**EECS738**

**Madhu Peduri (3006758)**

[**pmspraju@ku.edu**](mailto:pmspraju@ku.edu)

**Exploratory analysis of the dataset:**

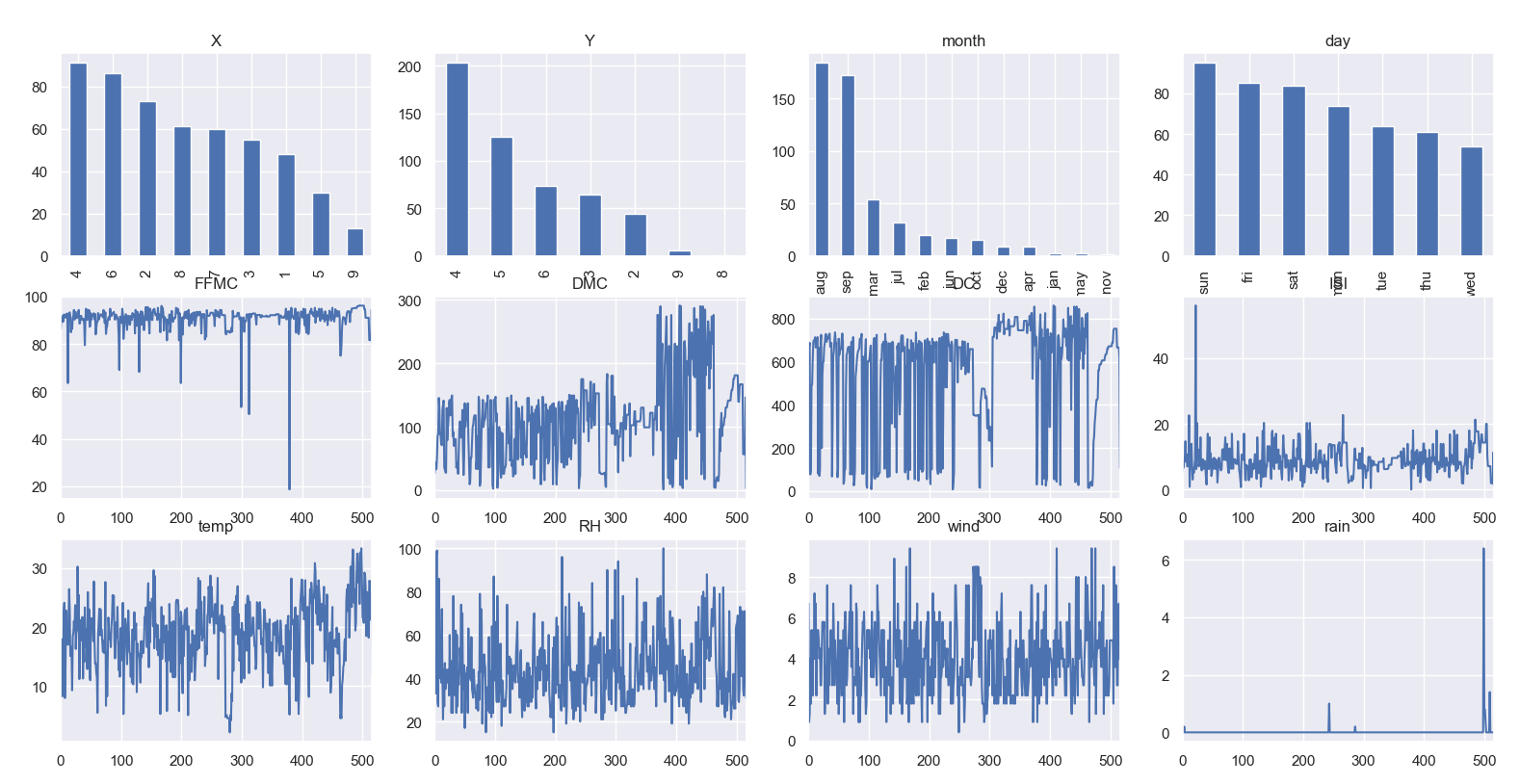
Total number of features: 13

Nominal features: X, Y.

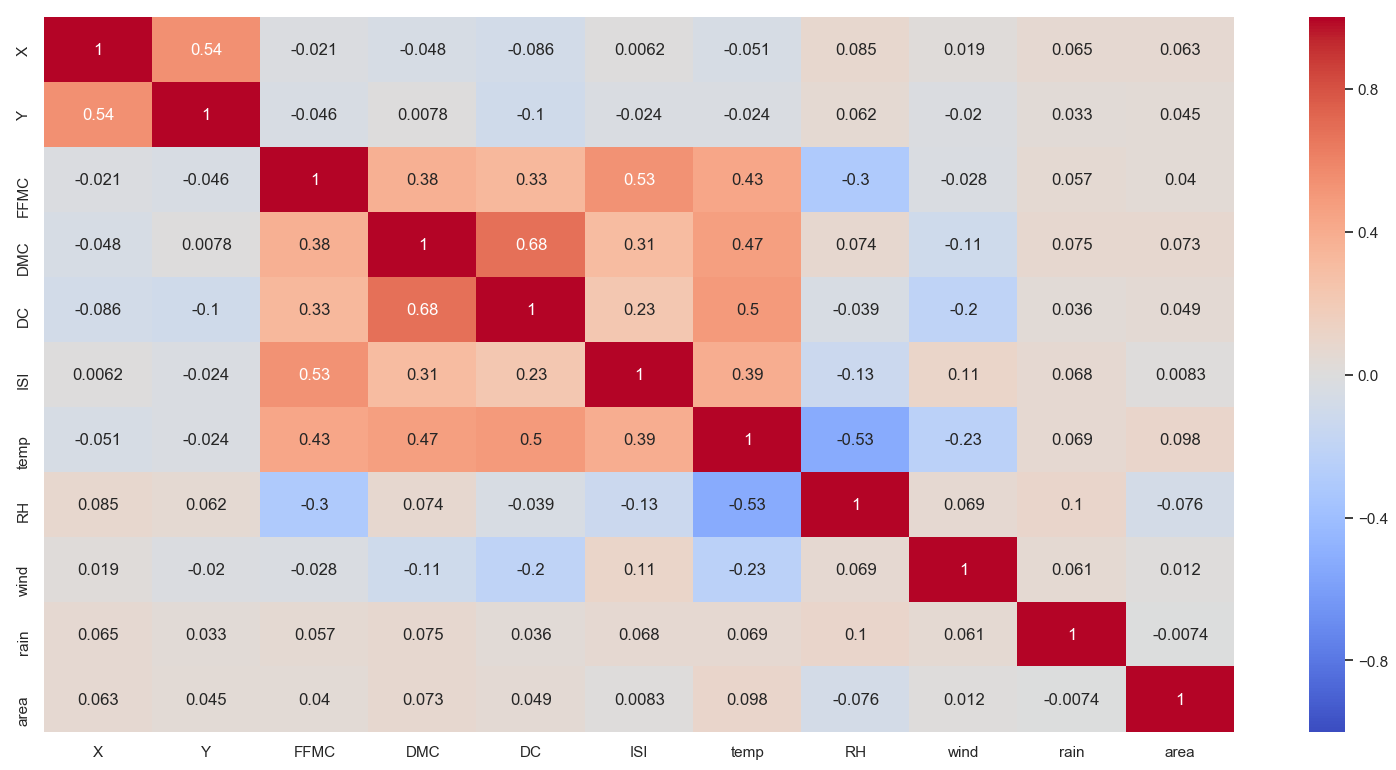
Categorical features: month, day

Numerical features: FFMC, DMC, DC, ISI, temp, RH, wind, rain

Target feature: area



* As X, Y are location of the forest areas, we can ignore these features from our model evaluation.
* We can observe from the above plot, that feature ‘rain’ is skewed and most of the values are zeros. We can ignore this feature from our model evaluation.
* All the numerical features are consistent and not much skewness is there.
* If we observe the target feature ‘area’, we can see that feature is not very unbalanced. It is a continuous feature having more zeros. In this case, zero is a valid value.
* Further we can see the heatmap showing the correlation between all the features.
* We can ignore the one of the features if they are highly correlated.



* We can see from the heatmap, only features that are having correlation DMC and DC with value 0.68. I tried by ignoring one of the features, but it is affecting the accuracy. I believe, 0.68 is not very high to be ignored.

**Pre- processing steps:**

* We can drop the columns X, Y, rain
* We have numeric features that are having different ranges. Using minmax scaler, we can bring them to common range.
* As our target feature is continuous and unbalanced. I used Decision tree, K-neighbors regressors but performance on r2 score is not good. To make it as a classification problem, target feature can be binarized with multiple labels. This makes multi-label classification.
* Below are the criteria I used to change our continuous target feature to multi – label feature

(target>0) & (target<=200) = 1

(target>200) & (target<=400) = 2

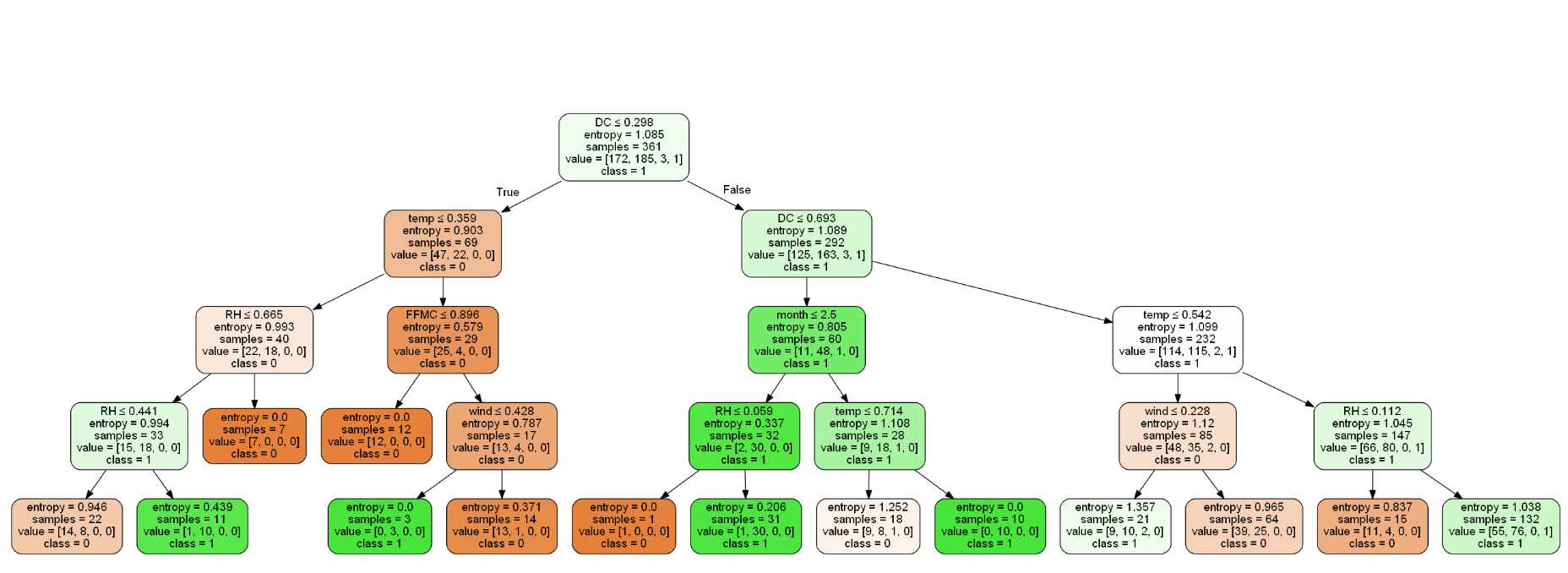
(target>400) & (target<=800) = 3

(target>800) = 4

* We have string features that are categorical. Using One-hot labeling, we changed string features into numeric features.
* Using above steps, we transform our data to make it readable by our classifiers and regressors.
* We used Grid search cross validation to implement K-Fold cross validation on our training dataset.

**Evaluation Metrics:**

* For classification, we used accuracy and f\_beta score. F\_beta score relies on precision and recall of the classification. Here we use beta = 1.5 favoring recall, because false negatives are more important than false positives for our data.
* For regression, we use r2\_score and evaluation metric.

**Results:**

Metrics:

Accuracy for Decision tree Classifier - Training, Test sets: 0.67867, 0.51282

R2 score for Decision tree regression -Training, Test sets: 0.21521, -0.29439

Accuracy for K-Neighbors Classifier-Training, Test sets: 0.63989, 0.56410

R2 score for K Neighbhors regression -Training, Test sets: 0.94427, -1.07405

Conclusion:

* As our target feature is unbalanced and continuous, regression model did not perform very well. Classification model performed better than regressors.
* K-nearest neighbors performed slightly better on our test dataset when compared to decision tree classification.