Deep Learning Course Project - Mushrooms

Reut Caspi Werber

August 2021

1 Abstract

This report explains the methods used to classify mushrooms by odor. The methods that were used are K-means and Multi Layer Perceptron. Also, Data Tree was used to learn more about the data. The best accuracy was achieved by the neural network model, and is approximately 85%. Reducing dimensions had no effect on the results. The python code for the models can be found at: https://github.com/reutwerber/DeepLearningHW

2 Introduction

Mushrooms has 20 different attributes (excluding being edible or poisonous), one of which is the odor. It is clear that the odor has great importance when deciding whether a mushroom is edible or not, so giving that attribute up is a pretty foolish move. Since the new researchers are anosmics it is important to "predict" the odor of a new found mushroom using the other attributes of said mushroom.

Removed veil-type cause it is always 0, removed musty because there is only one sample of this odor.

3 Methods

3.1 Clustering

Clustering is dividing the data into groups (clusters). The ideal clustering in this case will be dividing the data into 9 separate clusters, one for each odor: almond, anise, creosote, fishy, foul, musty, none, pungent and spicy. Ideally, when our model will encounter a new data sample (mushroom) it will classify it accurately to one of these odors. Since there is only 1 sample of a musty smell we will ignore it, and split the data into 8 clusters. To help visualize the results, PCA was used to lower the dimension to 2 dimensions. PCA was used since the important information about the data is it's the co-variance of the attributes(not the distances).

The clustering was done using K-Means algorithm.

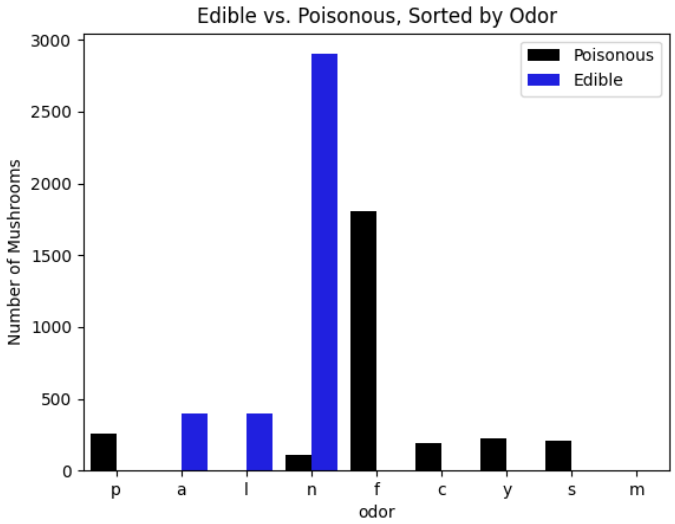


Figure 1: Count of poisonous and edible mushrooms, for each odor

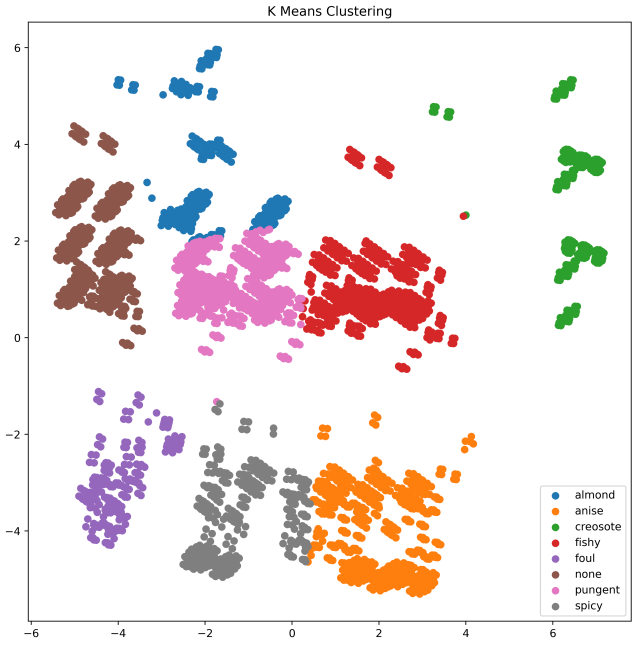


Figure 2: 8 Clusters after K Means

3.2 Neural Network

In order to build a machine, I built a Neural Network using the a Multi-layer Perceptron algorithm. The test **accuracy is around 86%**. The first layer is of 20 nodes, as the number of attributes. The output layer has 8 nodes.

3.3 Decision Tree and Data Manipulations

Using a decision tree we can calculate which attributes influence the result the least by calculating each feature's importance as seen in 3.

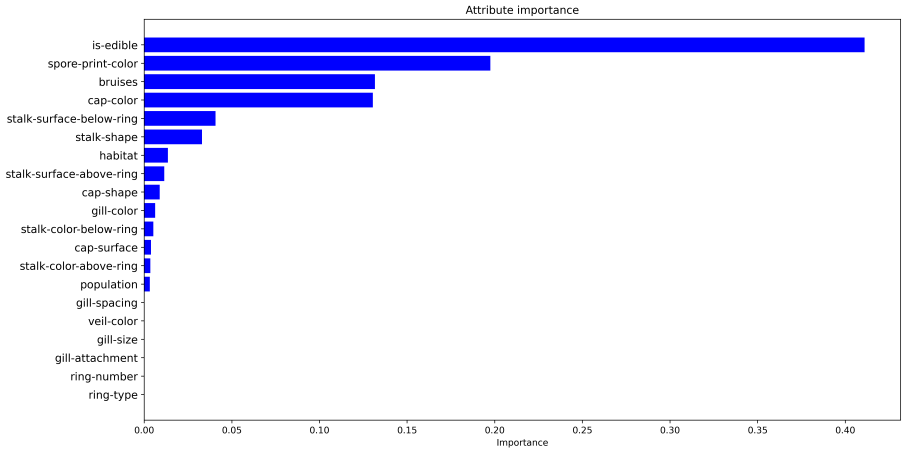
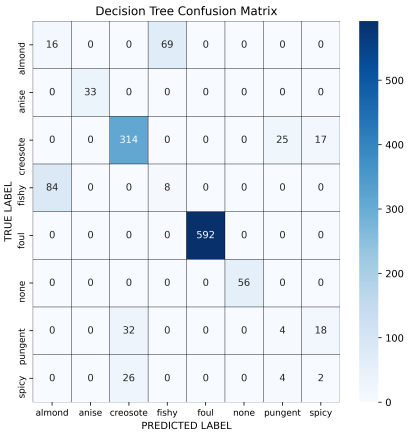


Figure 3: Attribute importance

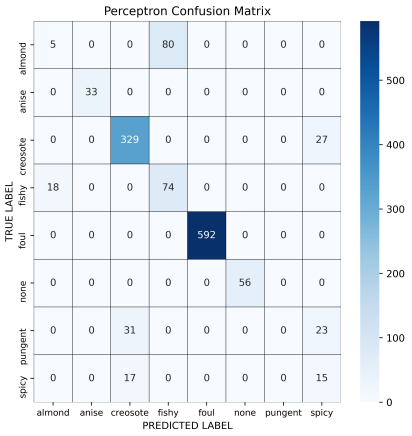
In order to find out which attributes has the most influence, we can also look at the co-variance matrix. The matrix is too big for this report, but it yields similar results (heatmap can be produced using the code referred in the abstract). The attributes with the least correlation with odor, has the most influence on the classification. The decision tree's test **accuracy is around 79%**. This is lower than the neural network's accuracy, but the tree gives us information about specific attributes and their importance: edible vs poisonous and the spore sprint color are the most important attributes. While we have the information whether a mushroom is edible or not, it is likely that when encountered with a new mushroom, this information won't be available and we will need to build a machine that will work without this attribute. The least important attributes are ring type, ring number, gill attachment, gill spacing, veil color, stalk shape and gill size.

Reducing the dimensions by removing these attributes caused no notable changes to the accuracy tests of the models. This is expected since the models "learn" to disregard these attributes. Nonetheless, to the human eye this may be beneficial.

Another important thing that can be noted about the attributes is that



(a) Decision Tree Confusion Matrix



(b) MLP confusion matrix

different attributes has different number of possibilities. For example, there are 10 cap colors, but only 2 types of gill size. Perhaps one hot encoding will represent the data better. Encoding this way, both with and without doing PCA on the encoded attributes,didn't have a major affect on the accuracy of the models. All in all, it seems that this method is not necessary.

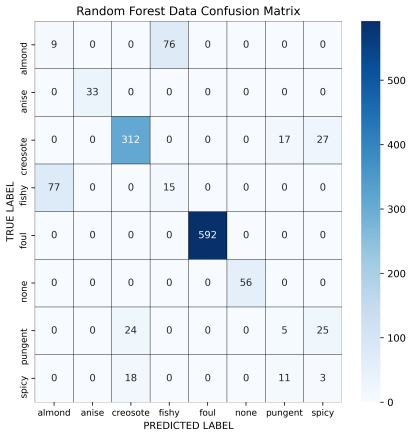
3.4 Anomaly Detection

Missing data is a type of an anomaly sample. If we use a semi supervised model we can use the existing tagging to classify the unknown data points. Running the data with no manipulations, AKA replacing the missing values with an arbitrary value in the python code, resulted in model accuracy of 46% (at best) for the MLP. This is obviously not good enough. Algorithms that are still reliable when there is missing data (or anomalies) are KNN and Random Forest. KNN yielded bad results for this data set and wasn't included in this report for that reason. Therefore, random forest is used to classify the samples with the missing data. The accuracy for the random forest model with the missing **is of around 77%.**

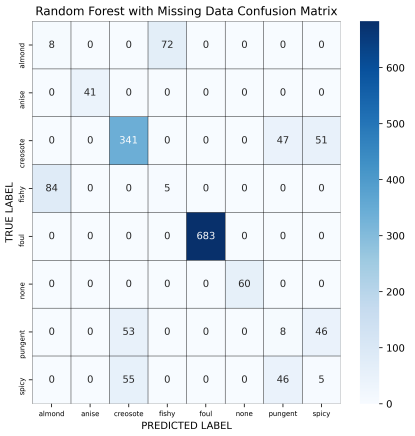
4 Results

The results show that no odor as well as anise, fishy, foul and odors can be detected with near 100% accuracy both with the random forest model and the perceptron models. Using a data tree yields results which are less accurate, though no odor, foul and anise odors can still be predicted with almost 100% accuracy.

The clustering model did not perform as well. There is no heat map for it, since it does not classify, but cluster. Nonetheless, the graph shows there is no



(a) Confusion matrix of random forest model, test vector did not include any samples missing data



(b) Confusion matrix of random forest model, test vector included samples with missing data

clear separation between clusters (2). The random forest model reacts well with missing data, as can be seen from the heatmaps and the test accuracy, which is around 77% for both cases, as can be seen in 5b.

5 Conclusions

The best outcomes were reached with the Multi-layer Perceptron model. Changing the data to reflect more meaningful attributes did not affect the results, and neither did using missing data.