983 | 0.697365 | 0.741463 | 0.662874 | | {'max\_depth': 90, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 200} | 0.411563 | 0.751239 | 0.713417 | 0.699177 | 0.738968 | 0.662873 | | {'max\_depth': None, 'max\_features': 'auto', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200} | 0.426652 | 0.735008 | 0.703108 | 0.704206 | 0.743696 | 0.662534 | | {'max\_depth': None, 'max\_features': 'auto', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100} | 0.425834 | 0.743482 | 0.702163 | 0.698172 | 0.740795 | 0.662089 | | {'max\_depth': None, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 5, 'n\_estimators': 200} | 0.409341 | 0.748068 | 0.717586 | 0.697251 | 0.737378 | 0.661925 | |

1. XGBoost

|  |  |
| --- | --- |
| Before hyperparameter tuning | After hyperparameter tuning |
| MSE: 0.2986  MAE: 0.3333  R2: 0.5092  MAPE: 2.0868 | MSE: 0.2447  MAE: 0.3270  R2: 0.6159  MAPE: 1.4418 |

|  |
| --- |
| Top 5 hyperparameters combinations |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | params | split0\_test\_score | split1\_test\_score | split2\_test\_score | split3\_test\_score | split4\_test\_score | mean\_test\_score | | {'learning\_rate': 0.05, 'max\_depth': 7, 'n\_estimators': 200} | 0.437769 | 0.650846 | 0.699482 | 0.761145 | 0.726452 | 0.655139 | | {'learning\_rate': 0.05, 'max\_depth': 7, 'n\_estimators': 100} | 0.432631 | 0.64354 | 0.703219 | 0.760949 | 0.731104 | 0.654288 | | {'learning\_rate': 0.05, 'max\_depth': 3, 'n\_estimators': 200} | 0.470787 | 0.636939 | 0.652548 | 0.756516 | 0.711082 | 0.645575 | | {'learning\_rate': 0.05, 'max\_depth': 7, 'n\_estimators': 50} | 0.411052 | 0.616562 | 0.704086 | 0.74168 | 0.718019 | 0.63828 | | {'learning\_rate': 0.05, 'max\_depth': 3, 'n\_estimators': 100} | 0.461738 | 0.613908 | 0.66708 | 0.743974 | 0.691531 | 0.635646 | |

1. Neural Network

|  |
| --- |
| Neural Network metrics result |
| MSE: 0.6219  MAE: 0.4105  R2: 0.6285  MAPE: 4.6892 |

|  |
| --- |
| Neural Network loss plot |
| |  | | --- | | Neural Network architecture: | | Layer (type) Output Shape Param #  =================================================================  dense (Dense) (None, 256) 3840  dense\_1 (Dense) (None, 256) 65792  dense\_2 (Dense) (None, 1) 257  =================================================================  Total params: 69,889  Trainable params: 69,889  Non-trainable params: 0 | |