What is most important in brewing process?

Have you ever thought what makes your coffee so delicious?

Is this connected with quality of beans, parameters of brewing process or just with skills of barista?

The main aim of his project is to find most important features of brewing process, but there is another value. I would like to make a model which will be able to predict mark of coffee witch accuracy at least of 0.75 and stability less than 5%.

So, let's go!

I would like to split project into parts:

1. My own data analysis based on data visualization
2. Taking care of missing values
3. Adding and removing columns
4. Getting dummies values for 'object' data
5. Fitting classifiers:
    - 5.1. Checking scores for basic data
    - 5.2. Looking for the best parameters of classifier - GridSearchCV
    - 5.3. Find the most important features
    - 5.4. Another GridSearchCV – only for features from 5.3
6. Checking scores for final model
7. Evaluation

1. **My analysis:**

I would like to start with basic things connected with dataset – types of values, properties of dataset, skew of each column etc.

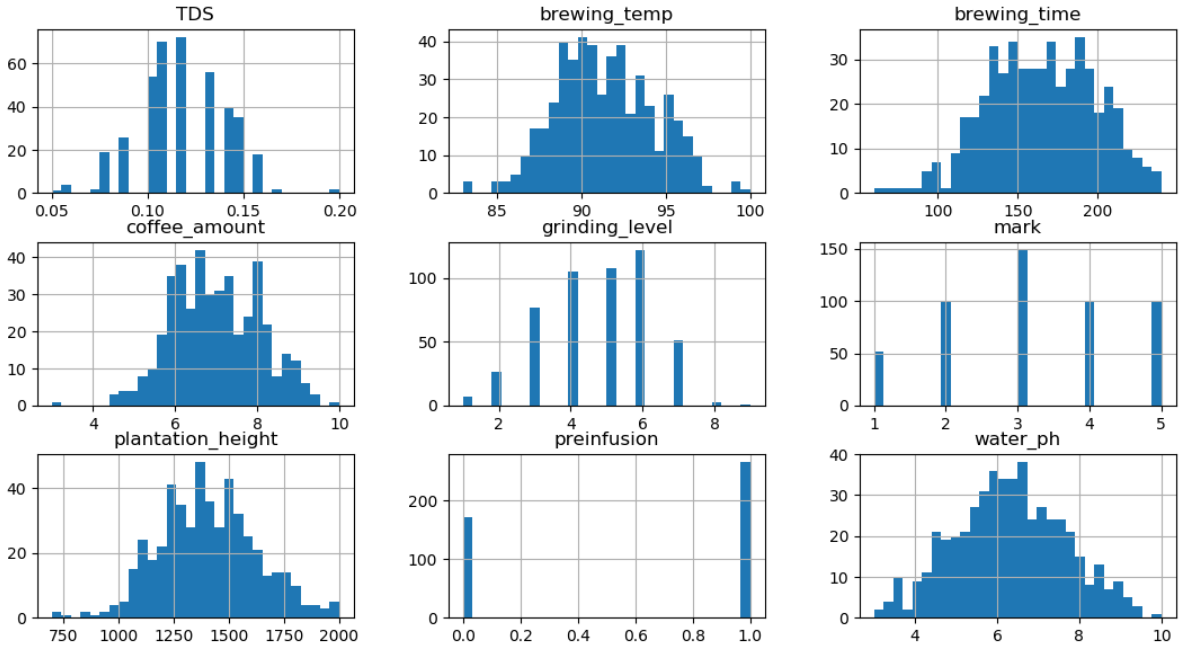
I would check it by functions:

df.dtypes -> it shows me types of each column
I know that I have only two non-numeric columns -> 'region' and 'processing_method'. Next step is to check the categorical values in each of these columns.

df.describe() -> helps me to get the view of the data from mathematical side
Here, I may see that there is some missing data in 'coffee_amount', 'preinfusion' and 'TDS' columns.

df.skew() -> it count the skew of each numeric column, the closer to 0 the more resemble the normal distribution

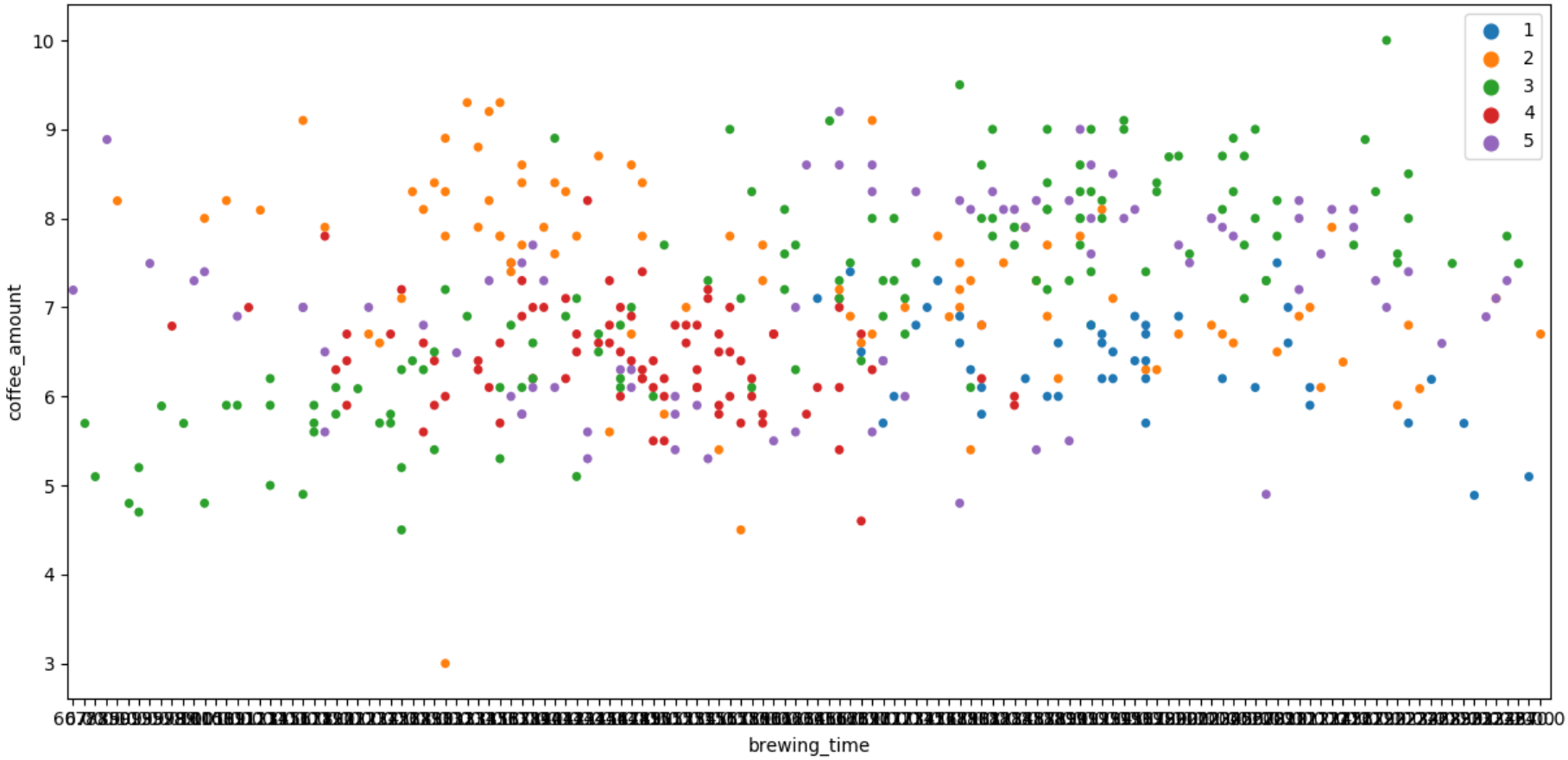
df.hist(bins = 30) -> shows histograms of each numerical data



As we can see, a lot of these columns are close to normal distribution. It means that during filling NaN's I may use median or mean, but these are not classical normal distribution so I will use only median to refill columns.

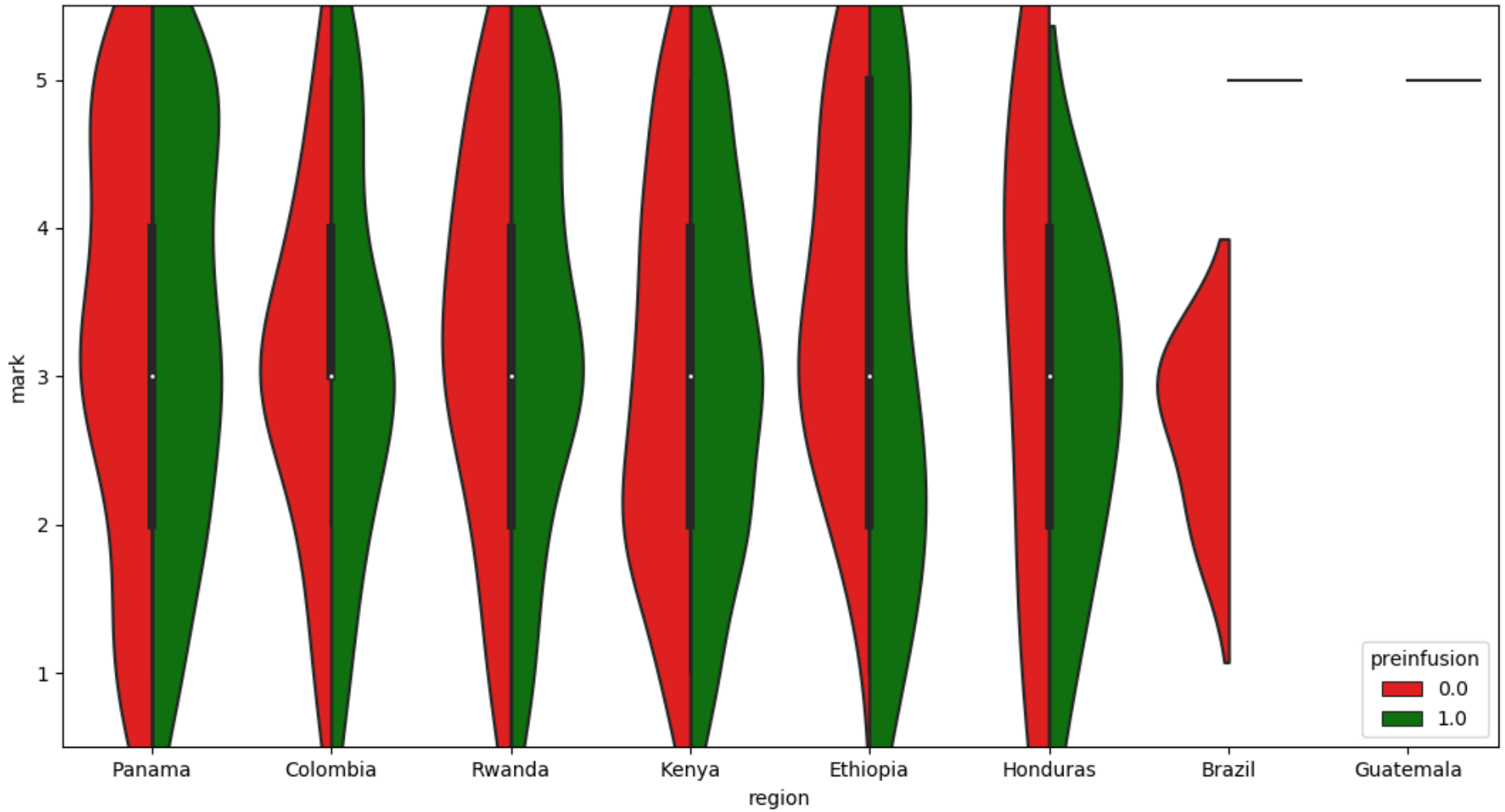
Visualizations:

Relation between **brewing time** and **coffee amount**



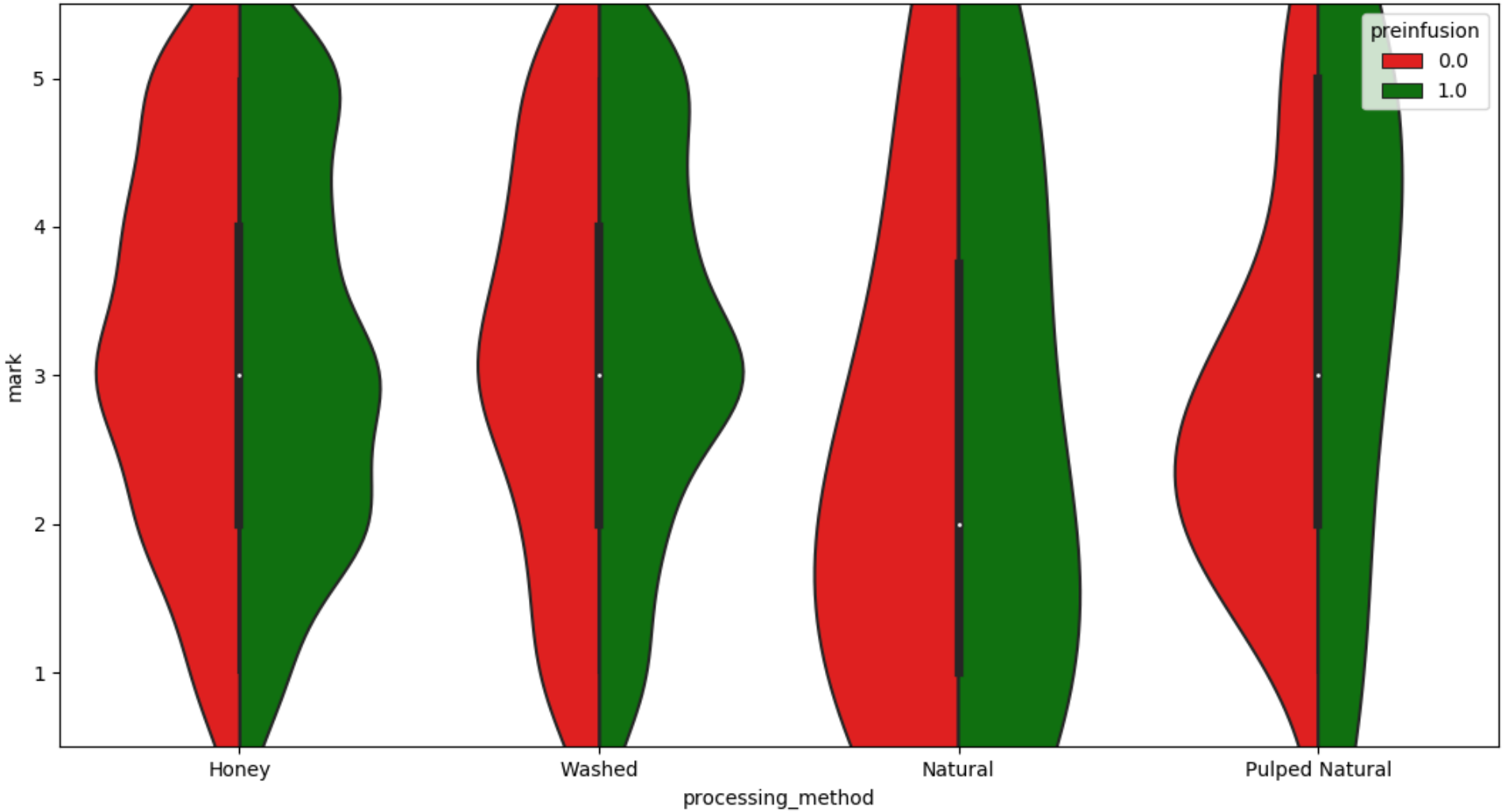
I hoped that there will be clear dependence between these columns. Unfortunately, there is not :(but I may use these columns to make another – **essential** – it would describe how many 'essence' of coffee came into fluid during brewing.

Relation between **region** and **mark**



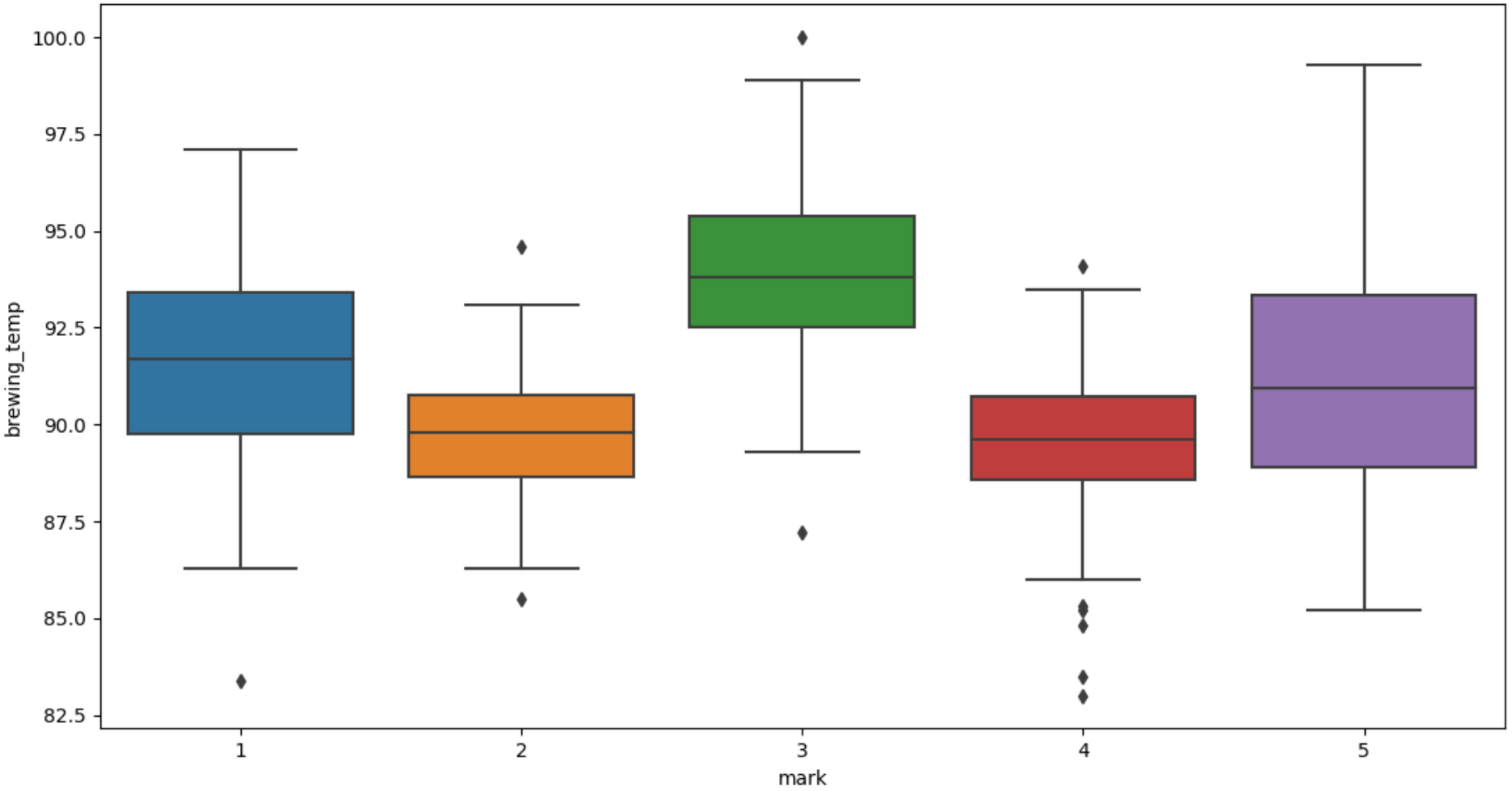
Also here we could not find anything special. There is no strong dependence between these columns. There is only one not normal thing, two completely flat violin plots for Brazil and Guatemala. It is caused by only one record for these countries connected with mark and preinfusion. I will let it be and not remove these columns yet but I think that in the final model I will not use them.

Relation between **processing method** and **mark**



Here might be correlation between these columns but it is not strong. Sides of violin plot are mostly similar witch one exception – **'Pulped Natural'** process -> there is bigger probability to get 2 mark than others and if we use preinfusion we probably will make better coffee. That might be important in final model.

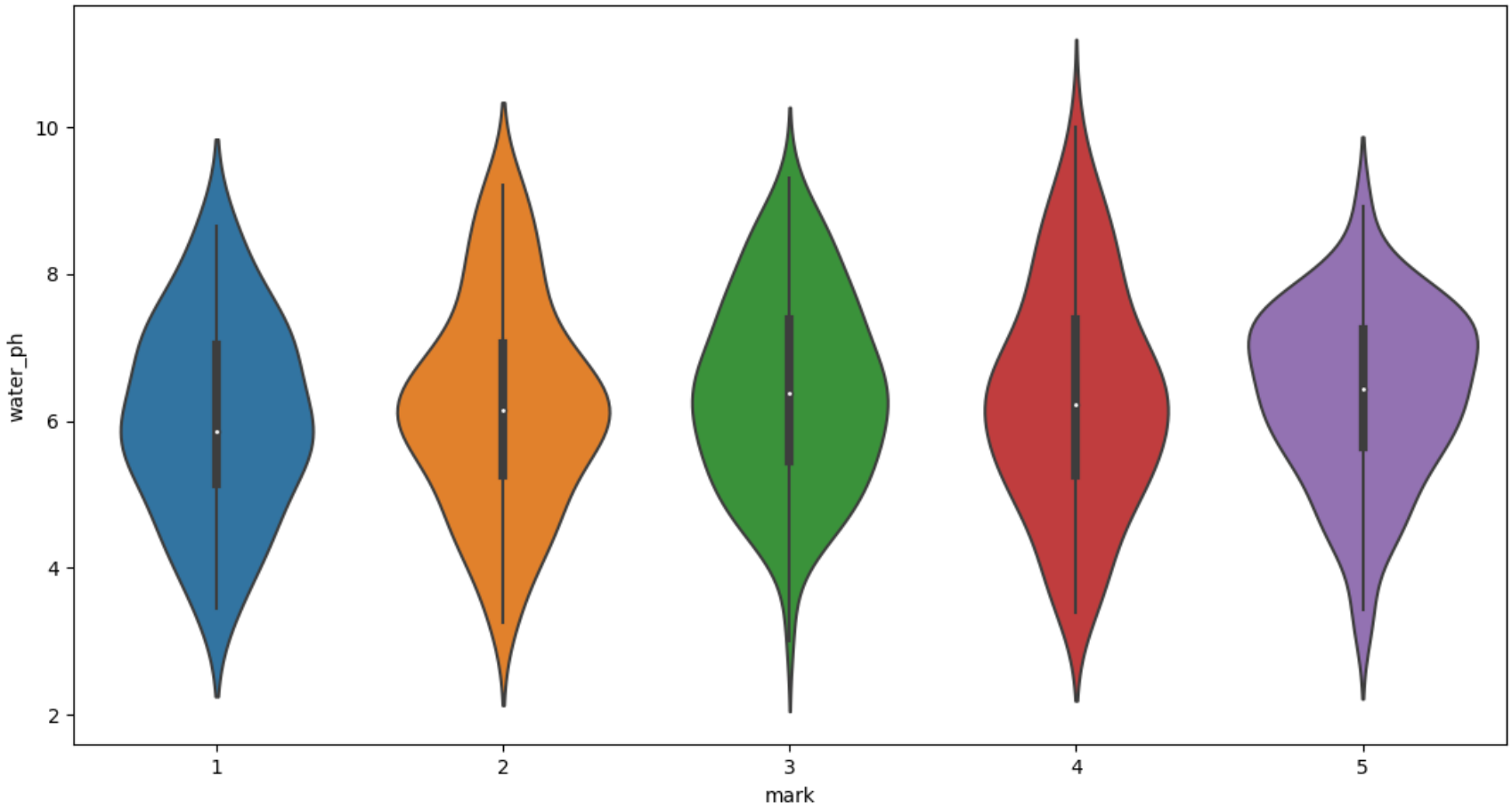
Relation between **brewing temp** and **mark**



From this boxplot we can learn that higher temperature increases probability of getting 3 mark, but there in completely no connection between other marks and temperature. I think that for each kind of beans there is right temperature to get the best taste.

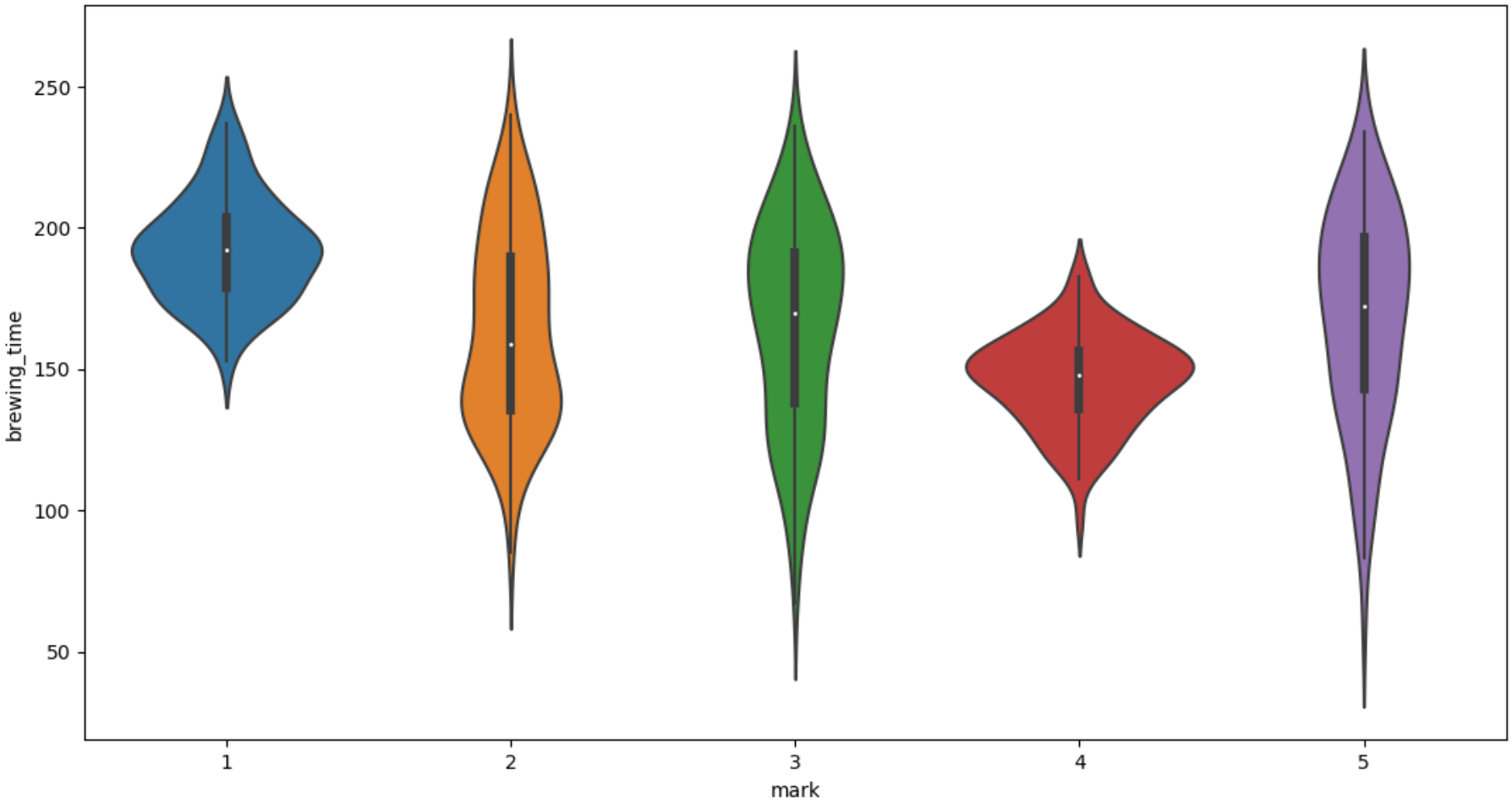
Analysis of relation between **coffee amount** and **mark** gives me the same conclusion. Each kind of coffee needs right amount.

Relation between **water pH** and **mark**



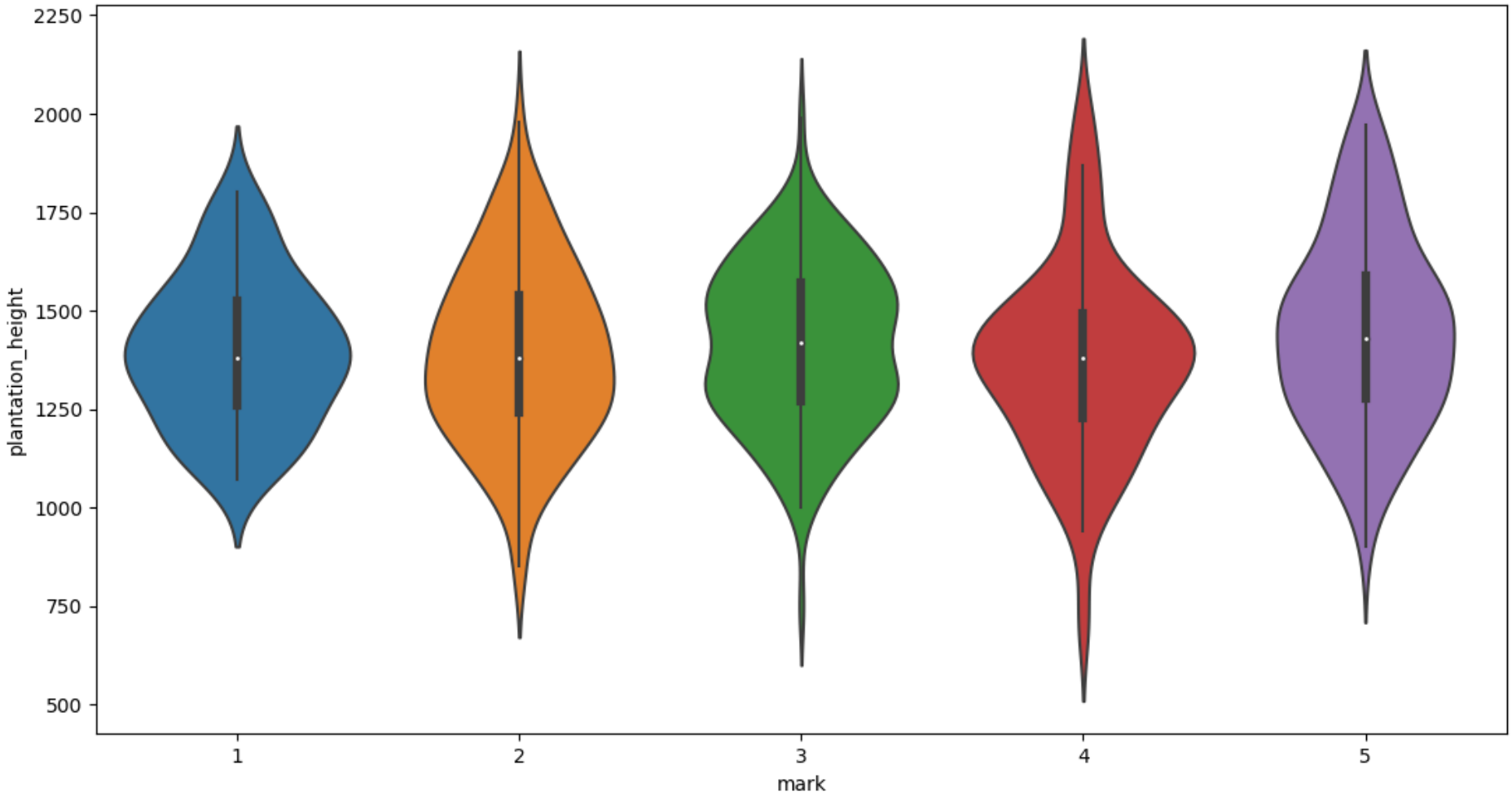
Most of these coffees have been made in the same range of pH. We may see that there is soft correlation, higher pH is important when we want to get 5 mark, and if we use lesser pH water we probably get lesser mark. I think that there might be correlation between kind of beans and water pH. Most of coffees are acid so making them in acid water might not be good idea.

Relation between **brewing time** and **mark**



During brewing process time is important parameter. We cannot boiled our coffee and let it in water for too much time. We can see the same things here. There is bigger probability to get good coffee witch decreasing time of brewing. One unexpected thing is mark 5, but I think that there could be correlation between kind of beans and brewing time. At least I have found visible dependence.

Relation between **plantation height** and **mark**



Most of these coffees have been growing on the same height. I cannot see the correlation between these columns, but I think that there important is relation region – plantation_height. It may describe the field and climate conditions of each region. I will fool for this data.

I have made some research on the internet. I had found site cafevirtuoso.com and there I have read an article about parameters of ideal coffee. The author have written that most important thing are:

- Grinding level
- Kind of coffee
- Quality of water
- Brewing temperature

I will add: essential, energy (all energy which came inside the coffee – brewing_temp*brewing_time) and robustness (grinding_level/TDS). I will take care of these columns after filling missing values.

2. Taking care of missing values:

To get precise information about missing values I have made a data frame witch three columns:

- Nulls -> boolean value, shows if there is any missing value
- NumberOfNan -> sum of all Nan values in each column
- Percentage -> percentage share of Nan values in each column

To fill missing places I will grouped data for each value and replace Nan's witch median of all values in each group.

To this time I have known that most important values are: **brewing_temp**, and **brewing_time.** Unfortunately in these columns are too much different values. To make them useful I have to add new columns: **temp_level** and **time_level.** For each I will split main column for three levels and based on it create next column.

Index	Nulls	NumberOfNan	Percentage
region	False	0	0.000000
brewing_temp	False	0	0.000000
coffee_amount	True	66	13.200000
preinfusion	True	64	12.800000
grinding_lev…	False	0	0.000000
TDS	True	99	19.800000
water_ph	False	0	0.000000
plantation_h…	False	0	0.000000
processing_m…	False	0	0.000000
brewing_time	False	0	0.000000
mark	False	0	0.000000

Coffee amount:

Coffee amount is a part of coffee preprocessing. You have to know the temperature and time of brewing, another important thing is grinding level.

temp_level is connected with kind of beans and kind of beans defines strength of coffee. The stronger taste the less coffee is needed to make great coffee.

time_level -> if you take more coffee you have to spend more time to brew it.

grinding_level is the most important. If grinding level is low you need more coffee to get the same results.

Based on it I am creating a grouped data frame with 51 different groups and median value of **'coffee_amount'** for each one. Based on it I am replacing Nan's with these values. Unfortunately, one of group still has Nan, so I am replacing this Nan with mean of all column.

Index	grinding_level	time_level	temp_level	coffee_amount
0	1.000000	1.000000	2.000000	nan
1	1.000000	1.000000	3.000000	4.800000
2	1.000000	2.000000	3.000000	5.800000
3	1.000000	3.000000	2.000000	7.100000
4	1.000000	3.000000	3.000000	4.500000
5	2.000000	1.000000	1.000000	6.300000
6	2.000000	1.000000	2.000000	5.900000
7	2.000000	1.000000	3.000000	5.000000
8	2.000000	3.000000	1.000000	5.900000
9	2.000000	3.000000	2.000000	5.650000
10	2.000000	3.000000	3.000000	5.150000

TDS:

TDS is a level of solid-state parts in fluid after brewing. I will use the same values as previous.

Cause of taking **temp_level** is the same as above.

time_level -> there should be higher probability of getting solid-state parts when time of brewing is longer.

grinding_level is, one more time, the most important. If you get high grinding level you will get smaller parts of coffee and it is easier to go through sieve.

One more time I get data frame with 51 different groups. I will replace Nan's with median of each group. As we can see there is still one Nan values. The same as above I will fill it with mean of all column.

Index	temp_level	time_level	grinding_level	TDS
0	1.000000	1.000000	2.000000	nan
1	1.000000	1.000000	3.000000	0.120000
2	1.000000	1.000000	4.000000	0.105000
3	1.000000	2.000000	3.000000	0.120000
4	1.000000	2.000000	4.000000	0.110000
5	1.000000	2.000000	5.000000	0.110000
6	1.000000	2.000000	6.000000	0.120000
7	1.000000	2.000000	7.000000	0.120000
8	1.000000	3.000000	2.000000	0.140000
9	1.000000	3.000000	3.000000	0.100000
10	1.000000	3.000000	4.000000	0.110000

Preinfusion:

I will use the same method as above – filling Nans with median of each different group. Preinfusion is a part of preprocessing so I will connect it only with values describing process before brewing.

In this case I will use only two values:

grinding_level -> there should be higher probability of getting preinfusion for bigger parts of coffee

processing_method -> it may describe kind of beans

As we can see there are three levels: 0, 0.5 and 1, but preinfusion should be only 0 and 1. After implementation, I checked that in all dataset there is no 0.5 value, so I will not change the method.

Index	grinding_level	rocessing_metho	preinfusion
0	1.000000	Honey	1.000000
1	1.000000	Washed	1.000000
2	2.000000	Honey	1.000000
3	2.000000	Natural	0.000000
4	2.000000	Washed	1.000000
5	3.000000	Honey	1.000000
6	3.000000	Natural	0.500000
7	3.000000	Pulped Natural	0.500000
8	3.000000	Washed	1.000000
9	4.000000	Honey	1.000000
10	4.000000	Natural	1.000000

3. Adding and removing columns:

As I mentioned I will need add some columns:

- **energy = brewing_time*brewing_temp** -> there should be right (for each kind of beans) amount of energy provides to coffee
- **robustness = grinding_level/TDS** -> describes the probability of getting solid-state parts in coffee
- **essential = brewing_time*coffee_amount** -> describes how much 'taste' will come into coffee during brewing process
- **barist_rank** -> region mapped by dictionary based on barista rank of each country (source: www.thrillist.com)
- **humidity** -> humidity of beans in processing method, levels based on Fobonacci's Code (source: blog.seattlecoffeeworks.com)
- **fermentation** -> fermentation of beans in processing method, levels based on Fobonacci's Code (source: blog.seattlecoffeeworks.com)

At next states of project I will not need **temp_level** and **time_level**, so I am removing these columns.

4. Getting dummies:

Right now I can split my dataset into independent (X) and dependent (Y) values. After that I am finishing data preprocessing with getting dummie variables for **'object'** columns -> **'region'** and **'processing_method'.**

I have finished with data frame contained: 500 rows and 28 columns.

Last step is scaling the data. I will use **MinMaxScaler** witch **feature_range = (0, 5)**

5. Fitting classifiers:

5.1 Checking scores for basic data

I will split dataset into train and test with **test_size = 0.15**

I will use two different algorithms: **RandomForestClassifier** and **XGBClassifier.** After first fitting I will check the scores and choose the best. Next I will use GridSearchCV to get the best parameters and RFE to choose most important features. At the end I will check accuracy on a test set.

Metrics: **accuracy** -> mean accuracy from cross validation
stability -> standard deviation of accuracy * 100 / mean accuracy

In first fitting I have used completely basic classifiers, and scores are:

	Accuracy	Stability
XGBClassifier	**0,7048**	**8,56%**
RandomForestClassifier	**0,6610**	**9,98%**

It is clear to see that better scores are with **XGBClassifier** so I will take it into next steps.

5.2 Looking for the best parameters of classifier - GridSearchCV

Next step is checking the best parameters of classifier. I will use **GridSearchCV** for different parameters: **learning_rate, max_depth** and **n_estimators.**

Results:
Best score: **0,7365**
Best parameters: **'learning_rate': 0.5, 'max_depth': 6, 'n_estimators': 30**

5.3 Find the most important features

To find the most important features I will use RFE method. I will take 20 iteration of calculations for different numbers of variables (from 3 to 12). Each calculation will be made with using cross validation (splitting dataset into 10 subsets). I will measure accuracy and stability and take the best score (highest ratio -> Accuracy / Stability).

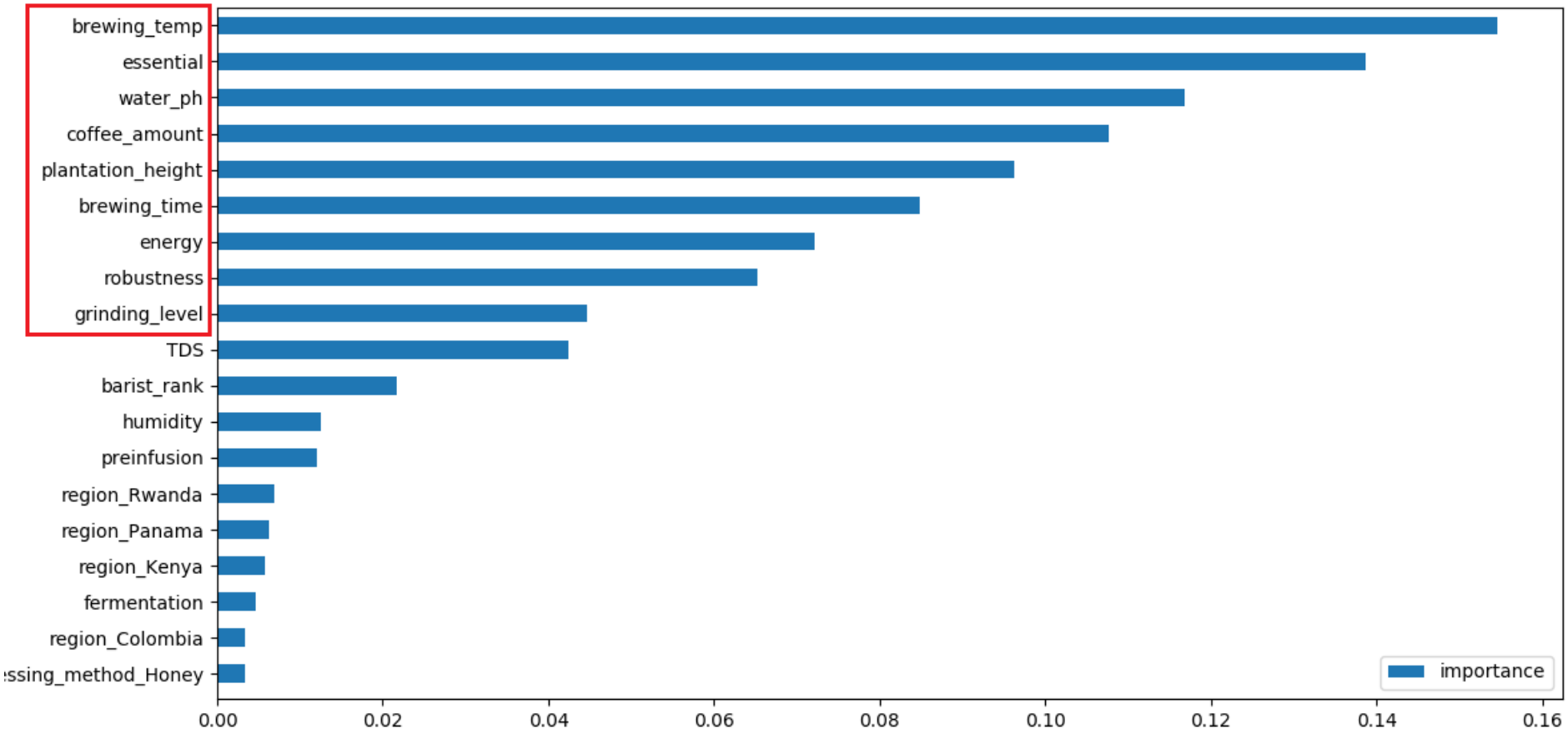
Results:

	3	4	5	6	7	8	9	10	11	12
Accuracy	**0,407**	**0,605**	**0,602**	**0,67**	**0,687**	**0,686**	**0,712**	**0,72**	**0,729**	**0,733**
Stability	**12,35**	**13,25**	**10,09**	**11,11**	**7,44**	**8,52**	**6,7**	**6,87**	**7,02**	**6,93**
Ratio	**0,0329**	**0,0456**	**0,0597**	**0,0603**	**0,0923**	**0,0806**	**0,1063**	**0,1047**	**0,1038**	**0,1058**

We will get the best results if we take 9 best values. And these are:

- **'brewing_temp',**
- **'coffee_amount',**
- **'grinding_level',**
- **'water_ph',**
- **'plantation_height',**
- **'brewing_time',**
- **'energy',**
- **‘robustness’,**
- **'essential'**

Using **feature selection** I am creating plot which shows the importance of each variable:



As we can see, my variables (**essential, energy** and **robustness**) are in most important part.

Truly, that could be everything. I have checked which parameters are most important in coffee processing. I could finished here but I would like to check my model on test set.

5.3 Another GridSearchCV – only for features from 5.3

Now I will repeat **GridSearchCV** for most important variables. I will checked the same parameters as in first attempt.

Results:
Best score: **0,7412**
Best parameters: **'learning_rate': 0.05, 'max_depth': 6, 'n_estimators': 130**

6. Checking scores for final model:

For last classifier I will use parameters form **5.3** to predict test values.

After fitting classifier I have got results:

	Accuracy
XGBClassifier	**0,72**

7. Evaluation:

In my opinion I have been working on incomplete data. During research about brewing I have got the knowledge that the most important is kind of beans and their parameters. As we can see, our data has only parts of the data (**'region', 'processing_method'** and **'plantntion_height'**). To get better results we will need another columns, for example:

- beans_ph
- beans_rank
- beans_price
- beans_strenght

My aim was to get >0,75 accuracy and <5% stability. At the end I get 0,72 accuracy and 6,7% stability. Unfortunately, this model is useless when we want to predict the mark and cannot be habituate. On the other hand it is great model to check the importance of each data and check what is the most important during brewing process.