Which parameters are most important in cancer classification?

Have you ever thought what may cause malignant cancer?

Is this connected with area, maximum radius or irregularity of cancer?

The main aim of his project is to find most important features causes malignancy, but there is another value. I would like to make a model which will be able to predict if the cancer might be malignant witch accuracy at least of 0.9 and stability less than 3%.

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. Find the most important features
    - 5.3. Looking for the best parameters of classifier - GridSearchCV
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.

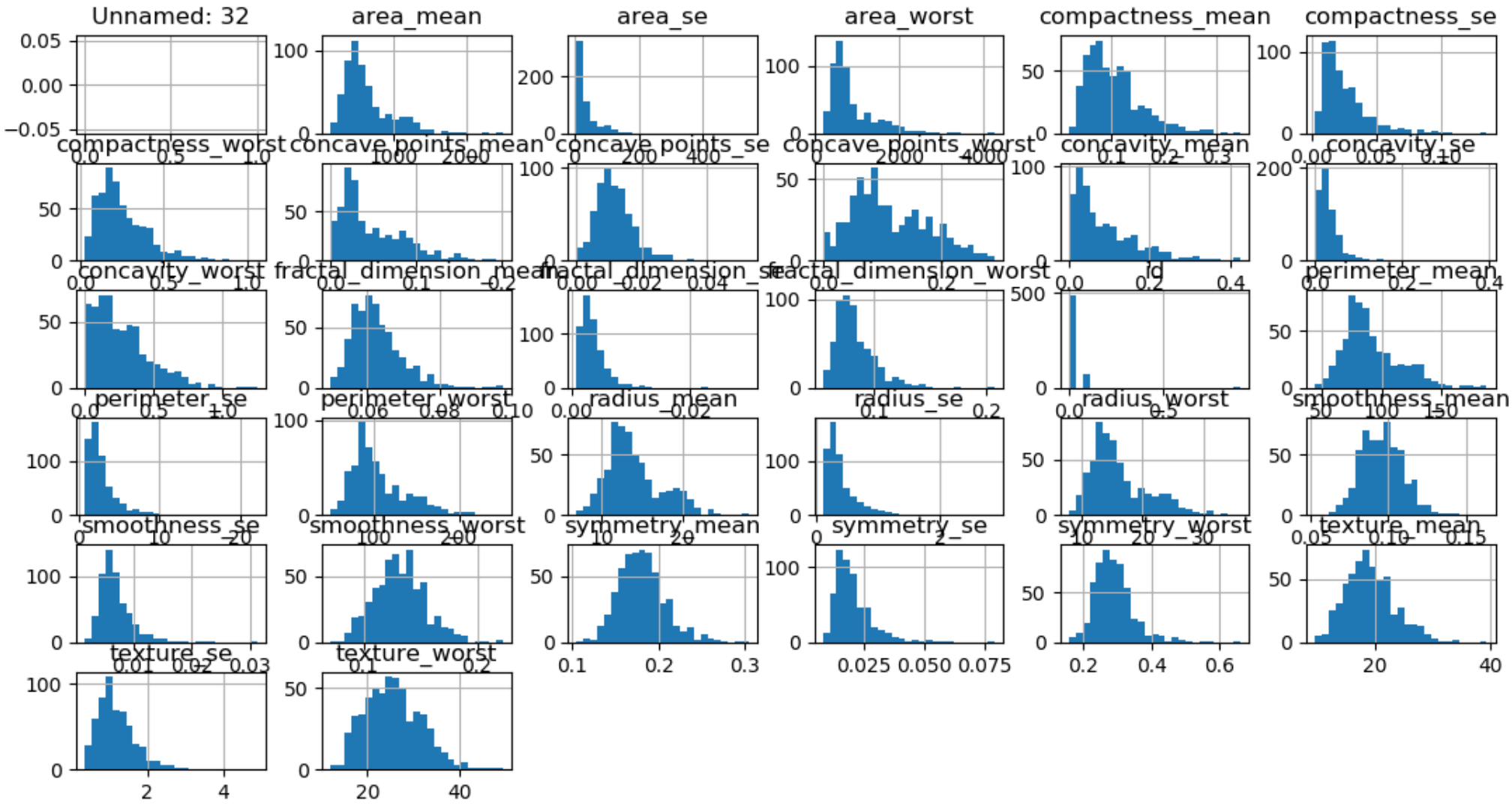
I would check it by functions:

df.dtypes -> it shows me types of each column
I know that there is only one object column -> 'diagnosis'. There are only Boolean values so I will map this column into 0 (normal cancer) and 1 (malignant).

df.describe() -> helps me to get the view of the data from mathematical side
Here, I may see that there is no missing values. Lucky day :)

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

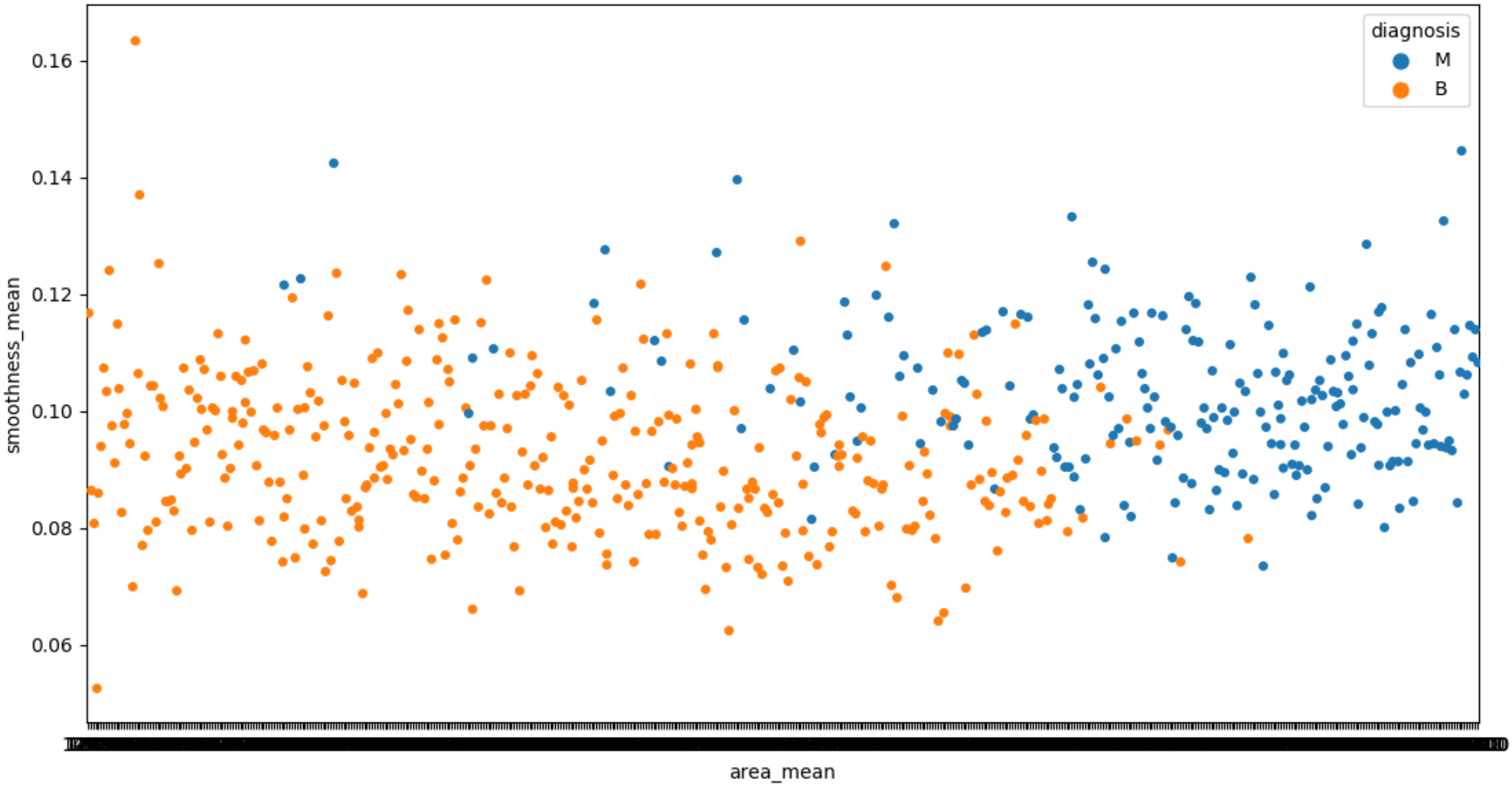
df.hist(bins = 25) -> shows histograms of each numerical data



As we can see, a lot of these histograms are moved into left side. In few cases (**'concave_points_worst', 'radius_mean', 'preimeter_mean'**) we can see huge dispersion. Looking at these histograms I may state that each NaN value I should replace by dominant value of right column.

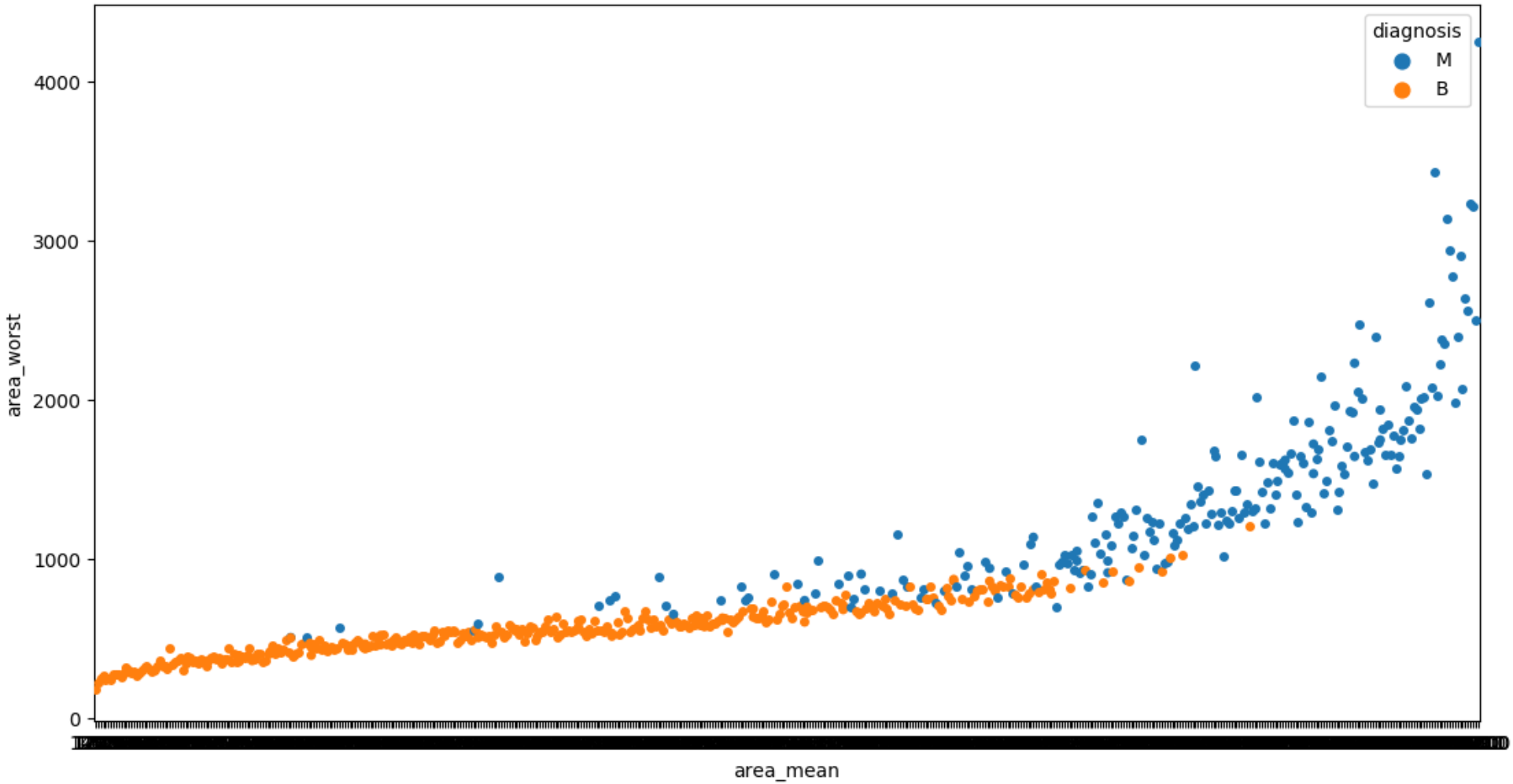
Visualizations:

Relation between **smoothness_mean** and **area_mean**



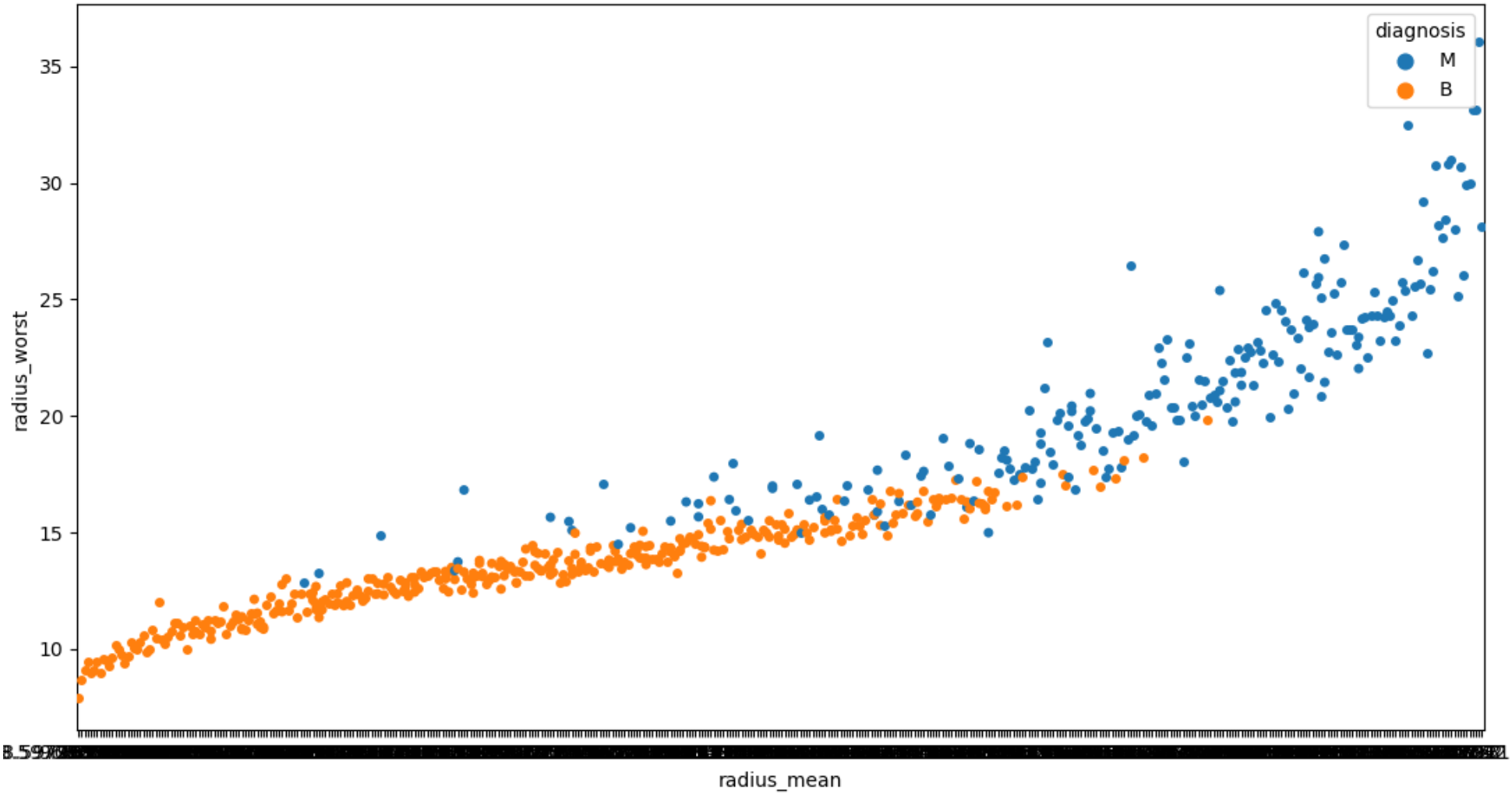
I hoped that there will be clear dependence between these columns and diagnosis, and I was right. As we can see there is strong correlation between **area_mean** and **diagnosis.** Unfortunately, there is no correlation between **smoothness_mean** and **diagnosis.**

Relation between **area_mean** and **area_worst**



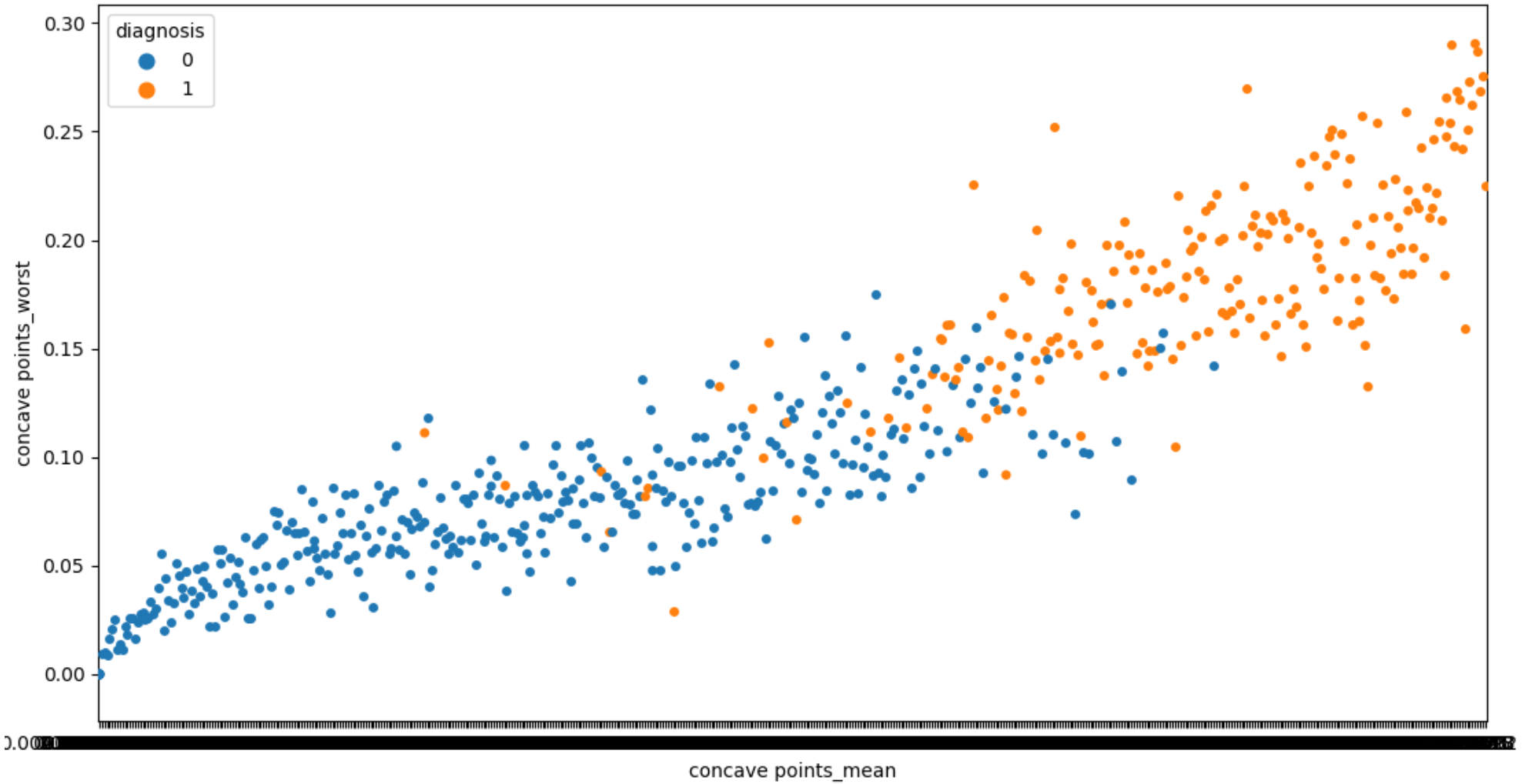
Great plot. We can see two separate datasets for normal and malignance cancer. In the future I should create another column describes **area_worst / area_mean** ratio. It may help the model to get the best score.

Relation between **radius_mean** and **radius_worst**



Also here we can observe strong correlation. We can split this dataset into two independent sets. In the future I will make new column -> **radius_worst / radius_mean** ratio.

Relation between **concave points_worst** and **concave points_median**



Another strong correlation. I should use this data to create another column: **concave points_worst / concave points_median** ratio, but there are 13 rows with **concave points_median = 0**, so I should not make a column with NaN values.

I have made some research among students of medicine. They have told me some things about parameters which may cause malignant cancer. These parameters are:

- Genetics
- Area
- Irregularity of cancer
- Age

Unfortunately I do not have information about genes and age of each person so I will add few features describing irregularity of cancer.

2. Taking care of missing values:

Fortunately there is no missing values.

3. Adding and removing columns:

As I mentioned I will need add some columns:

- **distortion_area = area_worst / area_mean** -> it will describe irregularity of area size
- **distortion_radius = radius_worst / radius_mean** -> it will describe irregularity of radius length

4. Getting dummies:

There is no categorical column so there will not be any dummy variables.

5. Fitting classifiers:

5.1 Checking scores for basic data

I will split dataset into train, validation and test with **test_size = 0.2** for test set and **0.1** for validation set.

I will use validation set to check accuracy and make ROC curve, after first classification I will check my model in action on test set.

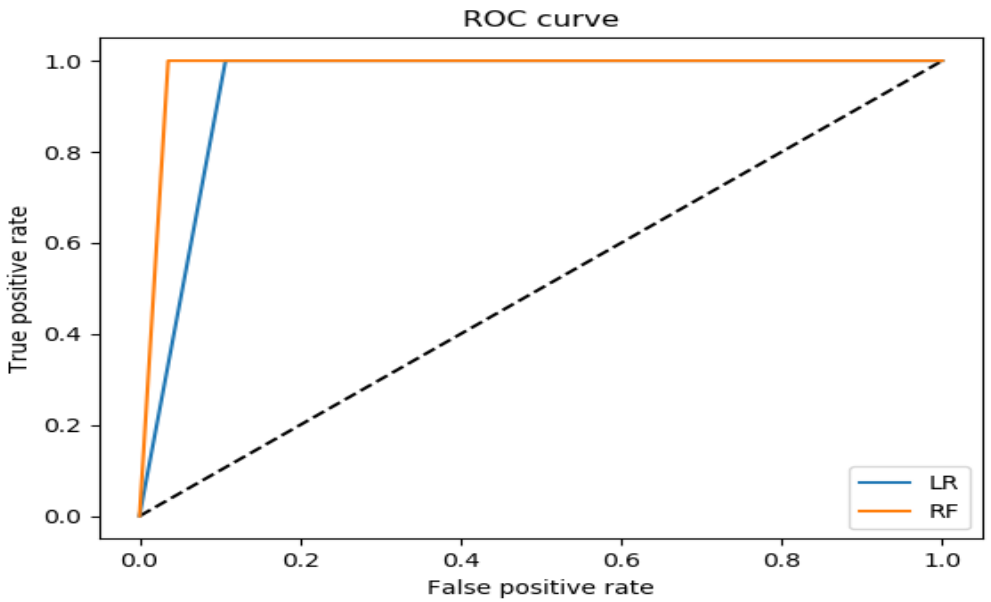
I will use two different algorithms: **RandomForestClassifier** and **LogisticRegression.** After first fitting I will check the scores and choose the best. Next I will use RFE to choose most important features, next I will use GridSearchCV to get best parameters. At the end I will check accuracy on 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
LogisticRegression	**0,9348**
RandomForestClassifier	**0,9782**

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



5.2 Find the most important features

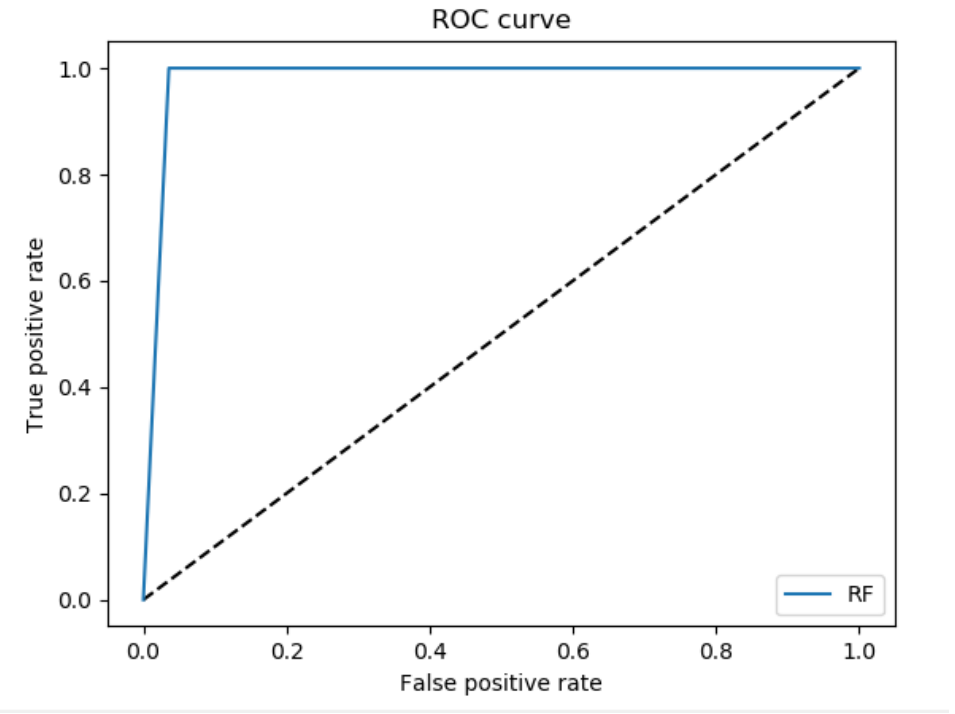
To find the most important features I will use RFE method. I will take 20 iteration of calculations for different numbers of variables (from 3 to 12). Each calculation will be made with using cross validation (splitting dataset into 10 subsets). I will measure accuracy and stability and take the best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

Results:

	3	4	5	6	7	8	9	10	11	12
Accuracy	**0,9207**	**0,9194**	**0,9372**	**0,9393**	**0,9432**	**0,9479**	**0,9463**	**0,9501**	**0,9516**	**0,9525**
Stability	**3,7768**	**3,7017**	**3,1748**	**3,0803**	**2,9158**	**3,0537**	**2,9548**	**2,8317**	**2,7351**	**2,9255**
Ratio	**0,2438**	**0,2484**	**0,2952**	**0,3049**	**0,3235**	**0,3104**	**0,3203**	**0,3355**	**0,3479**	**0,3256**

We will get the best results if we take 11 best values. And these are:

- **'texture_mean',**
- **'concave points_mean',**
- **'perimeter_se',**
- **'area_se',**
- **'radius_worst',**
- **'perimeter_worst',**
- **'area_worst',**
- **'concavity_worst',**
- **'concave points_worst',**
- **'distortion_area',**
- **'distortion_radius'**



As we can see, my variables (**'distortion_area'** and **'distortion_radius'**) are in most important values. After checking importance and use only columns from this list, the accuracy gain to **0,9873.**

5.3 Looking for the best parameters of classifier - GridSearchCV

Now I will use **GridSearchCV** for most important variables.

Results:
Best score: **0.9692**
Best parameters: **'criterion': 'gini', 'max_depth': 5, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 40**

After checking model on cross validation:

	Accuracy	Stability
XGBClassifier	**0,958**	**2,17%**

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	**0,9736**

7. Evaluation:

In my opinion I have been working on splendid data. This was complete, looked-after dataset with no NaN values. That was a pleasure to make this project. I think, that I will be able to get better scores if I have more patient describing data.

My aim was to get >0,90 accuracy and <3% stability. At the end I get 0,9736 accuracy and 2,17% stability. Result of my work is complete model which may help doctors in breast cancer treatment.