



Department of Economics and Management
Institute of Economics (ECON)
Statistical Methods and Econometrics
Prof. Dr. Melanie Schienle

Seminar Report

Predictive Data Analytics - An Introduction to Machine Learning

Written by **Elisabeth Brockhaus**

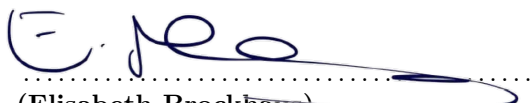
Matr. No. **2207143**
Econometrics

Supervised by
Dr. Sebastian Lerch
Nils Koster

September 30, 2022

I hereby confirm truthfully that I have authored this report independently and without the use of source material and aids other than those stated, that I have marked all passages literally or textually adapted from other sources as such and that I have respected the statutes of the Karlsruhe Institute of Technology (KIT) for ensuring good scientific practice.

Karlsruhe, September 30, 2022


.....
(Elisabeth Brockhaus)

Contents

Lists of Abbreviations and Symbols	VI
1 Introduction	1
2 Data	2
2.1 Pre-processing	3
2.2 Feature engineering	4
2.2.1 Most Memorable Characteristics	5
3 Benchmark Models	7
3.1 Simple Median Model	7
3.2 Baseline Regression Model	7
4 Machine Learning Model	8
4.1 Linear Models	8
4.2 Random Forest	9
4.3 Neural Networks	10
5 Results	12
6 Conclusion	13
Bibliography	14
A Appendix	15
A.1 Figures	15
A.2 Tables	19

List of Figures

1	Frequency of the realizations of the target variable.	2
2	Box plot of the rating depending on the review date.	4
3	Box plots assessing the influence of the company location. (a) Ratings for company locations with more than 25 observations in the training data. Locations with fewer observations are bundled under “other”. (b) Rating depending on whether company location and country of bean origin coincide or not.	5
4	Occurrence of each ingredient in the training data.	6
5	Histograms of the benchmark residuals.	7
6	Architecture of the pure flavor embedding neural network.	10
7	Architecture of the neural network with two input layers.	11
8	Metrics of all implemented and tuned models.	12
9	Rating with respect to the cocoa percent in the training data. Size of dots: Frequency of value pairs.	15
10	Rating with respect to the number of ingredients. On the right the number of ingredients are normalized and squared.	15
11	Shares of categories for features considered for one hot encoding and target encoding. Note that the U.S.A. bar in (b) clearly exceeds the y-limit.	16
12	Frequency of the compositions of the chocolate bars.	17
13	Correlation of the first 67 components from PCA of the embedded most memorable characteristics.	17
14	Rating depending on the label-encoded review date in the training data. Red line: Linear regression.	17
15	Non-zero feature importances in the initial random forest with flavor features. . . .	18
16	Non-zero feature importances in the random forest without flavor features.	18

List of Tables

1	Exemplary observations.	3
2	Number of unique categories and number of categories which occur only once. . . .	3
3	Metrics of the benchmark models.	7
4	Metrics of linear regression with different features on test data.	8
5	Metrics of advanced linear models.	9
6	Metrics of the random forest.	10
7	Best hyperparameters found for the neural network with two input layers.	11
8	Abbreviations used in the ingredients column.	19
9	Groups of features.	19

Lists of Abbreviations and Symbols

Abbreviations

Acc.	Accuracy
bl	baseline
MAE	Mean absolute error
MdAE	Median absolute error
ML	Machine learning
MSE	Mean squared error

Symbols

α	penalization parameter in LASSO
----------	---------------------------------

1 Introduction

There are countless different types of plain dark chocolate. These differ not only in the cocoa percentage, but also cocoa bean variety and processing affect the taste. Brelinski (2022) aims to value the taste of cocoa when processed into chocolate. Therefore, he provides ratings of over 2500 such chocolate bars. Besides the rating, descriptive characteristics for each are given in the database. Some refer to the company, such as the manufacturer and its location, others to the components, such as the bean origin, cocoa percentage and ingredients, and again others to the taste namely the most memorable characteristics.

The goal of this seminar paper is to build a machine learning model for the chocolate bar ratings. The large number of categorical predictors makes this task an exciting challenge. Many predictors have categories with only a few or even only one observation. When the dataset is split into training and test data, some categories occur only in the test data and would be unknown to a model fitted to the training data. Therefore, developing meaningful numerical features from the given data is essential and will be a major component of this report. Since encoding of categorical predictors often results in a large number of features, decision tree-based models could become helpful.

How good a chocolate tastes is primarily subjective. Therefore, expectations of machine learning models tend to be low. In particular, it makes a difference whether a high cocoa percentage is perceived as positive or negative. There is no information about who rated the chocolates and whether the ratings were given by one or more people. Thus, objective characteristics such as the cocoa percentage and the manufacturer might be overlaid by the subjective assessment of those. The explanatory value of such characteristics will have to be seen. Hopes therefore lie more in the list of ingredients and in the flavor descriptions. A high number of additives will probably have a negative influence on the rating. It is going to be a challenge to translate the flavor descriptions in a way that is understandable for the computer. If this can be accomplished, the resulting features should have a large impact on the performance of the model.

2 Data

The dataset was published by Flavors of Cacao¹. It consists of 2588 observations. The target variable is the rating of chocolate bars. It is numerical on a discrete scale from 1.0 to 4.0 in quarter steps. Values smaller than 2.5 are rare (Figure 1).

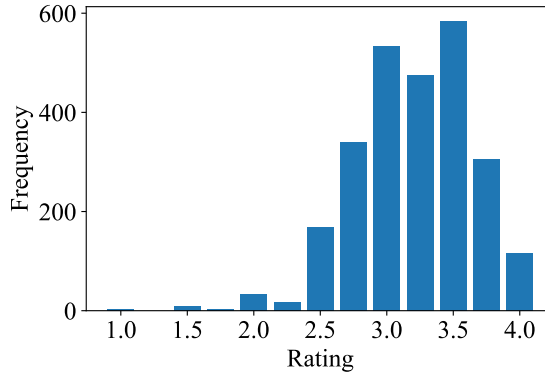


Figure 1: Frequency of the realizations of the target variable.

Additionally, there are a reference number and eight predictor variables (Table 1). Most predictors are categorical except for cocoa percent. The number of unique categories is large and many categories appear only once (Table 2). The reference number is not a unique identifier of the observations. It marks the time when an observation was added to the database. Higher numbers refer to more recent entries. It is not considered further below.

Even though the target is discrete, the problem is treated as a regression problem rather than a classification problem. The classes are on an ordinal scale, so misclassifying a rating of 3.25 as 3.0 is much better than misclassifying it as 1.0. This would not be captured by a classification model with accuracy as metric. As loss function the mean squared error (MSE) is used. Since low ratings are very rare, the performance of the models is also assessed via the median absolute error (MdAE) as a more robust metric. Finally, to account for the discrete nature of the target variable, also the accuracy is monitored. In order to calculate the accuracy, the predictions are rounded to the closest target value.

The split into training, validation and test data is done using *train_test_split* from *scikit-learn*. 15% of the data are used as test data, and 15% of the training data are taken for validation. If no validation data is needed, as in linear regression, the model is fitted to the combined training and validation data.

Because of the many categorical variables, a lot of pre-processing and feature engineering is required before building models. The following sections describe these. First, section 2.1 explains how missing values are handled and how data types are changed properly for regression. Afterwards, section 2.2 gives an overview of different encodings for the categorical features. A focus is on the column with the most memorable characteristics column (subsection 2.2.1).

¹http://flavorsofcacao.com/chocolate_database.html (accessed: 2022-06-26)

REF	Company Manufacturer	Company Location	Country of Bean Origin	Specific Bean Origin or Bar Name
2454	5150	U.S.A.	Tanzania	Kokoa Kamili, batch 1
2454	5150	U.S.A.	Madagascar	Bejofo Estate, batch 1
2458	5150	U.S.A.	Dominican Republic	Zorzal, batch 1
2542	5150	U.S.A.	Fiji	Matasawalevu, batch 1
2542	5150	U.S.A.	India	Anamalai, batch 1
2546	5150	U.S.A.	Venezuela	Sur del Lago, batch 1
2546	5150	U.S.A.	Uganda	Semuliki Forest, batch 1
797	A. Morin	France	Bolivia	Bolivia
797	A. Morin	France	Peru	Peru

REF	Cocoa Percent	Review Date	Ingredients	Most Memorable Characteristics	Rating
2454	76%	2019	3- B,S,C	rich cocoa, fatty, bready	3.25
2454	76%	2019	3- B,S,C	cocoa, blackberry, full body	3.75
2458	76%	2019	3- B,S,C	cocoa, vegetal, savory	3.5
2542	68%	2021	3- B,S,C	chewy, off, rubbery	3
2542	68%	2021	3- B,S,C	milk brownie, macadamia,chewy	3.5
2546	72%	2021	3- B,S,C	fatty, earthy, moss, nutty,chalky	3
2546	80%	2021	3- B,S,C	mildly bitter, basic cocoa, fatty	3.25
797	70%	2012	4- B,S,C,L	vegetal, nutty	3.5
797	63%	2012	4- B,S,C,L	fruity, melon, roasty	3.75

Table 1: Exemplary observations.

Predictor	Categories	occur once
Company (Manufacturer)	593	155
Company Location	65	5
Review Date	17	-
Country of Bean Origin	63	11
Specific Bean Origin or Bar Name	1643	1423
Ingredients	22	3
Most Memorable Characteristics	2545	2511

Table 2: Number of unique categories and number of categories which occur only once.

2.1 Pre-processing

The cocoa percent ranges between 42% and 100%. For most chocolate bars it lies between 70% and 80%. To be able to use the cocoa percent for ML models, it is converted into floats between 0.42 and 1. The correlation of this feature with the target in the training data is -0.1352. Several transformations are tested to obtain a higher correlated feature. Higher order polynomials perform best. Figure 9 (section A.1) shows that this is because small ratings occur mainly for 100% chocolates. These are further away from the 70% chocolates the higher the exponent becomes. Possibly increased weighting of high percentage chocolate improves some ML models. Thus, a column *cocoa.percent*⁴ is added. Its correlation with the rating in the training data is -0.1776.

The dataset is almost complete. Only 87 observations have no ingredients indicated. The ingredients are given as a string consisting of a number and a list of abbreviations. The number is the number of ingredients in the chocolate bar. The abbreviations stand for specific ingredients.

The meaning of the latter is given in Table 8 (section A.2). For observations with cocoa percent = 100%, it can be argued that there are only beans in the chocolate bar. Thus, these 14 missing ingredients are set to “1- B”. Then the number of ingredients is separated into a new column. Missing number of ingredients are replaced by the median. The median is calculated in the training data over observations with cocoa percent < 100%. The remaining 73 missing values in the list of ingredients are not filled. They will be handled within the encoding (section 2.2). Removing these observations from the dataset is not considered because the dataset is small and all other variables are known. The number of ingredients is almost uncorrelated with the rating. In the training data the correlation is -0.0565. Some transformations are considered aiming for a stronger linear dependence. Figure 10 (section A.1) shows the rating depending on the number of ingredients and the square of the normalized number of ingredients. The latter has a correlation with the target of -0.1400.

2.2 Feature engineering

The review date column is integer. Nevertheless, treating its values as numerical is not reasonable as they lie between 2006 and 2022. Figure 2 shows a box plot of the rating with respect to the review date in the training data. Ratings below 2 occur in the first years only. In later years the mean rating tends to be higher. Together a label encoding seems reasonable, i.e., the minimal review date is subtracted to obtain label-encoded review dates between 0 and 16. The correlation between encoded review date and rating is 0.1143. A linear fit is shown in Figure 14 (section A.1). The variance of the rating decreases with increasing review date. A White test also indicates heteroskedasticity with a p-value of the LM-test of 2.1306×10^{-22} . Squaring the encoded review date reduces heteroskedasticity, but simultaneously reduces correlation with the rating to 0.0231. A log-transformation of the rating slightly increases the correlation with the encoded review date. But it does not reduce heteroskedasticity significantly. So, no scaled version of the review date is considered in the following.

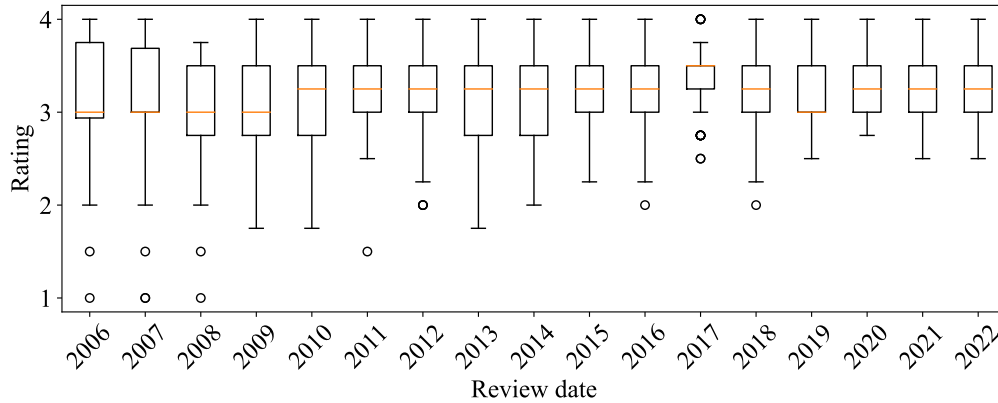


Figure 2: Box plot of the rating depending on the review date.

For features with many categories, two types of encoding are considered: One hot encoding and target encoding. This applies to the three predictors company manufacturer, company location and country of bean origin. The specific bean origin or bar name variable is not used, as almost every category occurs only once. For the most memorable characteristics column, the situation is the same. However, the strings might have some explanatory value. This is examined in subsection 2.2.1.

Company location consists of 65 extremely unbalanced categories (section A.1, Figure 11b). 12 categories make up over 80% of the observations. Over 45% of the chocolate bars are from companies located in the U.S.A. followed by around 7% in France and Canada. The influence

of the company location on the rating is shown in Figure 3a. For the country of bean origin the imbalance is not as large but still considerable (section A.1, Figure 11c). All company manufacturer categories are quite small. Even the largest one comprises hardly more than 2% of the observations (section A.1, Figure 11a). Overall, one hot encoding of the three predictors would result in an extremely large number of features and a sparse matrix. Therefore, categories with less than 25 or 50 observations, respectively, are grouped into the category “other”. Target encoding can be performed without this aggregation step. The built-in smoothing of the *scikit-learn TargetEncoder* is used for regularization. This reduces overfitting in categories with few observations.

For 13% of the observations, company location and country of bean origin coincide. Whether this has any impact on the rating is examined in Figure 3b. Although the variance of the rating might depend on it, the effect seems to be negligible on average.

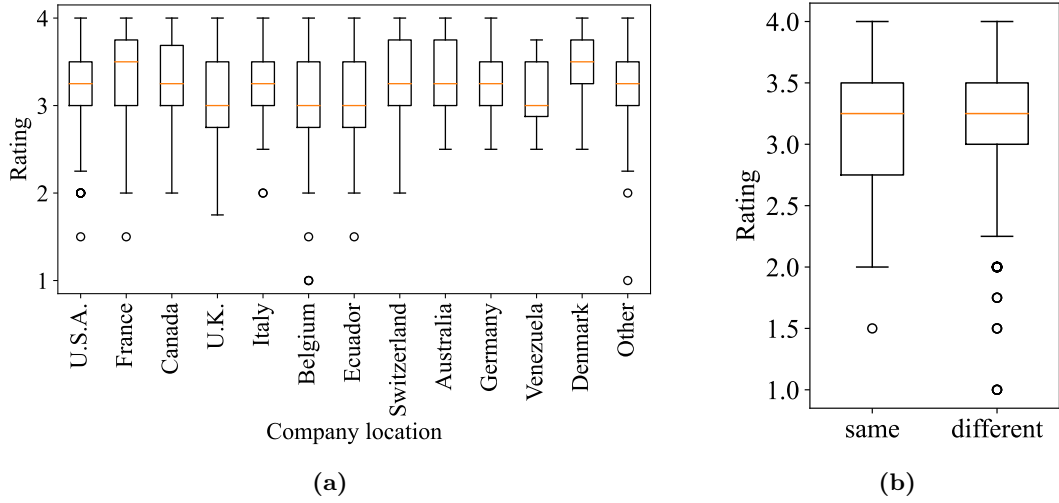


Figure 3: Box plots assessing the influence of the company location. **(a)** Ratings for company locations with more than 25 observations in the training data. Locations with fewer observations are bundled under “other”. **(b)** Rating depending on whether company location and country of bean origin coincide or not.

The number of ingredients is already extracted from the original ingredients column (section 2.1). From the remaining list of abbreviated ingredients dummy variables are created for each ingredient except beans. A dummy variable for beans would take the value 1 for all observations. The remaining missing ingredient values now have all dummies equal to zero. Instead, they could have been replaced before by the mode, which is “B, S, C” (section A.1 Figure 12). Figure 4 gives another reason to do so. It shows the proportion of chocolate bars in the training data containing each ingredient. Sugar is in over 93% of the chocolates, cocoa butter in approximately 2/3 of the chocolates. All other ingredients occur in only 20% or less of the observations. So missing values are eventually replaced by “B, S, C”.

2.2.1 Most Memorable Characteristics

The most memorable characteristics column needs special attention. It contains lists of adjectives and other descriptions of the chocolate’s flavor. For humans these of all columns would be most important to assess the quality of the chocolate and thereby estimate the rating. But a computer does not understand which words are positive, negative, or neutral in the context of chocolate.

The first approach to use such information for an ML model aims for some encoding. Since the descriptions are non-repetitive in the dataset, the lists are split into single words. But still most words occur only once or twice and no word occurs more than five times. There are some misspellings such as “sticy” instead of “sticky” and “vanila” instead of “vanilla”. Correcting them

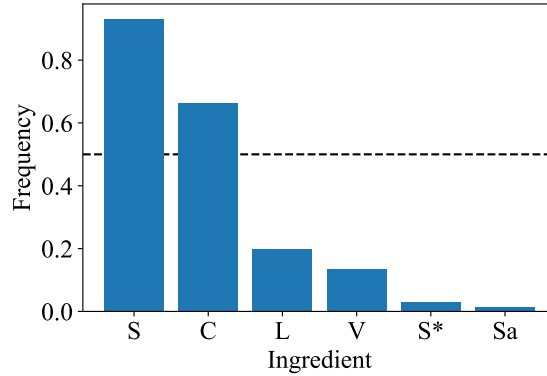


Figure 4: Occurrence of each ingredient in the training data.

properly could increase the number of word repetitions. For this purpose, several pre-implemented classes are tried (*Speller* from the package *autocorrect*, *SpellChecker* from *spellchecker*, and *Word* from *textblob*). All of them not only find misspellings, but also change originally correct words. E.g., “papaya” is replaced by “papal” and “macadamia” by “acadamia”. In addition, many words are unknown to the packages. Overall, this approach is not pursued any further. Similar to this idea, word stemming is applied. Some words occur in different versions, e.g., “nut”, “nuts” and “nutty”, or “chocolate”, “choco”, “chocolatey” and “chocolaty”. For taste, these differences do not matter, and the words should be placed in the same category. The *nlTK* package provides several classes for such applications. The *SnowballStemmer* does not work. It replaces most of the endings with “i” or simply cuts them off. The *WordNetLemmatizer* seems better suited. It replaces all plural forms with the corresponding singular form. However, it is not able to turn all versions of “chocolate” into the same word. In general, it does not change “y” endings of the words. So this approach is also discarded.

All approaches that try to find useful categories from the most memorable characteristics column seem ineffective. Next, the text itself should be interpreted and converted into numerical features. For such applications, there are many pre-trained text embedding models on TensorFlow Hub². These models map strings into a high dimensional numerical space. In this space, more similar descriptions are closer together. Such models can either be used to construct numerical features from text which then serve as input to ML models, or they can be embedded as a *KerasLayer* in some neural network (chapter 4). Here, a model trained on the English Wikipedia corpus is used³. It is a normalized model which means that the punctuation is automatically removed. Thus, the raw feature can be used in the text embedding without further pre-processing. The embedding returns a 250-dimensional vector representing each entry of the most memorable characteristics. The number of features is then reduced with a principal component analysis. 50% of the empirical variance of the 250-dimensional data matrix can be explained using 6 features. 22 features explain more than 80% of the variance, and 67 features more than 95%, respectively. The first features have a correlation with the rating of up to about 0.2 in absolute values (section A.1 Figure 13). The correlation decreases for features that explain less variance. Individual peaks in correlation may be coincidental and need to be handled with care.

²<https://tfhub.dev/>

³<https://tfhub.dev/google/Wiki-words-250-with-normalization/2>

3 Benchmark Models

This chapter describes the two benchmark models. At first, the median rating is considered. Second, a linear regression with three simple features is performed.

3.1 Simple Median Model

The simplest model is to use the mean or median value of the target variable in the training data as a prediction. The target variable in this case only takes discrete values. Therefore, the median is chosen as benchmark.

Resulting metrics are displayed in Table 3a. A histogram of residuals is shown in Figure 5a.

3.2 Baseline Regression Model

For a first linear regression three features are used: cocoa percent, review date (label-encoded) and number of ingredients. Cocoa percent and number of ingredients are chosen because they are numerical. Thus, they can be used for linear regression directly. Review date is integer but has to be treated as categorical (section 2.2). Nevertheless, label-encoding this predictor is straight forward. Thus, it is considered for the benchmark, too.

Table 3b and Figure 5b summarize the performance of the baseline regression model. Many predictions are off by up to 0.75 points.

Metric	Train	Test
MSE	0.1967	0.1992
MdAE	0.2500	0.2500
Acc.	0.1825	0.1825

(a) Simple median model.

Metric	Train	Test
MSE	0.1879	0.1876
MdAE	0.2911	0.2929
Acc.	0.2006	0.2134

(b) Baseline regression model.

Table 3: Metrics of the benchmark models.

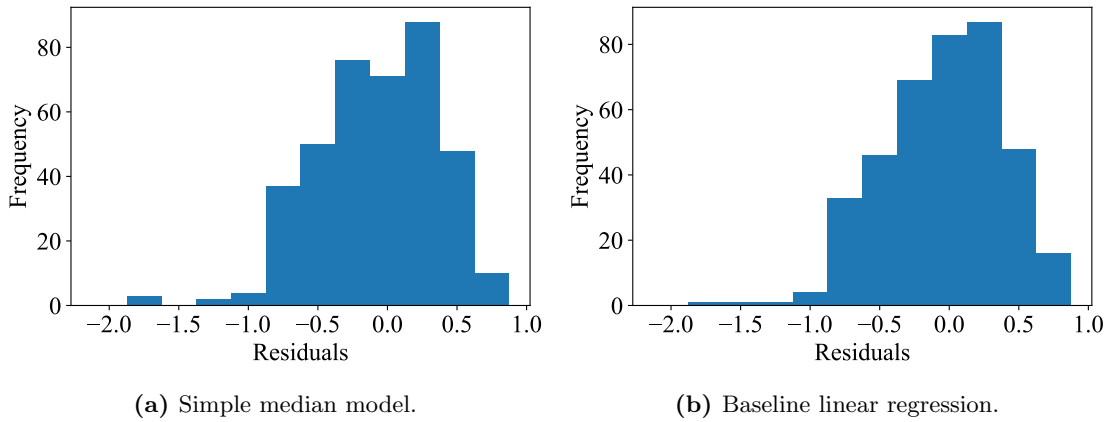


Figure 5: Histograms of the benchmark residuals.

4 Machine Learning Model

Before implementing less explainable ML models, linear models are used to assess the quality of the features developed above (section 4.1). First, the individual groups of features are used in linear regressions. Then backward elimination is used to find important features. Finally, LASSO is performed with all features. In section 4.2 random forests with and without the embedded most memorable characteristics are presented. A focus is on the features importance and the hyperparameter tuning. Next, section 4.3 describes neural networks using the pre-trained text embedding model as in subsection 2.2.1. The first neural network is based on this feature only. Afterwards, a second input layer for the other features is added.

4.1 Linear Models

As opposed to random forests and neural networks, linear models are white-box models. This means that they not only provide estimates for the target variable, but also allow for the interpretation of the predictors. For this reason, they are used not only for a benchmark model, but also to assess the quality of the features constructed in section 2.2. Starting from the baseline linear model (section 3.2), the groups of features (section A.2 Table 9) are added to the model one at a time. The resulting metrics are displayed in Table 4. The one hot encoded features have no or even a negative impact on the metrics, while the target encoding results in a decrease in MSE from 0.1876 to 0.1699 and an increase in the accuracy of more than three percentage points. The dummies for each ingredient have a small positive impact on the metrics, except for the accuracy, which is almost the same as for the baseline model. It does not matter whether the number of ingredients is scaled or not. The largest improvement of the baseline linear model is achieved with the embedded flavors from subsection 2.2.1. Here, the 22 features that explain 80% of the variance are included. This linear model reaches an MSE of 0.1320 and an accuracy of 0.2776. The best model so far uses all these features together. Its MSE is 0.1257 and its accuracy is 0.3316.

Group of features	MSE	MdAE	Accuracy
baseline	0.1876	0.2929	0.2134
bl & one hot encoded features	0.1832	0.2939	0.2057
bl & target encoded features	0.1699	0.2922	0.2365
bl & ingredient dummies	0.1759	0.2751	0.2108
bl (number ingredients scaled) & ingredient dummies	0.1752	0.2761	0.2134
bl & flavors embedded	0.1320	0.2285	0.2776
all	0.1257	0.2138	0.3316

Table 4: Metrics of linear regression with different features on test data.

Next, backward elimination is performed. This method does not identify the most important features in terms of causality. Instead, insignificant features are successively removed until all remaining features are significant. This does not improve the resulting metric because no information is added to the model. It does, however, ensure that the final model does not contain highly correlated features. In backward elimination the cocoa percent column and most of the one hot encoded features are removed. The target encoded features are retained except for the company location. Even though the ingredient dummies only slightly improve the model (Table 4), none of

them is removed. The selection of features from the embedded flavor columns is consistent with their correlation with the rating (section A.1 Figure 13).

Instead of removing features from the design matrix, LASSO can be applied which penalizes coefficients that are not zero. Unlike ridge regression, the coefficient vector estimated with LASSO contains zero entries. This leads to automatic feature selection (Tibshirani (1996)). However, it should be used with care because the selection is not based on causality. It can be proven that there is always a penalization parameter α for which LASSO has a smaller MSE than linear regression. LASSO is fitted to the training data for different values of α . The resulting MSEs on the validation data are compared. $\alpha = 0.00025$ is optimal and selects similar features as backward elimination.

Metrics after backward elimination and from the best LASSO model are given in Table 5. Both are about as good as linear regression with all features.

Metric	Train	Test
MSE	0.1004	0.1261
MdAE	0.2002	0.2137
Acc.	0.3162	0.3111

(a) With backward elimination.

Metric	Train	Test
MSE	0.0989	0.1283
MdAE	0.1998	0.2241
Acc.	0.3194	0.2905

(b) LASSO with $\alpha = 0.00025$.

Table 5: Metrics of advanced linear models.

4.2 Random Forest

The linear dependence of the features on the target is rather low. Nonlinear dependencies can be captured, for example, with random forests. Therefore, this is the first advanced machine learning model to be implemented. Feature selection has a low priority in random forests. In the splits, features with higher explanatory value are considered more often, and unimportant features consequently receive automatically little attention and have small influence on the predictions. The effect on the computation time when using all features instead of a selection of important features is negligible when training a single random forest in the given application. When tuning the hyperparameters, unimportant features are removed to speed up the process as much as possible.

First, a random forest with non-tuned hyperparameters is trained on all features (95% embedded flavors). Hyperparameters are set to their default values except the minimal number of samples per leaf which is set to 100. This leads to overfitting. The metrics for the test data are intermediate between the benchmark models and the linear regression models from the previous section (Table 6a).

To investigate whether all features are used by the random forest, the permutation feature importance is examined. The results are shown in Figure 15 (section A.1). Features with zero importance are hidden. The model relies mainly on the features representing the flavor. Other than that, only the encoded review date and the target encoded versions of the company manufacturer and the country of bean origin are considered. Therefore, in order not to unnecessarily lengthen the training duration in the following, the one hot encoded features, cocoa percent and the ingredient features are excluded from the model.

To improve the performance, hyperparameters are tuned with Bayesian optimization (Nogueira (2014)). Stratified k-fold cross validation with $k = 3$ is used to obtain the validation metric for each parameter combination. After 100 iterations, the parameter bounds are adjusted with respect to the current best parameters. The final parameters are obtained via grid search on the smaller bounds. Metrics before and after hyperparameter tuning are given in Table 6.

Even after adjusting the hyperparameters, the random forest performs worse than the linear models from the previous section. It still tends to overfit and the MSE of the test data does not fall below 0.15. So, neural networks are considered as an alternative ML method in the

Metric	Train	Test
MSE	0.1183	0.1611
MdAE	0.2185	0.2731
Acc.	0.2970	0.2365

(a) Before tuning.

Metric	Train	Test
MSE	0.0970	0.1511
MdAE	0.1901	0.2544
Acc.	0.3414	0.2468

(b) After tuning.

Table 6: Metrics of the random forest.

following. However, before implementing an even less interpretable model, the random forest is used to investigate the importance of the non-flavor features. Therefore, it is fitted to all columns except those created from the most memorable characteristics. As expected, the performance of this random forest is worse. But now all three target encoded features are relevant (section A.1 Figure 16). Together with the label encoded review date and the cocoa percent, they are the most important features. In addition, the number of ingredients, the dummy for whether the company is located in the U.S.A., and the cocoa butter dummy gain some importance. These findings are in the following used to select the features for the neural network.

4.3 Neural Networks

Equivalent to subsection 2.2.1, the pre-trained text embedding model is now incorporated into neural networks. As a starting point, the most memorable characteristics are used as the only predictor. This network is based on the one from The TensorFlow Hub Authors (2019). It consists of the pre-trained model as a non-trainable *KerasLayer*, a dense layer with 32 units and *ReLU* activation function, and a dense layer with a single neuron as the output layer (Figure 6). *Adam* is used as optimizer, the loss is the MSE and the metric is the MAE. It is trained for 200 epochs, but early stopping with patience = 5 is added to avoid overfitting. Without any tuning, it already achieves an MSE of 0.1121 on the test data in 121 epochs.

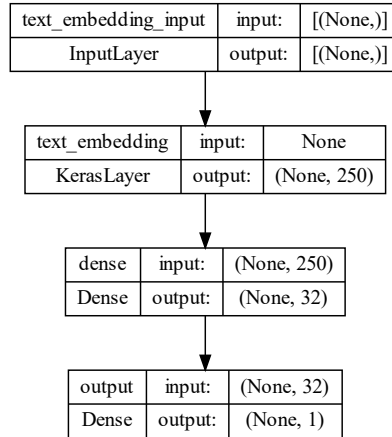


Figure 6: Architecture of the pure flavor embedding neural network.

Next, other features are added to the neural network. For this purpose, a second input layer is introduced. The architecture is shown in Figure 7. Regularization is increased with a dropout layer. Otherwise, the network consists mainly of dense layers. Optimizer and loss function are the

same as before. Based on the results of the previous sections, the following features are included in the model: baseline, one hot encoded company manufacturer, target encoded, ingredient dummies (section A.2 Table 9). This model achieves an MSE of 0.1084 on the test data.

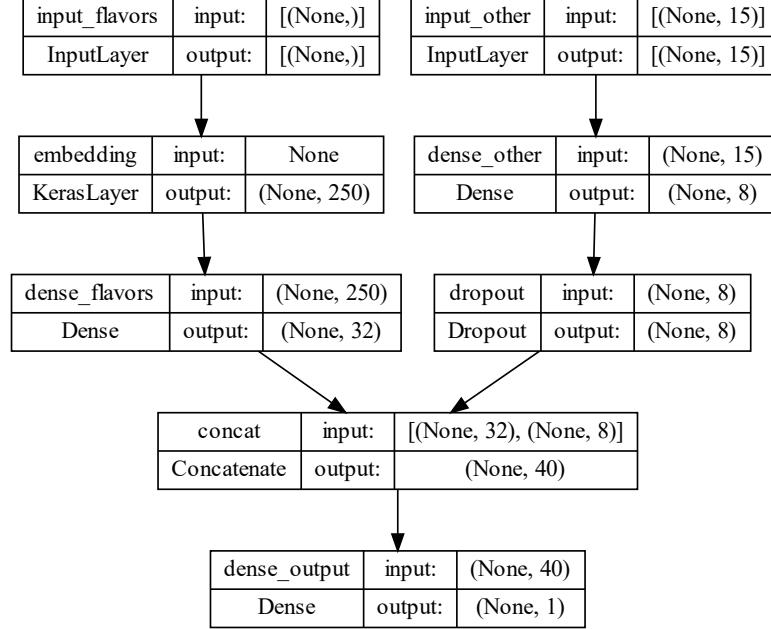


Figure 7: Architecture of the neural network with two input layers.

Next, its hyperparameters are tuned using *BayesianOptimization Tuner* from Keras. Here, the pre-trained text embedding model used above is compared to another model from TensorFlow Hub¹. This has a smaller output shape, was trained on Google News, and is also normalized. In addition, the units of the dense layers before concatenation are optimized and the learning rate of the optimizer is chosen between 0.001, 0.0005 and 0.0001.

As a result, the previously used text embedding model (*Wiki-words-250-with-normalization*) is found to outperform the alternative (*nnlm-en-dim50-with-normalization*). It is used in all the 10 best trials. The best hyperparameters are given in Table 7. The resulting model achieves an MSE of 0.1095 on the test data, which is worse than before the tuning.

Hyperparameter	Best Value
Text embedding model	Wiki-words-250-with-normalization
Units dense layer (flavors)	64
Units dense layer (other)	10
Dropout rate	0.1
Learning rate	0.001

Table 7: Best hyperparameters found for the neural network with two input layers.

¹<https://tfhub.dev/google/nnlm-en-dim50-with-normalization/2>

5 Results

Most of the results have already been used as a line of reasoning in the previous chapter. Beyond that, this chapter only gives an overview of the performance of all implemented models. The metrics on the test data are compared in Figure 8. The best metric achieved is highlighted by a red line. All models outperform the two benchmark models on almost every metric. Only the random forest without flavors has a higher MdAE than the simple median model. The neural network using only the most memorable characteristics has the best overall metrics. The linear regression outperforms both random forests.

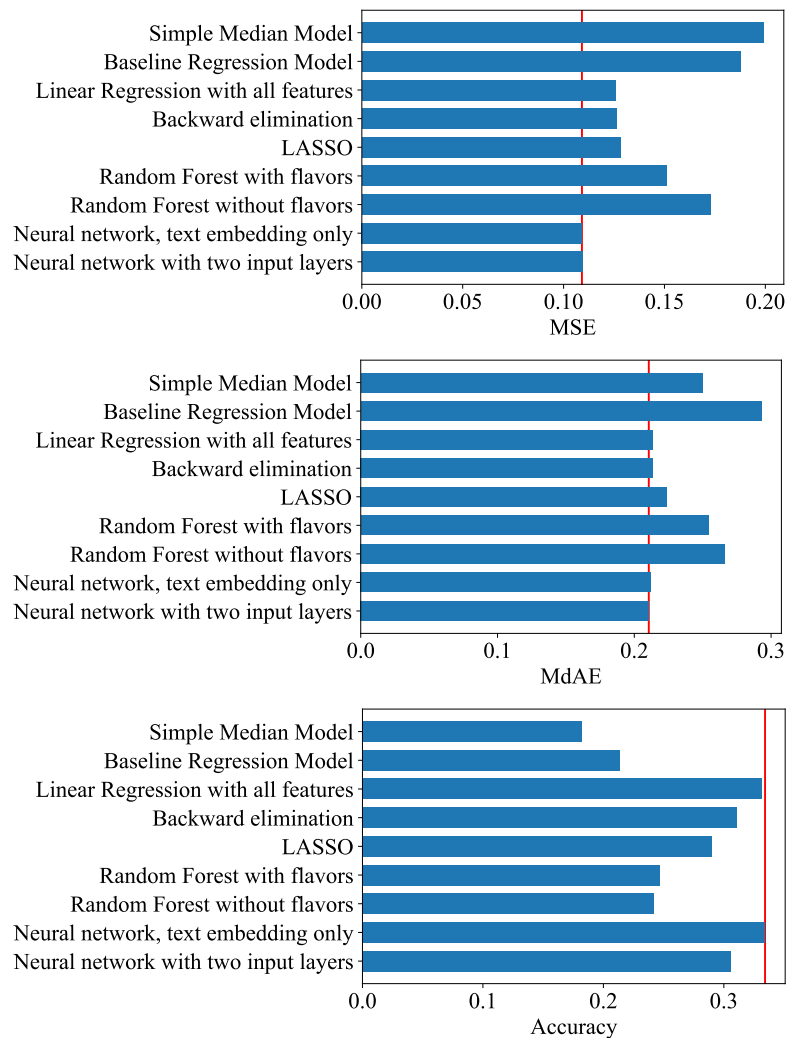


Figure 8: Metrics of all implemented and tuned models.

6 Conclusion

The best results were obtained with a neural network. It offered the best possibility to include the most memorable characteristics. By using pre-trained text embedding models, the computational costs were kept within acceptable limits. Overall, the most important predictor is definitely the most memorable characteristics. The text embedding allowed its inclusion in ML models. All models with this information significantly outperformed models based only on the others predictors. The neural network trained on no predictor other than the most memorable characteristics was even the best model.

Nevertheless, some explanatory value was found for the other predictors. For example, a simple linear model with ingredient dummies or target-encoded categorical features was better than both of the benchmark models. Still the dependence of the rating on the other features is low. The numerical features were more or less irrelevant. This is already evident from the fact that both benchmark models have similar metrics, although the basic linear regression model includes all numerical features and the simple median model has no information except the target on the training data.

For the categorical features, the challenge was to find a good way to encode them. The large number of categories made it necessary to aggregate most of them before constructing dummy variables. As a result, most of the information was lost, and one hot encoded features were excluded by most feature selection procedures. Unlike expected, the dummy variables of the ingredients were also of little importance. Text embedding for features such as the company manufacturer is not considered because the text itself has no actual meaning that could influence the rating. As a possible additional feature, the geographical distance between bean origins could have been introduced. Also, the incorporating of the embedded most memorable characteristics could have been optimized. For example, the predictions of the neural network using only these could have been used as input to the other models. These types of ensembles can be very powerful.

Contrary to expectations, the random forest was not a suitable choice for the dataset. Even after tuning the hyperparameters, its performance was only slightly better than the benchmark. In particular, overfitting was a major issue for the random forest in this dataset.

What needs to be improved is the tuning of the hyperparameters. It has very little effect on the metrics. Some models were even worse after tuning. One factor that made tuning the hyperparameters difficult was the low available computing capacity. If this had been higher, it would also have been an option to further train the parameters of the pre-trained text embedding model.

In summary, machine learning models were not only appropriate for this dataset, but even necessary. Without neural networks, the most memorable characteristics would have had no value to the models. However, after finding numerical features from this predictor, linear models perform the task almost as well as the implemented ML models or, looking at the random forest, even better.

Bibliography

Brelinski, B. (2022), ‘Flavors of Cacao: Chocolate bar ratings database’. Accessed: 2022-06-26.

URL: http://flavorsofcacao.com/chocolate_database.html

Nogueira, F. (2014), ‘Bayesian Optimization: Open source constrained global optimization tool for Python’. Accessed: 2022-09-28.

URL: <https://github.com/fmfn/BayesianOptimization>

The TensorFlow Hub Authors (2019), ‘Basic Text Classification: Text classification with movie reviews’. Accessed: 2022-09-27.

URL: https://colab.research.google.com/github/tensorflow/hub/blob/master/examples/colab/tf2_text_classification.ipynb

Tibshirani, R. (1996), ‘Regression shrinkage and selection via the lasso’, *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288.

A Appendix

A.1 Figures

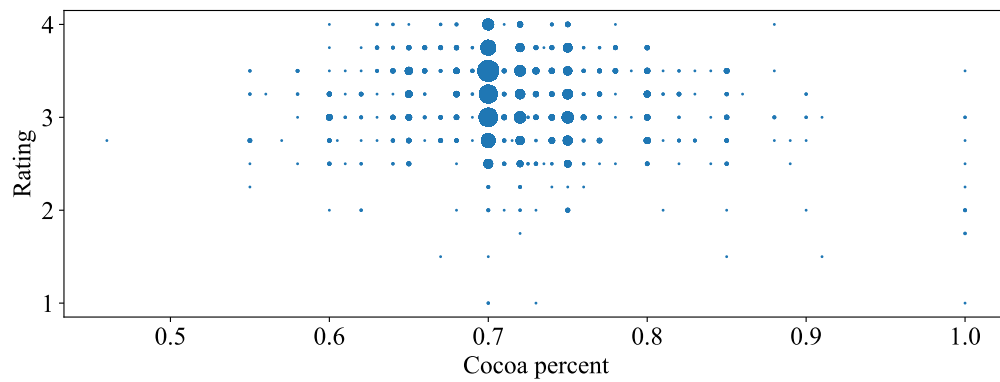


Figure 9: Rating with respect to the cocoa percent in the training data. Size of dots: Frequency of value pairs.

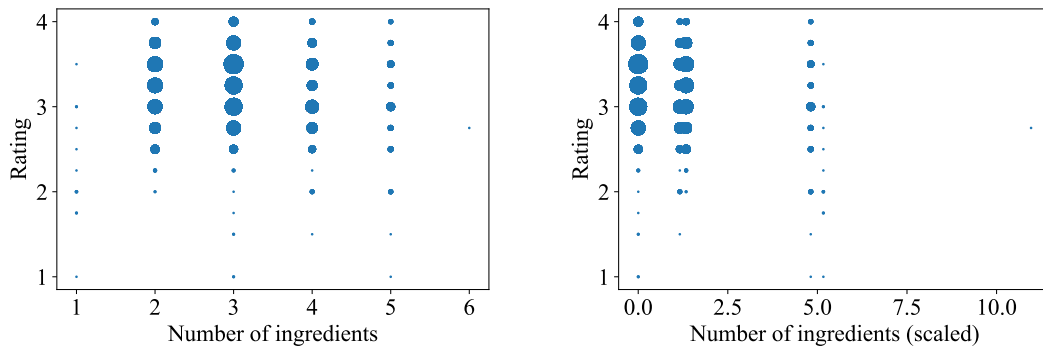
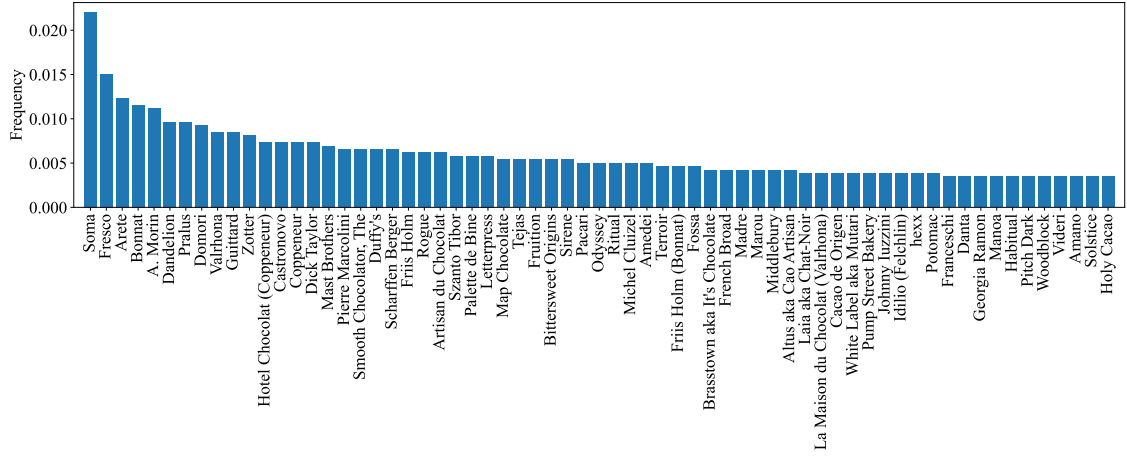
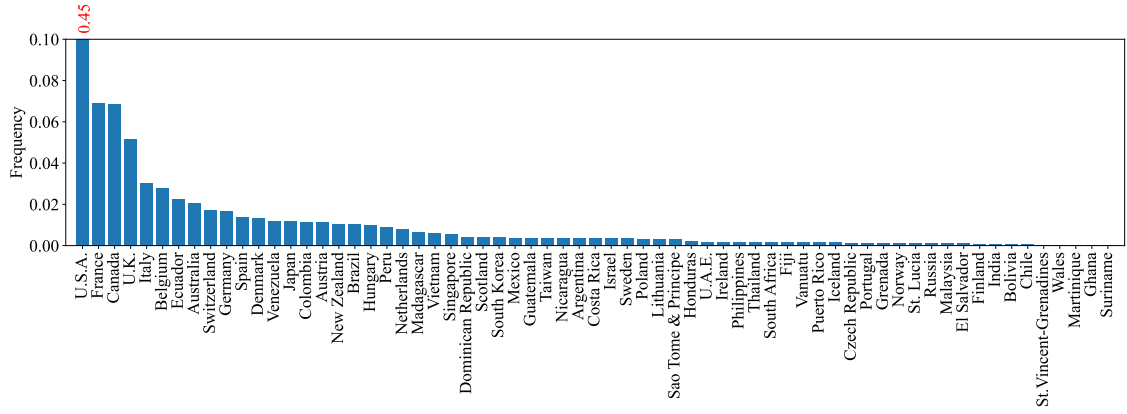


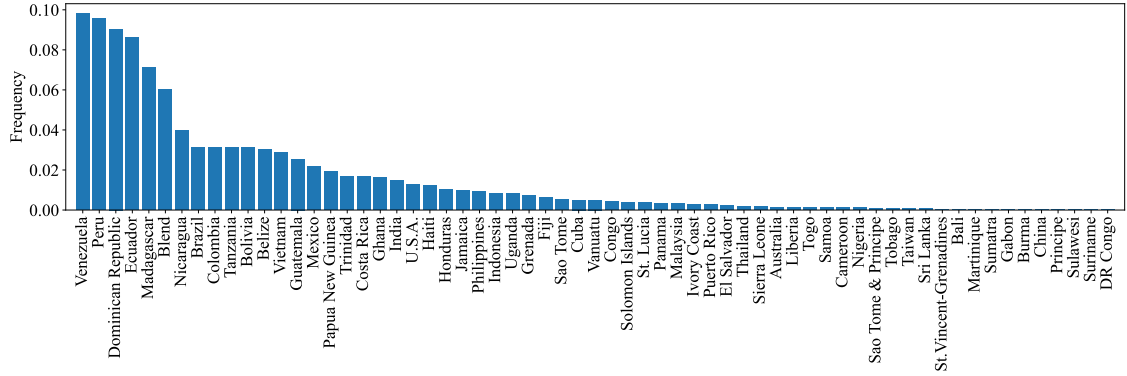
Figure 10: Rating with respect to the number of ingredients. On the right the number of ingredients are normalized and squared.



(a) Company manufacturers.



(b) Company locations.



(c) Countries of bean origin.

Figure 11: Shares of categories for features considered for one hot encoding and target encoding. Note that the U.S.A. bar in (b) clearly exceeds the y-limit.

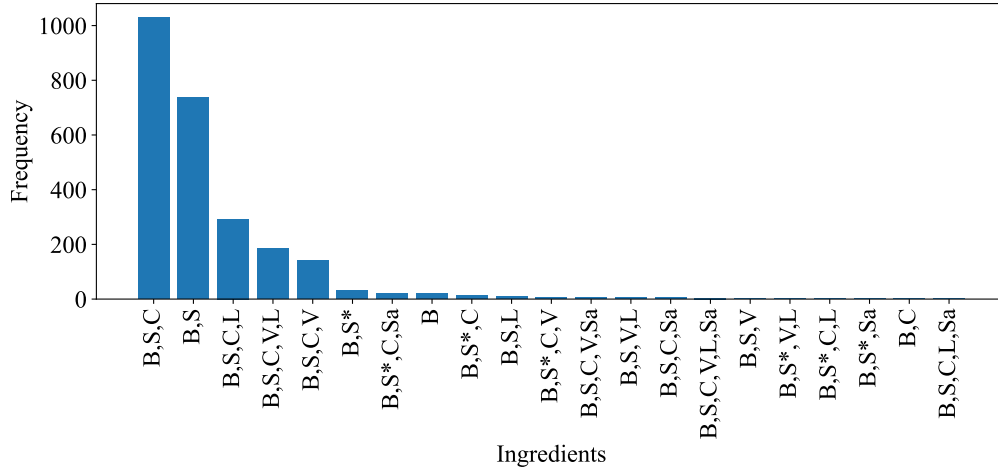


Figure 12: Frequency of the compositions of the chocolate bars.

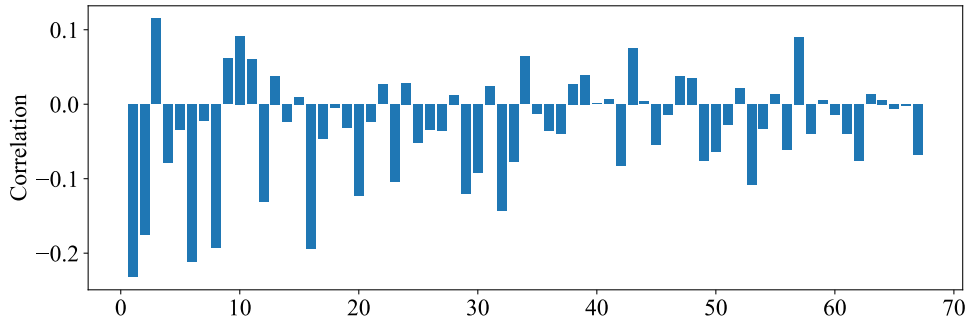


Figure 13: Correlation of the first 67 components from PCA of the embedded most memorable characteristics.

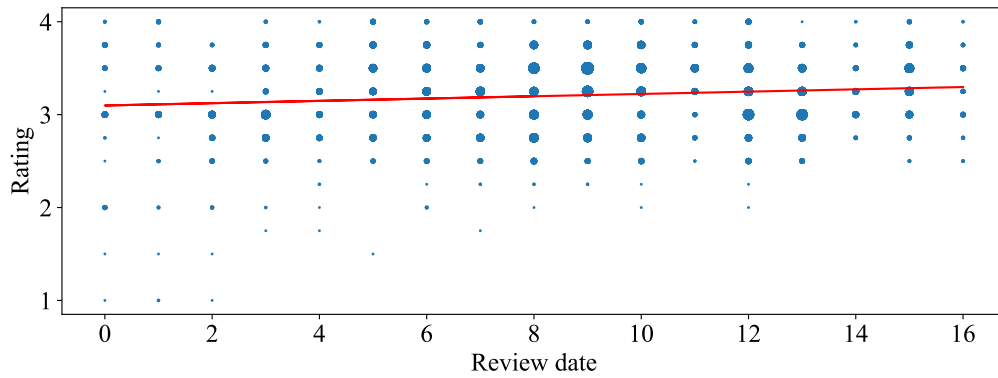


Figure 14: Rating depending on the label-encoded review date in the training data. Red line: Linear regression.

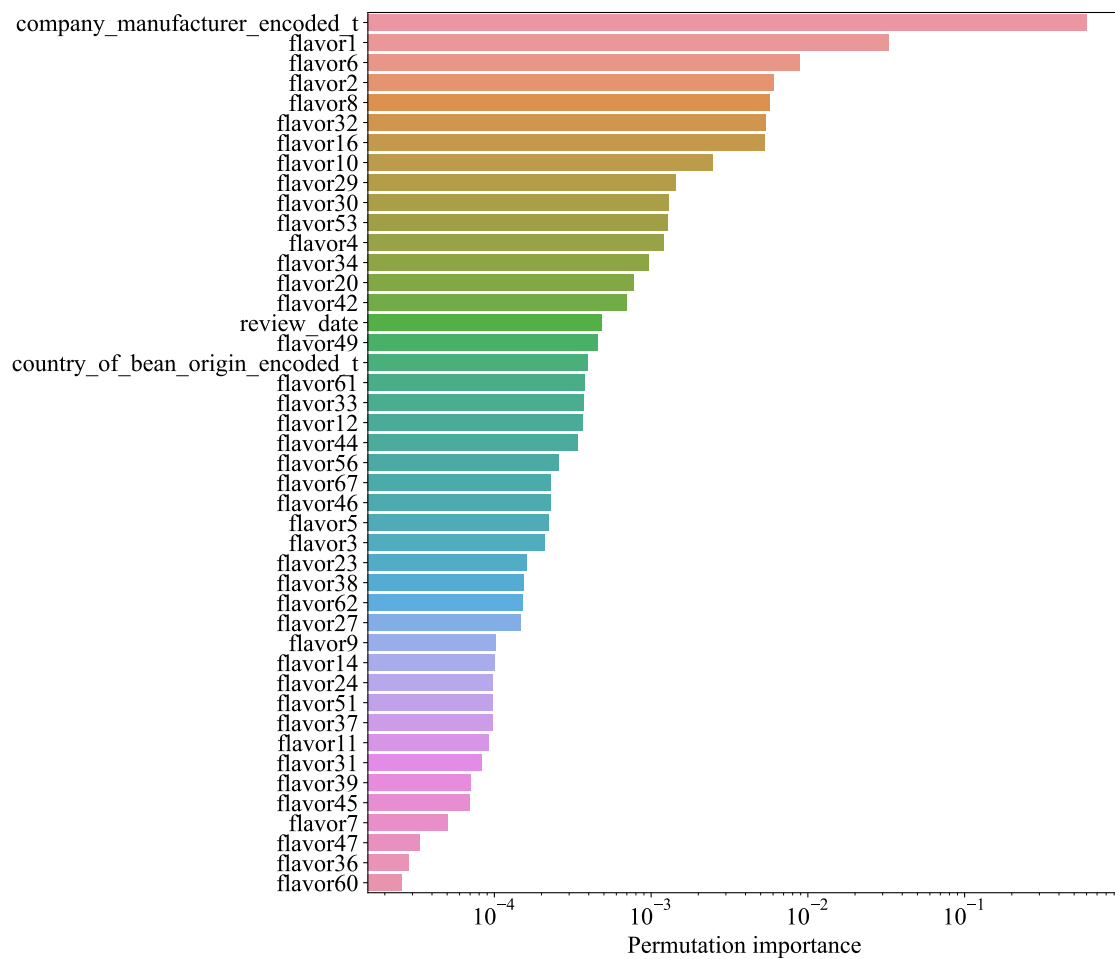


Figure 15: Non-zero feature importances in the initial random forest with flavor features.

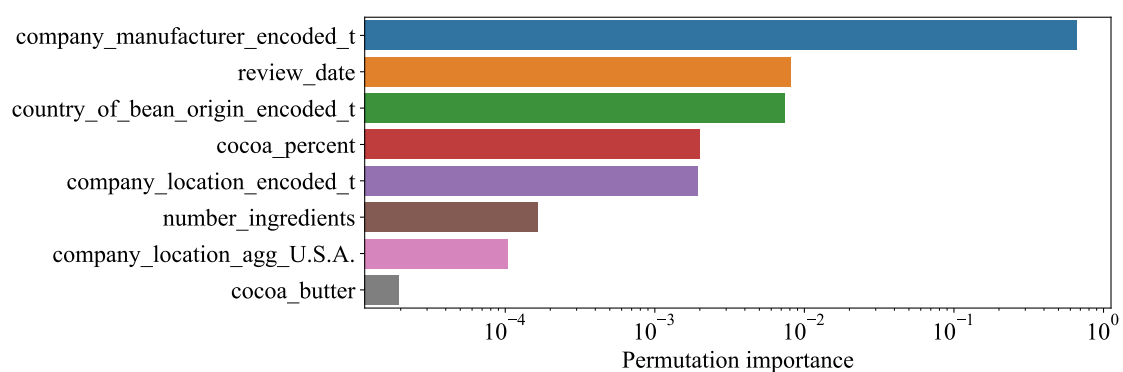


Figure 16: Non-zero feature importances in the random forest without flavor features.

A.2 Tables

Abbrev.	Ingredient
B	Beans
S	Sugar
S*	other Sweetener
C	Cocoa Butter
V	Vanilla
L	Lecithin
Sa	Salt

Table 8: Abbreviations used in the ingredients column.

baseline	cocoa percent review date (label encoded) number of ingredients (raw)
one hot encoded	dummies for the three most frequent company manufacturers dummies for the 11 most frequent company locations dummies for the 12 most frequent countries of bean origin
target encoded	company manufacturer (target encoded) company location (target encoded) country of bean origin (target encoded)
ingredient dummies	dummies for each ingredient
number ingredients (scaled)	normalized and squared number of ingredients
embedded flavours (x%)	features from PCA of the output of the text embedding explaining x% of the variance

Table 9: Groups of features.

Data and Code Provision

https://github.com/ElisabethBrockhaus/Chocolate_bar_rating