Coffee origin prediction

Terézia Mikulová, Jakub Balga

The aim of the project was to design a model that would be able to predict the origin of coffee (country) based on attributes such as cupping score, taste profile or text description of coffee.

Datasets

The first dataset used is a database of coffee samples from the Coffee Quality Institute (CQI). Part of the used dataset was obtained from https://www.kaggle.com/datasets/volpatto/coffee-quality-database-from-cqi. After preprocessing, newer data from the original database were added (https://database.coffeeinstitute.org/coffees)https://database.coffeeinstitute.org/coffees. The final dataset contains 1494 coffee samples. For the purposes of prediction, 10 numeric attributes with values in the range 0-10 (aroma, flavor, aftertaste, acidity, body, balance, uniformity, clean cup, sweetness and cupper points) were considered.

The second dataset we used is Sweet Maria's (SM) dataset which contains data from the Sweet Marias coffee retailer. We created the dataset by collecting the current and archived coffee offers from their website https://www.sweetmarias.com/green-coffee.html?sm_status=2. The dataset contains both cupping scores and a taste profile for each coffee. Additionally, it contains a short and a long description of the coffee and its flavour.

Since both CQI and SM datasets contained cupping scores, we initially considered merging the two datasets. However, the scoring approach seemed to differ, so we decided to work with them separately.

Due to the differences in the attributes of both datasets, we processed each separately.

Preprocessing

CQI dataset

The first step was to filter unnecessary attributes and remove coffees coming from countries for which there are not enough samples available (we set 100 samples per country as a minimum). Then we divided the dataset into a training and a test set, reserving 30% of the dataset for testing. Thus, the numbers of samples were as follows:

Country of origin	Train set	Test set
Mexico	159	77
Guatemala	139	62
Colombia	135	51
Brazil	100	40
Taiwan	81	34

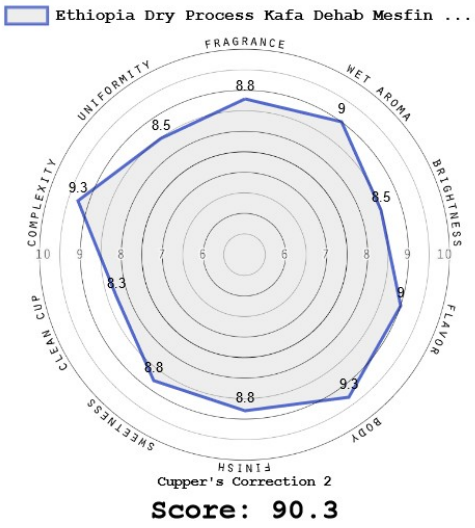
It can be seen that the dataset is not completely balanced, but the imbalance of classes is not so great that it has to negatively affect learning models.

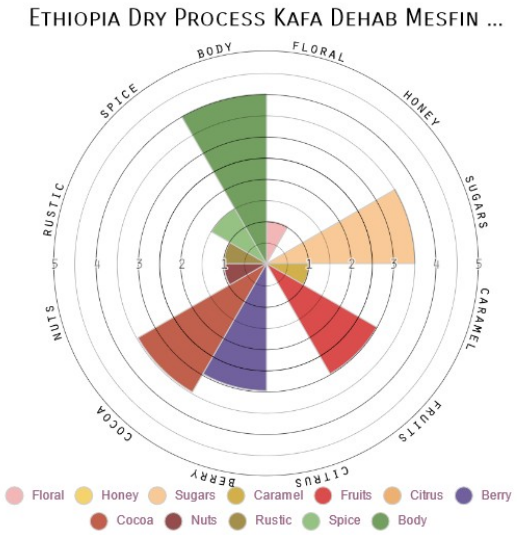
Sweet Maria's dataset

The initial dataset contained 326 examples and 38 attributes. For our project, we used only the five most frequent classes (which contained at least 25 examples). The resulting dataset contained 176 examples (Ethiopia 42, Guatemala 40, Kenya 36, Colombia 30, Indonesia 28).

We used 12 attributes on a scale from 0 to 5 to predict the country from the flavour. We used 9 attributes on a scale from 6 to 10 to predict country from cupping scores. Since the scales were already set and the same for all the attributes used for training one model, we did not normalize the attributes.

The following pictures show visualizations of the cupping scores and a taste profile from Sweet Maria's website:





We merged two available text descriptions to predict the country from a text description: an overview and a more detailed description. To work with the text descriptions, we created a bag of words and removed stopwords and country or location mentions from the words. This was not straightforward since the location could be pointed to even by a farm name.

We combined two approaches to remove most of the location mentions. First, we removed all capitalized words, even at the cost of losing other words (we did not expect them to be many). However, we did not remove capitalized words at the beginning of a sentence. To minimize the chance of leaving location info by leaving the first word, we created a country and a region list from dataset attributes and removed all such words. Also, all capitalized words before a word farm are removed. Hopefully, with this approach, only a small chance of leaving location information in a description exists.

Finally, from the bag of words, we created a TF-IDF vectors.

Example of the short description:

Delicious fruits and bittersweetness, notes of berry, cinnamon streusel, and floral accents. Also a syrupy textured espresso shot of inky dark chocolates with fruits underneath. City to Full City+. Good for espresso.

Example of the long description:

The dry fragrance shows a complex range of fruit and cocoa smells that are highly dependent on roast level. City roasts have a scent of fruit and sweet cream, like strawberry milkshake, with a background layer of cocoa. Full City roasts are a little less fruited, with focused cocoa bittersweetness. The wet aroma is teeming with fragrant browned sugars, dried fruits, and a creamy hazelnut accent. This is one of those naturals ("dry process") that's pretty difficult to mess up in the roaster. I made three passes, one City (14F development), City+ (23F Dev), and the darkest Full City (35F post 1st C). All three roasts were super sweet, complex, and big bodied. Compared to other Ethiopia's, this isn't a bright cup either, though the City roast has ample structure. Light roasts have a nice mixture of spice and fruit notes, strawberry fruit filling, cinnamon streusel topping, and hints of fruit jams. This, and the City+ roast, where the only two that provided some floral aromatics too, though lower intensity than the fruits. Full City roasts bring about intensely bittersweet chocolate flavors, like high % cacao bar, and some baking chocolate in the aftertaste. Within the cocoa flavor matrix is fruited sweetness too, and as the coffee cools, profiles of cooked fruits come into play as well. We don't give a whole lot of Ethiopia's the green light for espresso either, but Dehab's coffee works incredibly well at Full City roast level and beyond. The syrupy textured shot pulls out rich dark chocolates with a swirl of red berry. I don't think it's the ideal flavor profile for milk drinks, but deserves all the spotlight as a single origin offering.

Images and descriptions were retrieved from: https://www.sweetmarias.com/ethiopia-dry-process-kafa-dehab-mesfin-farm-7465.html (31.5.2023)

Evaluation metric

We used weighted F1 score as an evaluation metric. Classical (macro-averaged) F1 score weighs precision and recall for each class separately and then averages them. Weighted F1 score gets greater importance to more common classes by weighting each class by its support.

CQI – models

Dummy classifier

As a baseline, I used dummy classifier which predicts class randomly, with probabilities corresponding to the proportion of classes in training dataset. It gave F1 score of 0.25.

Multinomial logistic regression

Logistic regression is one of the basic classification methods, mapping the numerical output of linear regression (whose parameters are set during the learning process) to the probability of whether a sample belongs to a given class or not. In the binary case, we have only one linear regression output (that is, one probability). If we want to perform multi-class classification, we have to learn linear regression model for each class and during classification, transform their numerical output to

probabilities using softmax and choose the highest probability as predicted class. This is quite basic model, so I use it like a benchmark for the more sophisticated ones. The F1 score was 0.47.

Random forest classifier

Random forest is an ensemble of individual decision tree classifiers. Each decision tree is learned on a subset of data (which produces different trees) and when sample for classification appear, these trees vote for its class and the class with the most votes is selected. I reached F1 score of 0.50 with the random forest.

Ensemble of SVM classifiers

Support vector machine classifier (SVC) is a method that splits space of attributes by hyperplanes and classifies the samples by the subspace they belong to. To address class imbalance, I learned SVC for each of the classes separately – I took all negative samples (not belonging to the class) and oversampled the positive samples to have the same count of positives and negatives. Then I took the prediction from the classifier which was the surest about its prediction (returned the highest probability for its class). This ensemble reached F1 of 0.50.

Auto SKLearn classifier

Auto SKLearn classifier is an automated tool that examines performance of different machine learning algorithms and hyperparameter settings and picks the best one. In my case it chose random forest as a best model, giving F1 score of 0.52

Gradient boosting classifier

Gradient boosting classifier is an ensemble method, where similarly as in random forest classifier, several weak classifiers (individual trees) contribute to the final classification. During learning, new decision trees are created to fit classification errors of previous decision trees, so together they can get closer to the correct classification.

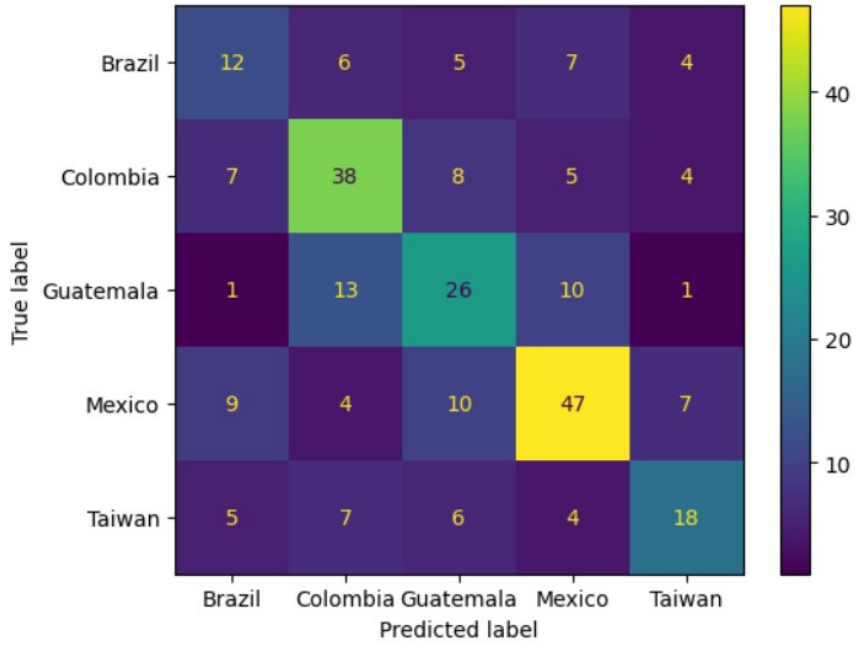
Further, I used SMOTE resampling technique to balance the classes. SMOTE creates new samples of minority class as weighted averages of similar (close lying) samples from that class.

Gradient boosting classifier proved to be the best one, but only slightly outperformed previous models, with F1 score 0.53.

Neural network

I also tried simple fully connected neural network, with 2 hidden layers (10 neurons each). However, it did not perform very well (F1 only 0.40), probably due to the limited number of samples.

Confusion matrix for the gradient boosting classifier



Conclusion

I was not able to reach a very high score on CQI dataset samples. This may be due to the small number of samples, poor relation between cupping scores and coffee origin and possibly large variation of climatic conditions within a single country. The taste profile appears to be much more informative for the purpose of determining coffee origin.

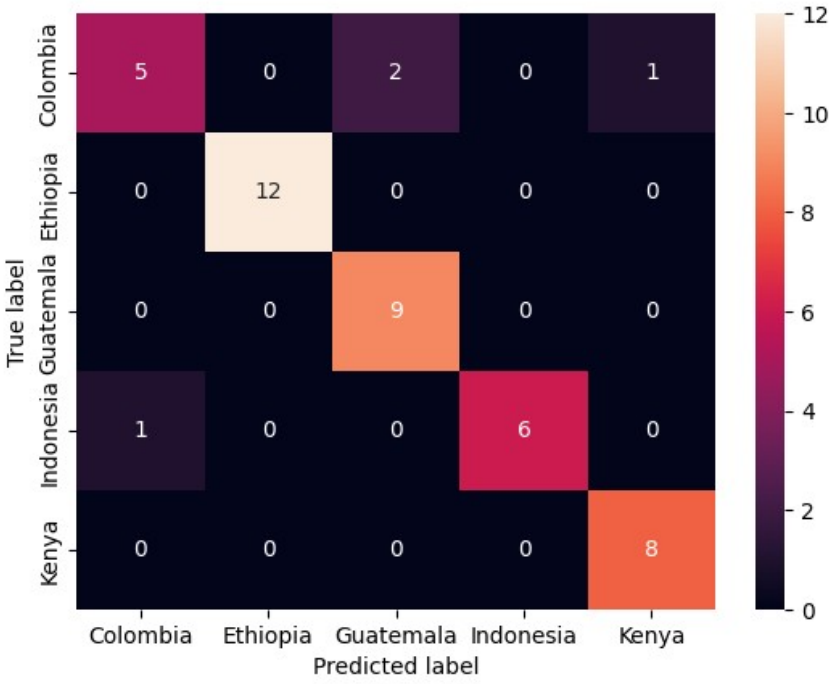
Sweet Maria’s - models

Baseline model

The baseline model used is the same as in the case of the CQI dataset.

TF-IDF with stochastic gradient descent model

This model predicts labels from a text description of a coffee represented as a TF-IDF vector. We used a stochastic gradient descent classifier with a log_loss setting, which results in a logistic regression model. This model performed the best compared to other approaches tried with the SM dataset.



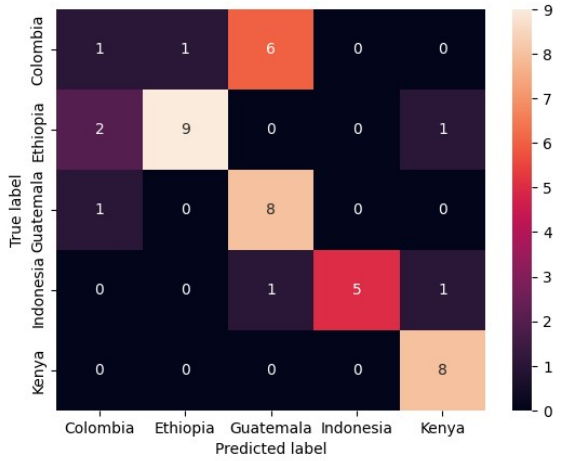
Random forest classifier with flavor attributes

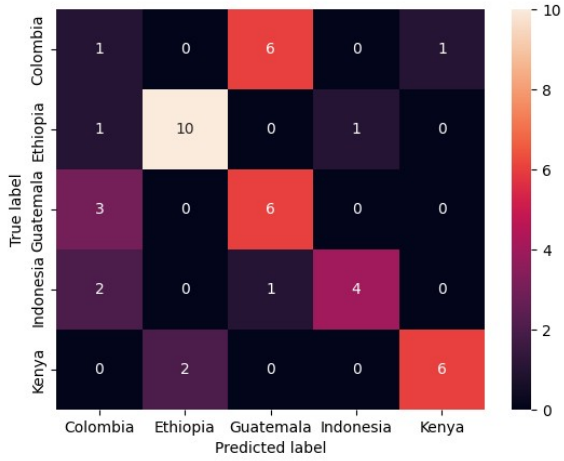
This model predicts the country from attributes that represent the coffee taste profile. The model had the second-best results. The model is set to 100 base learners, 5 maximum features and balanced class weights.

Random forest classifier with cupping scores attributes

The same model as in the previous section, only it predicts from attributes describing the coffee quality. This model was outperformed by the others.

Prediction from the flavour on the left, prediction from the cupping score on the right:





Comparision of all three models average F1 scores' obtained in cross-validation on the training dataset:

- Predicting from the description: 0.808
- Predicting from flavor: 0.681
- Predicting from the cupping score: 0.643

Comparision of all three models F1 scores' obtained on the testing dataset:

- Predicting from the description: 0.905
- Predicting from flavor: 0.684
- Predicting from the cupping score: 0.615
- Baseline model (stratified): 0.134

Implementation and usage

Learning models were implemented as scripts in Jupyter notebook. To test models for CQI, run individual cells sequentially in the notebook "cqi_models". For correct import of Auto SKLearn library, notebook should be run in Google Colab environment and first cell might be necessary to run 2 times.

To test models for Sweet Marias, run cells in the notebook “sweetmarias_models” sequentially. To see exploration and data visualization of the CQI dataset, see notebook “cqi_visualisation”.