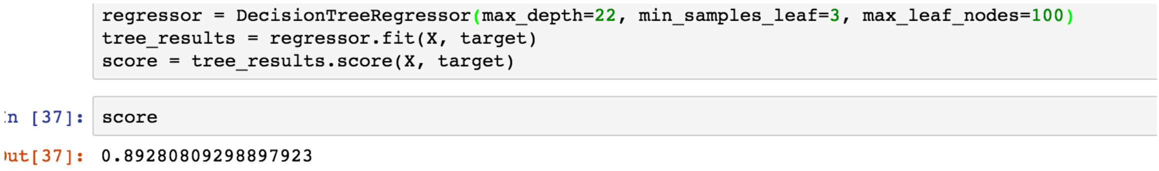
Dataset Conclusion

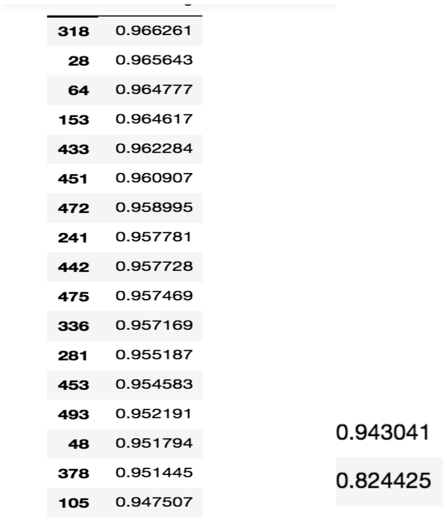
The madelon dataset that had the 500 features created a best benchmark score value of .89. This means that the Decision Tree Regressor generated the highest accuracy of fitting to the data points.



```
regressor = DecisionTreeRegressor(max_depth=22, min_samples_leaf=3, max_leaf_nodes=100)
tree_results = regressor.fit(X, target)
score = tree_results.score(X, target)
```

```
n [37]: score
ut[37]: 0.89280809298897923
```

The reason for performing the benchmark models was to be able to get an idea about which model could potential perform better with other data. This allows for me to make sure I chose the best model to score the way to get the features. Looking at the below pictures below show the specific features that have the most importance.



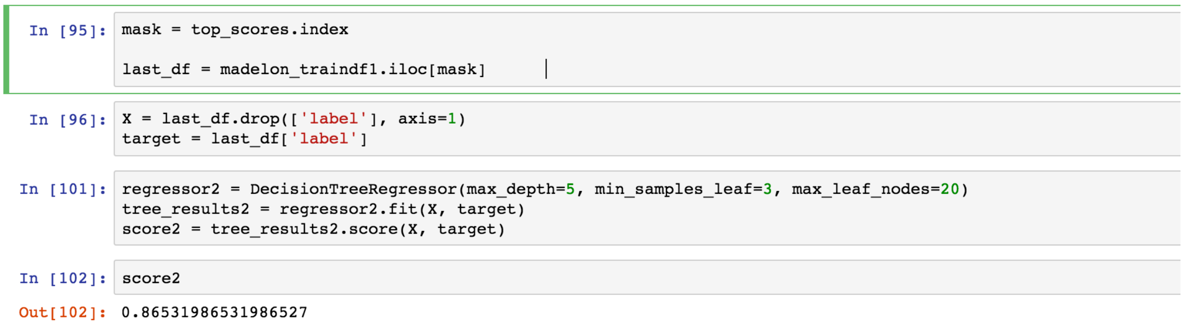
318	0.966261
28	0.965643
64	0.964777
153	0.964617
433	0.962284
451	0.960907
472	0.958995
241	0.957781
442	0.957728
475	0.957469
336	0.957169
281	0.955187
453	0.954583
493	0.952191
48	0.951794
378	0.951445
105	0.947507

0.943041

0.824425

Theses 20 features had the best score. After these top 20, score went negative. I then took the top 20 features and generated dataset with only those features and the target to run the DecisionTreeRegressor model with just those features. I used since it had scored the highest in the benchmark test.

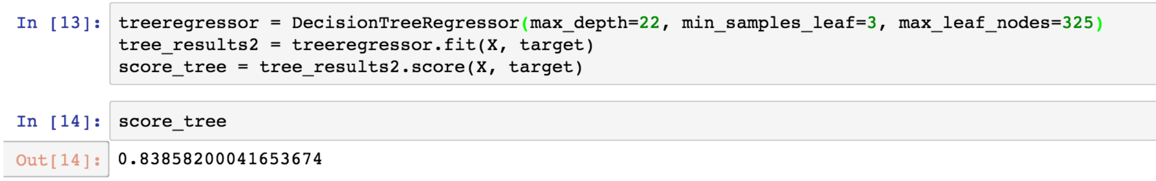
The results were that with these top features using the DecisionTreeRegressor model, changing some parameters in order to make sure the results are the most accurate. I had to generate a subset dataframe to run the



```
In [95]: mask = top_scores.index

         last_df = madelon_traindf1.iloc[mask]

In [96]: X = last_df.drop(['label'], axis=1)
         target = last_df['label']

In [101]: regressor2 = DecisionTreeRegressor(max_depth=5, min_samples_leaf=3, max_leaf_nodes=20)
          tree_results2 = regressor2.fit(X, target)
          score2 = tree_results2.score(X, target)

In [102]: score2

Out[102]: 0.86531986531986527
```

features in the model. I was able to get a score of .86 or 86%

For the bigger data set, I followed similar steps. I also took a sample of 6,506 data points to test because based on the overall data, with only a one percent margin of error and 90 percent confidence interval, this would give us the highest scoring sample amount. I also created a benchmark model test of different models to see how they performed. The DecisionTreeRegressor once again performed the best as a benchmark test. In order to select the top features, I used SelectKBest. These results gave the following top features: 271, 683, 703, and 922. I then also took these features to run in the DecisionTreeModel, since it was the best performing benchmark model, to see how well just those features score in selecting those features. I also set parameters to try to help make sure that the parameters provided a stronger result. The result gave us an 83.9 percent score for select the important features.



```
In [13]: treeregressor = DecisionTreeRegressor(max_depth=22, min_samples_leaf=3, max_leaf_nodes=325)
         tree_results2 = treeregressor.fit(X, target)
         score_tree = tree_results2.score(X, target)

In [14]: score_tree

Out[14]: 0.83858200041653674
```