AI Final Project

Amit Kabya

ID: 213828858

August 12th 2021

1 Introduction

Kfir is a fan of collecting mushrooms in nature in his free time. Since he is an orderly researcher, he classified each mushrooms according to the families and gave each mushrooms general characteristics (size, smell ....).

Kfir disappeared and left a collection of mushrooms to classify. The data is in the file - "mushrooms_data.txt".

Kfir's three assistants (Ofek, Haim and Lior) who are anosmic, compete for the role that was surprisingly vacant, but they did not have the vast knowledge he had. Therefore, Kfir's three assistants suggested three possible approaches to the data, with the labeling being according to the odor feature:

- First approach: Cluster the data and see the match between each cluster for labeling the mushrooms.
- Second approach: build a machine that learns based on the characteristics of the mushroom's family.
- The third assistant said that there is first a need to build more meaningful characteristics and only then use the implementation of the approaches of the other two assistants.

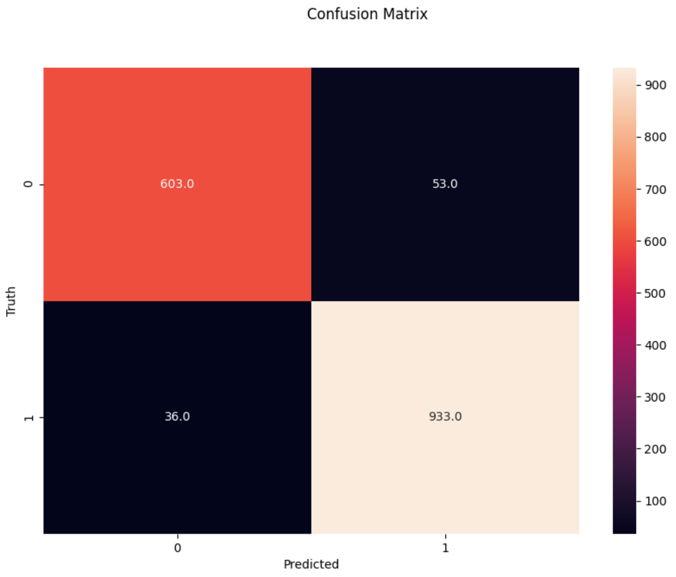
1.1 Introduction To The Problem

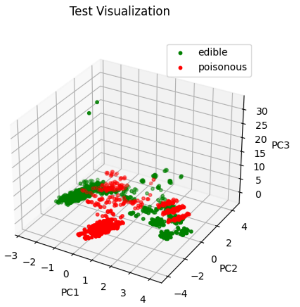
The problem above requires me to use three different approaches of AI and ML in order to solve it.

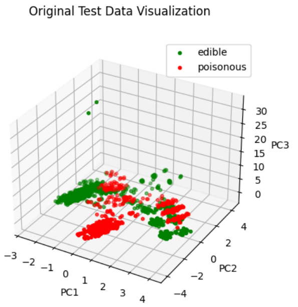
1. **The first approach is Unsupervised Learning.** We are meant to cluster the data into 2 clusters and determine which cluster is poisonous and which is edible.
2. **The second approach is Supervised Learning.** We are meant to build a learning machine that will predict weather a mushroom is poisonous or edible.
3. **The third approach is Dimension Reduction.** We are meant to reduce the data's dimensions in order to made the first 2 approaches to be more efficient in time and predict more accurately.

2 First Approach

In the first approach I chose to cluster with SVC Algorithm, which is similar to SVM, but is used for clustering. I chose this algorithm because I had to divide the data into 2 clusters - poisonous and edible. Therefore, an algorithm that cluster the data into 2 cluster exactly would be the most accurate.







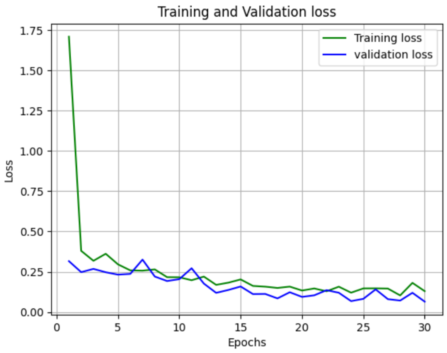
After a run of the SVC algorithm on our data we get an Accuracy of about 94%, which matches the confusion matrix and the visualization of the results. After getting the results I used PCA algorithm to reduce dimensions in order to visualize the data.

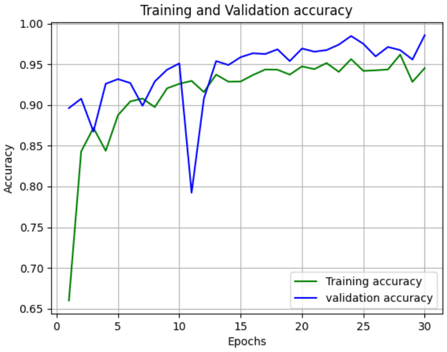
3 Second Approach

In the second approach I built a Neural Network using the TensorFlow package.

The Network has 2 hidden layers with 128 neurons in them. I used the Relu activation function, and a Binary Cross Entropy loss function. I split the data into 3 sets: test set, train set, validation set. I wanted the network to have the best results but train a fair amount of time, so I chose it to do 30 epochs that run on a batch size of 32 mushrooms.

After a run of the training the NN using the training data we get an Accuracy of about 98.6%.



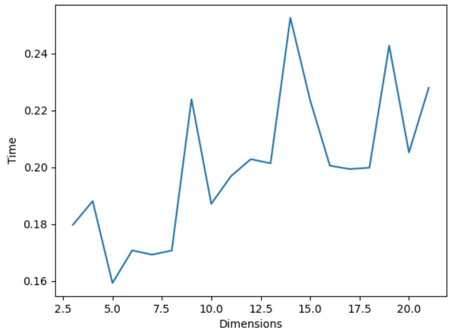


We can see that in both the loss progression graph, and the accuracy progression graph. the training data graph is similar to the validation data graph which indicates that overfitting is not likely

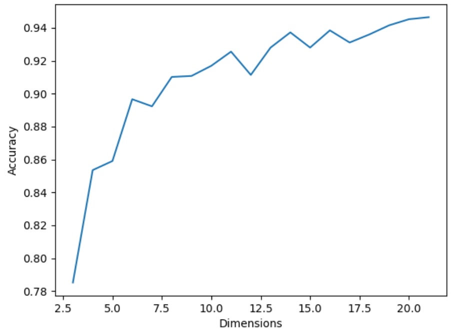
4 Third Approach

In the third approach I used the PCA algorithm, but instead of using it for visualization, like in the first approach, after I already have the results, now I will use it before classifying the data, and than classify the data.

The advantage of this method is that it is less likely to overfit models because we use less dimensions.
In addition, by reducing the data's dimensions, we have less data to store and any algorithm we will run on it will be faster.



(a) Dimension VS Time: SVC

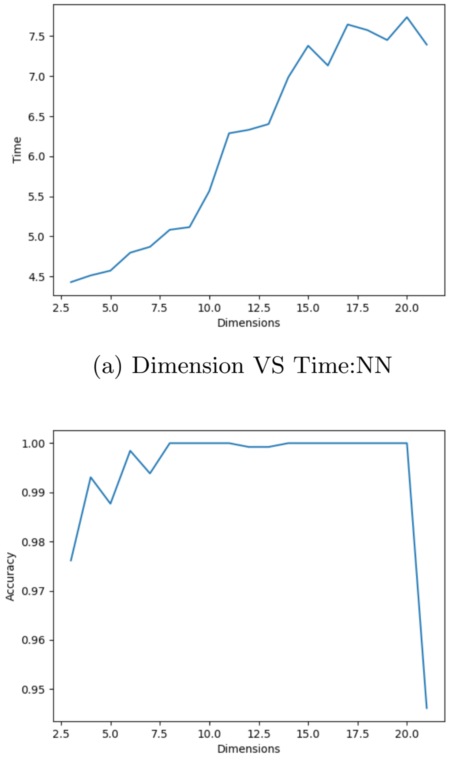


(b) Dimension VS Accuracy: SVC

In the graphs above we can see a run of SVC algorithm and the affect that different dimensions had on the running time, and the accuracy of the algorithm. We detect that for example reducing the dimensions by 5, to 16 dimensions can cause a significant amount of save in time

but a miniature reduction of accuracy relatively: $\frac{0.9464-0.9384}{0.9464}$ $=$ **0.00845308537 VS** $\frac{0.2297-0.2018}{0.2297}$ $=$ **0.12146277753**

In this example, the relative reduction ratio to 16 dimensions $\frac{relativediffinTime}{relativediffinAccuracy}$ **is** $\frac{0.12146277753}{0.00845308537}$ $=$ **14.369,**



(a) Dimension VS Time:NN

(b) Dimension VS Accuracy:NN

When using the PCA before running the NN, we get that the network trains even better when there are less dimensions up to a point where we get accuracy of 1. Again, the cost of Time VS Accuracy is great, and worth the dimension reduction

5 Conclusions

In conclusion, after examining all three approaches I can confidently say that the third one, is the best one!
NN made better predictions on the data after training on it then SVC. Even after reducing the data's dimensions, the accuracy of the model raised and we got that a hundred percent of the predictions made on the test set were correct.

6 Missing Data

Eventually, we got mushrooms that some of their data is deleted. We should help Lior to classify the mushrooms and tell how accurate we are.

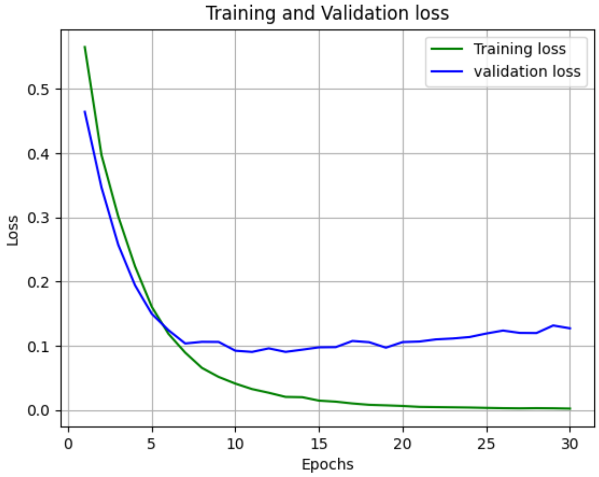
After reaching the conclusion that the third approach gives us the best results, we just have to modify the data and the algorithm to match the new data and give us the model accuracy.

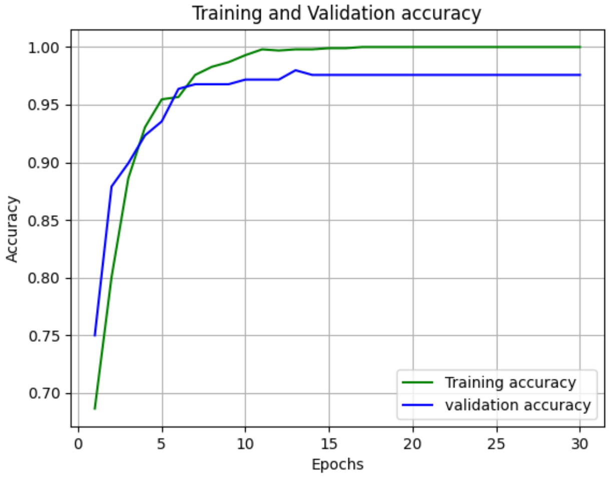
My solution is to divide the mushrooms into 2 sets:
1. Mushrooms that their classification is known.
2. Mushrooms that their classification was deleted.

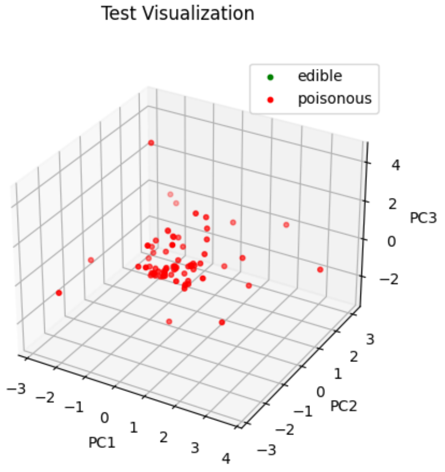
Where there is missing data, I will put (-1) and then run the third and second approaches on mushroom set 1. Having a classification is required in order to train the model and calculate its accuracy. After training the model and test it to get its accuracy using mushroom set 1, I will predict the classes of the mushrooms from set 2 using the trained model.

In order to get the best accuracy and according to the test I have made in the third approach, I reduced the data's dimensions to 16 and ran the NN with it, like I mentioned above. After training the NN it had accuracy of 97.4%

Like before, we can see that in both the loss progression graph, and the accuracy progression graph. the training data graph is similar to the validation data graph, so overfitting is not likely.
The graph at the bottom is the visualization of the prediction the NN had for the mushrooms in set 2.


Training and Validation loss
Training loss
validation loss
Loss
0.0
0.1
0.2
0.3
0.4
0.5
0
5
10
15
20
25
30
Epochs


Training and Validation accuracy
Training accuracy
validation accuracy
Accuracy
0.70
0.75
0.80
0.85
0.90
0.95
1.00
0
5
10
15
20
25
30
Epochs


Test Visualization
edible
poisonous
PC1
−3
−2
−1
0
1
2
3
4
PC2
−3
−2
−1
0
1
2
3
PC3
−2
0
2
4