Documentation about modeling process

Ivan Grujičić

30. decembar 2020.

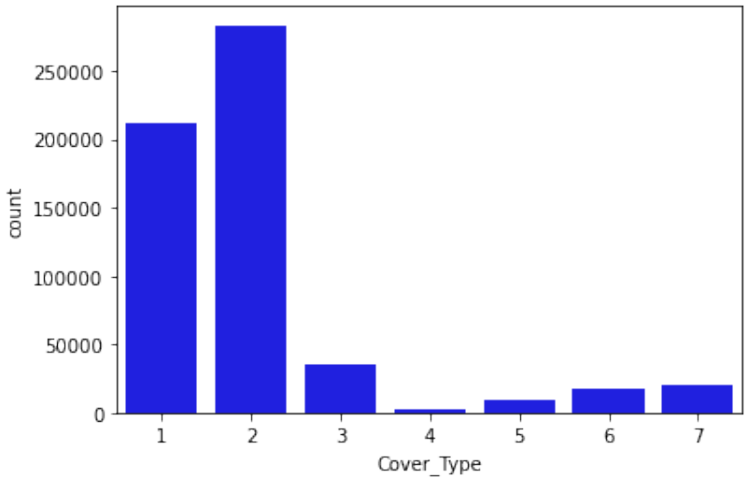
Content

1 Exploratory data analysis

Attributes used in modeling which are in starting set are:

```
Index(['Elevation', 'Aspect', 'Slope', 'Horizontal_Distance_To_Hydrology',
       'Vertical_Distance_To_Hydrology', 'Horizontal_Distance_To_Roadways',
       'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm',
       'Horizontal_Distance_To_Fire_Points', 'Wilderness_Area1',
       'Wilderness_Area2', 'Wilderness_Area3', 'Wilderness_Area4',
       'Soil_Type1', 'Soil_Type2', 'Soil_Type3', 'Soil_Type4', 'Soil_Type5',
       'Soil_Type6', 'Soil_Type7', 'Soil_Type8', 'Soil_Type9', 'Soil_Type10',
       'Soil_Type11', 'Soil_Type12', 'Soil_Type13', 'Soil_Type14',
       'Soil_Type15', 'Soil_Type16', 'Soil_Type17', 'Soil_Type18',
       'Soil_Type19', 'Soil_Type20', 'Soil_Type21', 'Soil_Type22',
       'Soil_Type23', 'Soil_Type24', 'Soil_Type25', 'Soil_Type26',
       'Soil_Type27', 'Soil_Type28', 'Soil_Type29', 'Soil_Type30',
       'Soil_Type31', 'Soil_Type32', 'Soil_Type33', 'Soil_Type34',
       'Soil_Type35', 'Soil_Type36', 'Soil_Type37', 'Soil_Type38',
       'Soil_Type39', 'Soil_Type40', 'Cover_Type'],
      dtype='object')
```

In data analysis is noticed that there are 7 different target classes which are not equally represented in the set:



We can see from plot above that classes 1 and 2 are more often. So we have to be careful during training process to not come into overfitting problem.

1.1 Outliers

Outliers are checked using box plot diagrams and there is not noticed anything strange or bad in data. All values for all attributes has sense.

1.2 Null values

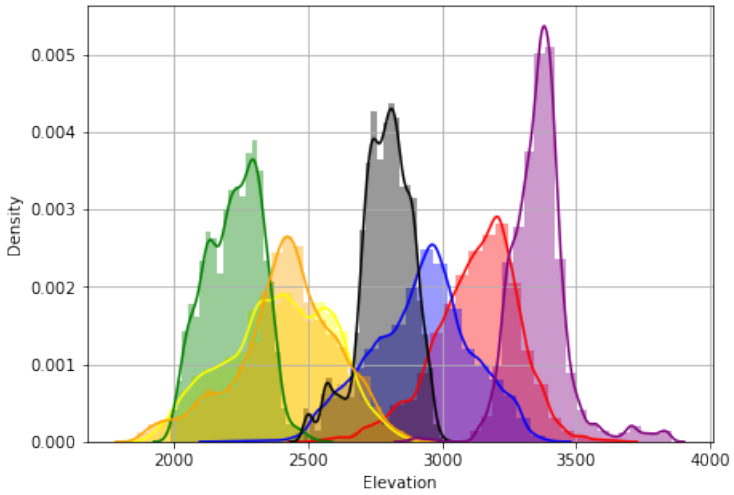
It is checked also does dataset contain null values for some attributes. Null values were not found.

1.3 Statistics

Also basic statistics is also checked, so in case that there is some attribute with variance 0 we can remove it. Nothing strange were not found in this section.

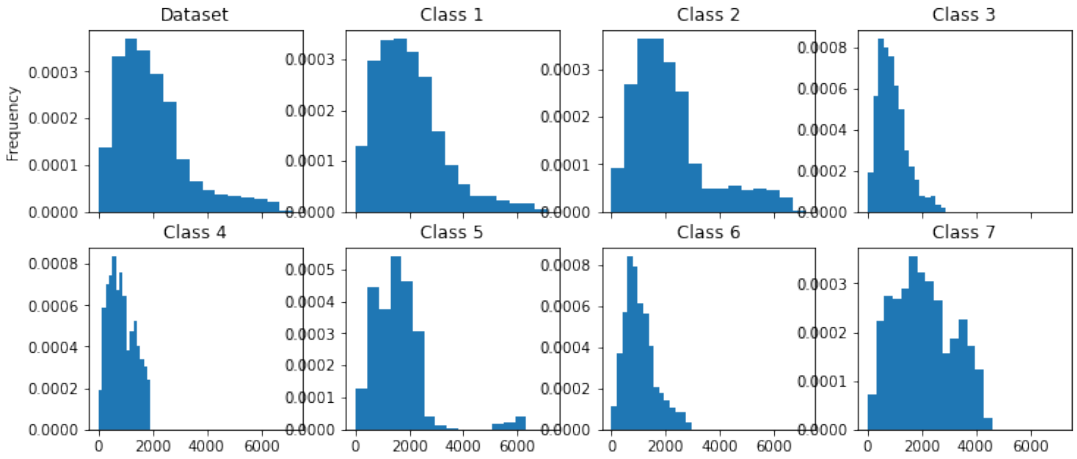
1.4 Distributions

Next checked thing are distributions per classes of each attribute. We noticed few interesting things.



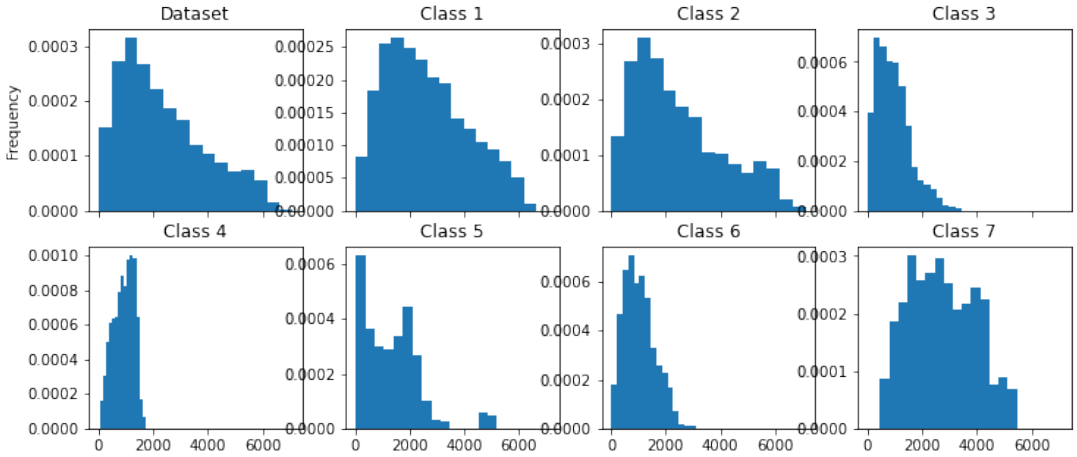
Attribute *Elevation* has very different distribution for each class. It can indicate that this attribute can be important in model prediction.

Next is about distribution per class for attribute *Horizontal distance to fire points*:



We can see here that each class has a little different distribution.

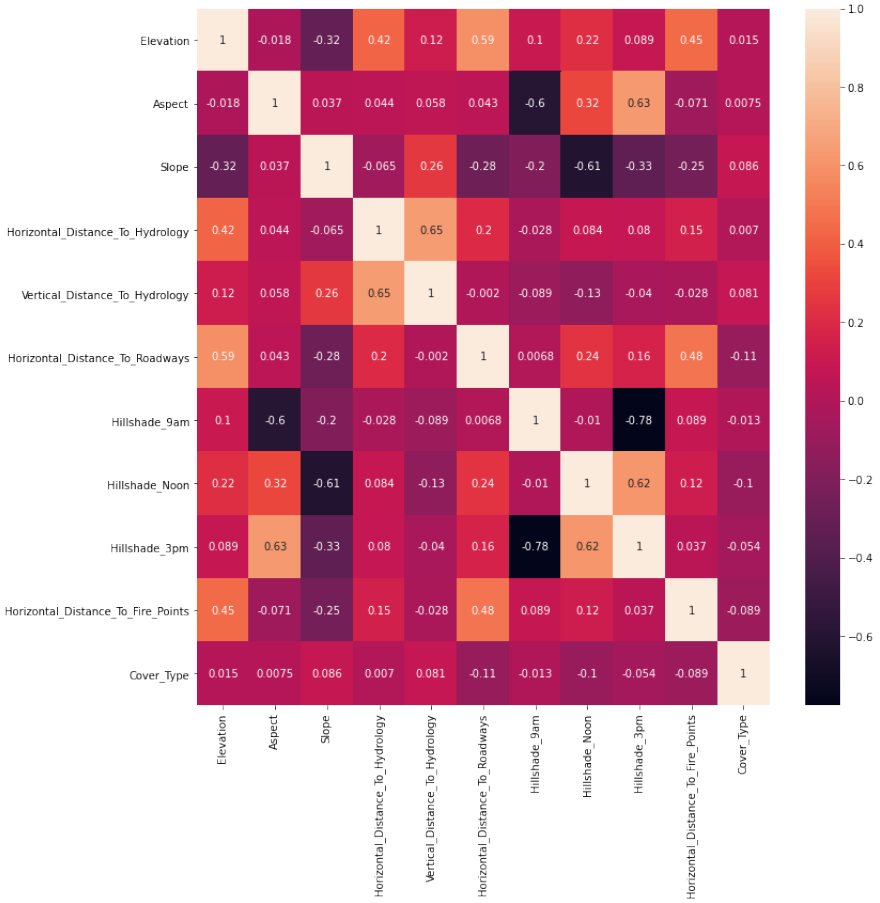
And next is for attribute *Horizontal distance to road*:



Where also we can see a little different distribution depending on the class.

1.5 Correlations

Correlations between attributes is also checked, how we can have a little better intuition what is going on there. Also attributes with very high correlation are not in same training process. Correlation plot:



1.6 Duplicates

It is also checked are duplicate rows exist, and they were not found.

2 Modeling

Given problem is problem of multi-class classification, how we have 7 different values for target attribute. We want to predict which tree class can grow in area with specific values for given attributes. So algorithms used for predictions are classification algorithms.

2.1 Preprocessing

Dataset is splitted on traning set and test set by sampling technique. All class in test set are equally represented. Test set is not used during training process, but for evaluation after model was trained to check models performance.

Next step was to preprocessed numerical attributes, ie. to normalize them. It is noticed that in dataset categorical attributes are not existed, ie. probably Soil_* and Wilderness_* were categorical and already preprocessed using one-hot-encoding technique.

For evaluation during training is used stratfied cross-validation technique with 10 iterations, and validation set 0.3 of whole train dataset.

2.2 Trening

In training process are used two algorithms: xgboost and SVM, so we can compared which of these two will give better results for given data.

2.3 Results

	XGBoost	**SVM**
Accuracy	0.74	0.68
P - macro avg	0.64	0.49
P - weighted avg	0.78	0.70
R - macro avg	0.88	0.61
R - weighted avg	0.74	0.68
F1 - macro avg	0.71	0.53
F1 - weighted avg	0.75	0.69

Table 1: Results on training set

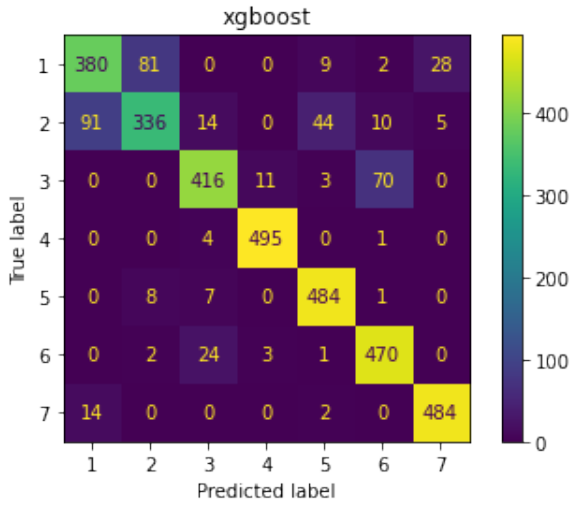
	XGBoost	SVM
Accuracy	0.88	0.60
P - macro avg	0.87	0.66
P - weighted avg	0.87	0.66
R - macro avg	0.88	0.60
R - weighted avg	0.88	0.60
F1 - macro avg	0.87	0.60
F1 - weighted avg	0.87	0.60

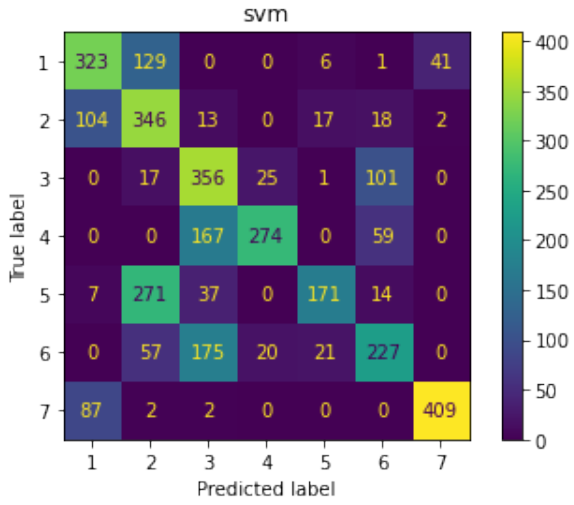
Table 2: Results on test set

How we know that we have imbalanced dataset we should be careful with result analysis. Using accuracy metric to decide which model is better could lead us in wrong direction. We usually use F1 score in that case.

How xgboost actually in background use random forest with few improvements, we decided to not use random forest for this case.

Let's check one more metric, confusion matrix, which will give us visual representation of model performance on each class:





From these two confusion matrix we see that xgboost model is a lot better that svm model, because we expect to see greater number on diagonal what is the case for xgboost but not for svm. So there is no doubt which of two model is better.

2.4 Further improvement

Some ideas for further improvement could be feature engineering (use existing features to find some feature which can be more correlated with target attribute), do some parameter tuning or try other classification algorithm.