

Jan Gniedziejko, Franek Gajda, Olga Granecka

BREWER'S FRIEND BEER RECIPES [[link](#)]

1. General description of the data set

This is a dataset of 75,000 homebrewed beers with over 176 different styles. Beer records are user-reported and are classified according to one of the 176 different styles. These recipes go into as much or as little detail as the user provided, but there's at least 5 useful columns where data was entered for each: Original Gravity, Final Gravity, ABV, IBU, and Color

2. Discussion of data mining goals and success criteria

2.1. Goals

1.1. Identifying the Genre/Subgenre of Beer Based on Attributes

Objective: Determine classification rules and common characteristics for beer within one genre.

Outcome: Successfully categorize beers into distinct genres/subgenres based on attributes.

1.2. Success Criteria:

Accuracy: At least 85% accuracy in correctly classifying beers into their respective genres/subgenres.

Precision and Recall: Precision and recall values for each genre/subgenre should be above 80%.

2.1. Does brewing method have an impact on color of the beer?

Objective: Determine whether and how the brewing method impacts the color of the beer.

Outcome: Establish a clear understanding of the relationship between brewing methods and beer colour.

2.2 Success Criteria:

Statistical Significance: p-value: The analysis should reveal a statistically significant relationship between brewing methods and beer color, with p-values less than 0.05 indicating significance.

Correlation Coefficient: Strength of Association: Calculate the correlation coefficient between specific brewing method variables and beer color. A correlation coefficient (Pearson or Spearman) greater than ± 0.5 would indicate a moderate to strong association.

3.1. Quality Control

Objective: Detect outliers in the ABV, IBU, or Color that might indicate issues with the brewing process or data recording.

Outcome: Ensure quality control by identifying and addressing potential issues.

3.2. Success Criteria:

Detection Accuracy: At least 95% accuracy in identifying outliers.

False Positive Rate: Less than 5% false positive rate.

Corrective Actions: Ability to propose corrective actions or checks for identified outliers.

3) Discussion of further steps

3.1. choosing the data mining task

1.1. Identifying the Genre/Subgenre of Beer Based on Attributes

Objective: Determine classification rules and common characteristics for beer within one genre.

Outcome: Successfully categorize beers into distinct genres/subgenres based on attributes.

1.2. Success Criteria:

Accuracy: At least 85% accuracy in correctly classifying beers into their respective genres/subgenres.

Precision and Recall: Precision and recall values for each genre/subgenre should be above 80%.

3.2. choosing the modeling algorithm

For classifiers, the most common modeling algorithm is a **decision tree**.

Key Characteristics:

- simple and easy to interpret.
- fast to train on small to medium datasets.
- prone to overfitting, especially with complex trees.
- high variance, low bias.
- can estimate the importance of features.

Advantages:

- easy to visualize and understand.
- requires little data preprocessing.
- handles both numerical and categorical data.

Disadvantages:

- overfitting: High variance if the tree is too complex.
- instability: Small changes in the data can lead to a completely different tree.

3.3. choosing the evaluation method

Keras models can be used to detect trends and make predictions, using the *model.predict()* class. This function enables us to predict the labels of the data values on the basis of the trained model.

4) Description of data preparation

4.1. Handling missing data

Missing data:

BoilGravity - 4% (2 990)
MashThickness - 40% (29 864)
PitchRate - 53% (39 252)
PrimaryTemp - 31% (22 662)
PrimingMethod - 91% (67 094)
PrimingAmount - 94% (69 087)
UserID - 68% (50 492)

Figure 4.1.1. Missing Data Matrix before handling missing data

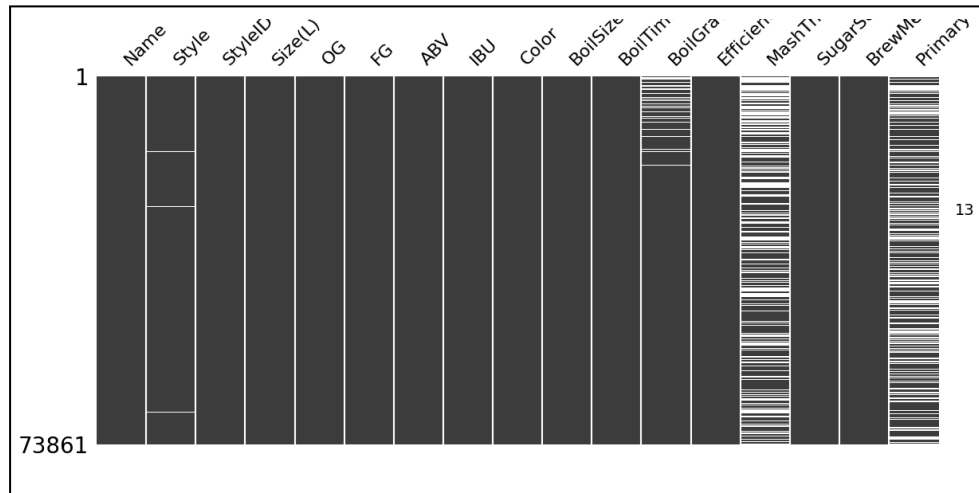
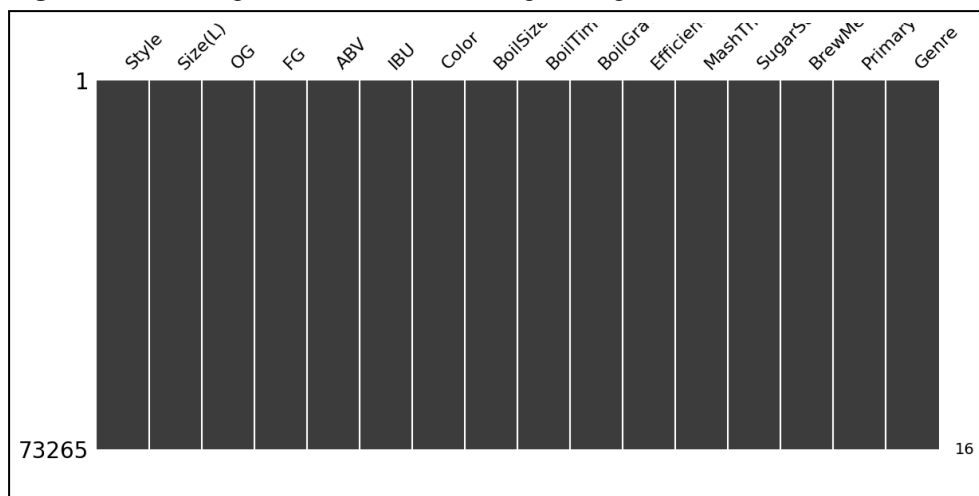


Figure 4.1.2. Missing Data Matrix after handling missing data



As it was established in the first part of our project (section 8.), the PrimingMethod and PrimingAmount will be removed due to the very high percentage of missing data. The userID won't be taken into account either, since this attribute has no impact on other attributes.

BoilGravity [4%] - due to the low percentage of missing data it was decided to replace the empty values with mean value.

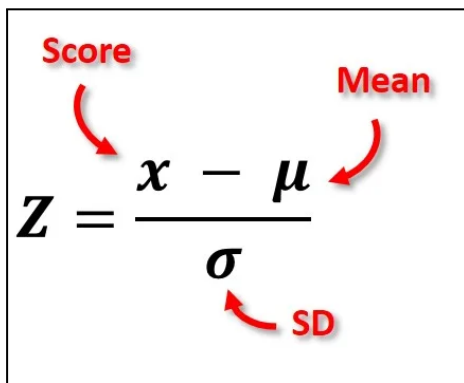
PitchRate [53%] - it was quite hard to decide whether to consider this attribute or not, but in the end we chose to keep it, and since it is a nominal data type, the empty values will be treated as a separate category. While building our model for beer genre

classification, we noticed that this attribute was causing issues, so we ultimately decided to omit it.

MashThickness [40%], PrimaryTemp [31%] - considering the relatively big percentage of missing data, replacing empty values with mean, or median may not be as effective as it was with BoilGravity. Hence, it was decided to handle missing data by using regression imputation.

4.2. Data normalization

The values' range of attributes varies from each other, therefore it was established to normalize every numerical data type that is in our dataset. To perform that, we used z-score standardization, with the following formula:



The diagram shows the Z-score formula:
$$Z = \frac{x - \mu}{\sigma}$$
 Red arrows point from the labels to the variables: 'Score' points to x , 'Mean' points to μ , and 'SD' (Standard Deviation) points to σ .

Figure 4.2.1. The Z-score formula in a population.
(image taken from www.simplypsychology.org)

4.3. Selecting a Subset of Data

The Brewer's friend beer recipes data contains 73 863 unique rows. We deleted around 600 rows, since some of the beer recipes were lacking 'Style' attribute (this is crucial for classification). This dataset is relatively small, so we chose not to implement any additional steps and decided to leave it as is.

4.4. Conversions of attributes' data types

Originally in the dataset from kaggle, the BoilGravity attribute is stored as nominal value, however we decided to treat it as a numerical value, since each value consists of digits.

Additionally, the nominal attributes "SugarScale," "BrewMethod," and "Genre" were converted into numerical values to be used in our model To do that, we used one_hot encoder (SugarScale) and label encoder (BrewMethod,Genre)

5) Model creation

The model was created with the use of Keras Python library. It is an easy to use and highly accurate high-level neural network library.

5.1. choosing work parameters for the algorithm

The default parameters were used:

criterion: Any = "gini",
splitter: Any = "best",
max_depth: Any = None,
min_samples_split: Any = 2,
min_samples_leaf: Any = 1,
min_weight_fraction_leaf: Any = 0.0,
max_features: Any = None,
random_state: Any = None,
max_leaf_nodes: Any = None,
min_impurity_decrease: Any = 0.0,
class_weight: Any = None,
ccp_alpha: Any = 0.0,
monotonic_cst: Any = None

5.1. executing the algorithm

Root node: StyleID

Table 5.1. Classification raport

Genre	Precision	Recall	Support
0	0.4	0.48	925
2	0.75	0.73	1976
3	0.34	0.35	694
5	0.42	0.44	1860

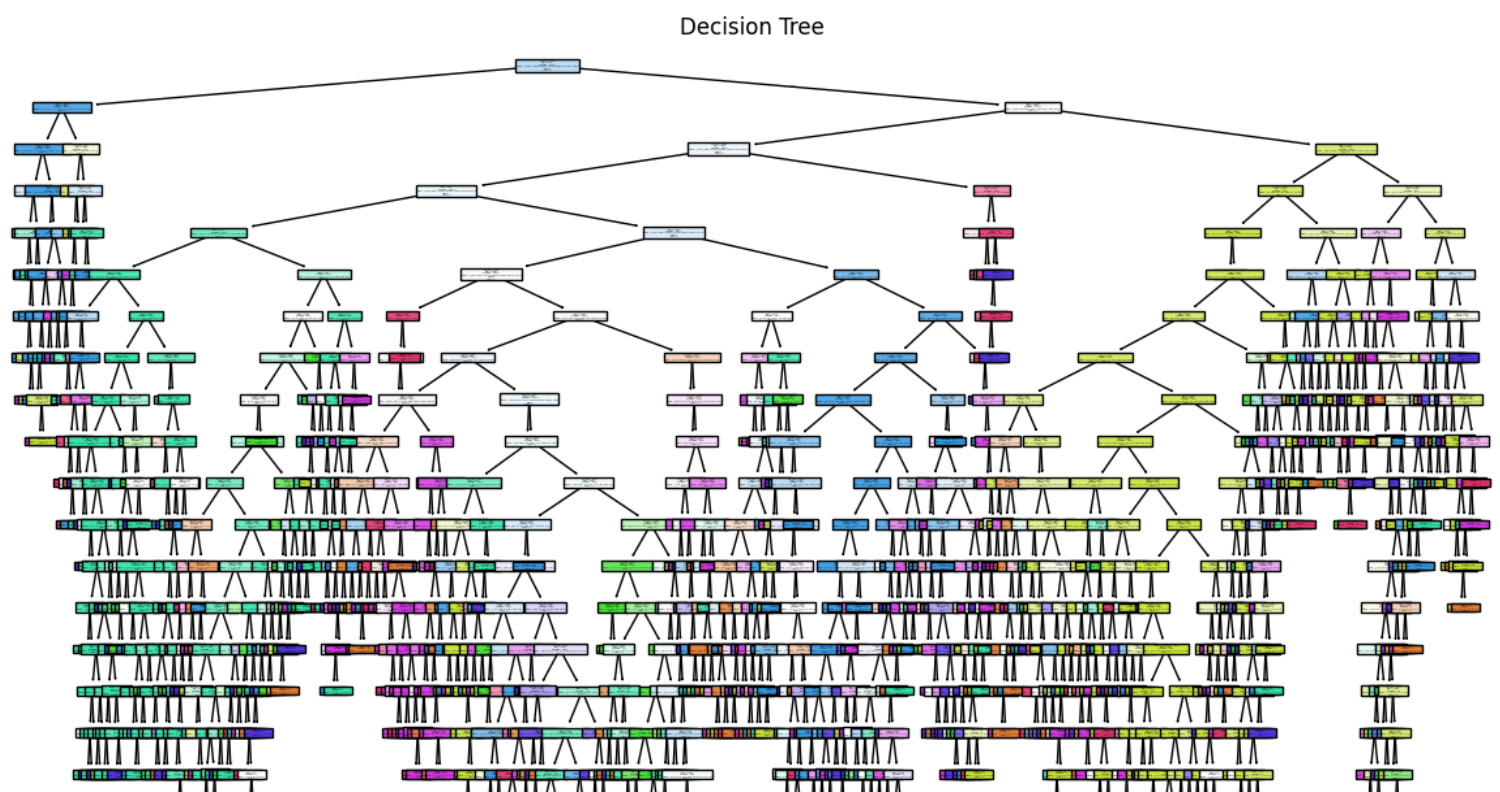
6	0.74	0.73	6264
7	0.15	0.16	649
8	0.41	0.41	1150
9	0.49	0.48	904

Decision Tree accuracy: 0.59

This accuracy is not very satisfactory. Almost the same results were obtained when the **Regression Decision Tree** algorithm was used.

Our dataset is rather complicated and with such dataset decision trees can be inaccurate because, as it was mentioned before, they are prone to overfitting. In section 6 other algorithm will be used and analyzed.

5.3. analyzing the resulting model itself (e.g. examining the shape of the decision tree)



6) Experiments with the model and the dataset

6.1. checking alternative algorithm

Random forest classifier

Key Characteristics:

- an ensemble of multiple decision trees
- slower compared to a single decision tree due to training multiple trees
- less prone to overfitting than individual decision trees
- lower variance, slightly higher bias compared to a single decision tree
- provides a more reliable estimate of feature importance

Advantages:

- combining multiple trees **reduces the risk of overfitting**
- generally more accurate than a single decision tree
- more robust to noise in the data.
- can handle missing data effectively.

Disadvantages:

- more complex and harder to interpret compared to a single decision tree.
- slower to train due to multiple trees.
- requires more computational resources.

Algorithm execution

Compared to the result of model training when the decision tree was used, random forest classifier usage resulted in **69% accuracy** of classification. Such result might be more probable, definitely the results were not overfitted, but it is not satisfactory having in mind our established goal criteria.

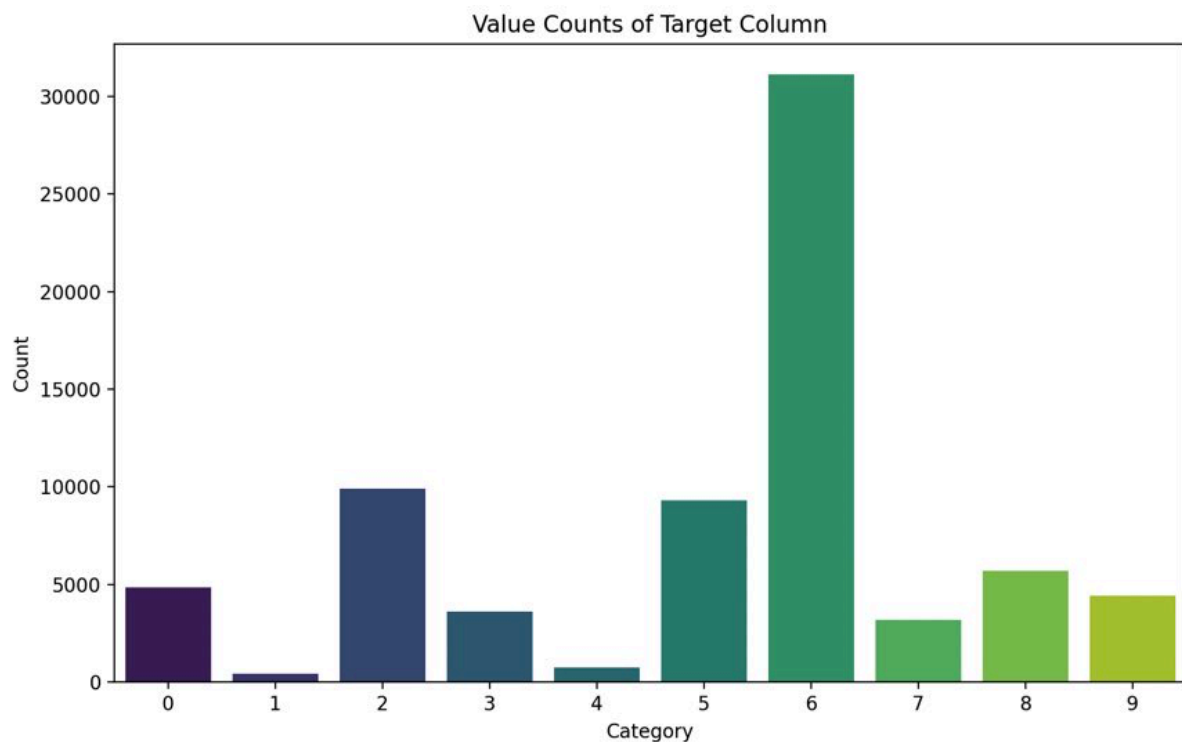
6.2. checking how experiments affect the created model and its evaluation

After analyzing the data, we concluded that 2 genres should be deleted due to the small number of beers that belonged to them - that might be the reason for smaller model accuracy.

Genre 1: Ciders and Meads number of samples: 420
Genre 4: Hybrid and Specialty number of samples: 739

The tuples that were of those genres were deleted.

Figure 6.2. Number of samples in each category.



Those actions - changing the algorithm and deleting two classes, and omitting “PitchRate” attribute - resulted in an improvement of accuracy to 71%.

Table 6.2. Classification report

Genre	Precision	Recall	Support
0	0.66	0.59	925
2	0.76	0.89	1976
3	0.71	0.35	694
5	0.66	0.46	1860
6	0.72	0.91	6264
7	0.50	0.07	649
8	0.66	0.48	1150
9	0.65	0.58	904

Random Forest accuracy: 0.7096103175703786

To resemble the success criteria of our goal:

Success Criteria:

Accuracy: At least 85% accuracy in correctly classifying beers into their respective genres/subgenres.

Precision and Recall: Precision and recall values for each genre/subgenre should be above 80%.

We can conclude that our goal was **NOT successfully reached**.

6.3 statistics about classification errors

With a use of confusion matrix, statistics about model's mistakes could be made.

Table 6.3.1 Confusion statistics

Genre number and name	Number of samples in test set	Most often confused with genre	Number of misclassified samples
(0) Belgian and French Styles	925	4	222 [24.0%]
(1) Dark Ales	1 976	4	123 [6.22%]
(2) German Styles	694	3	188 [27.09%]
(3) Lagers and Bocks	1 860	4	706 [37.96%]
(4) Pale Ales	6 264	3	156 [2.49%]
(5) Seasonal and Specialty	649	1	81[12.48%]
(6) Strong Ales	1 150	4	446 [38.78%]
(7) Wheat and Rye Beers	904	4	219 [24.23%]

62% of the misclassified samples were confused with genre no.4 (Pale Ales). It might be due to the fact that this class had the most training samples (31 130). The genre no. 6 was genre with the highest misclassification rate (38.78%) which might be connected to the fact that the classes 4 and 6 probably have similar features.

6.4 analyzing the resulting model

Figure 2. The shape of the random tree forest trees

Decision Tree from Random Forest

