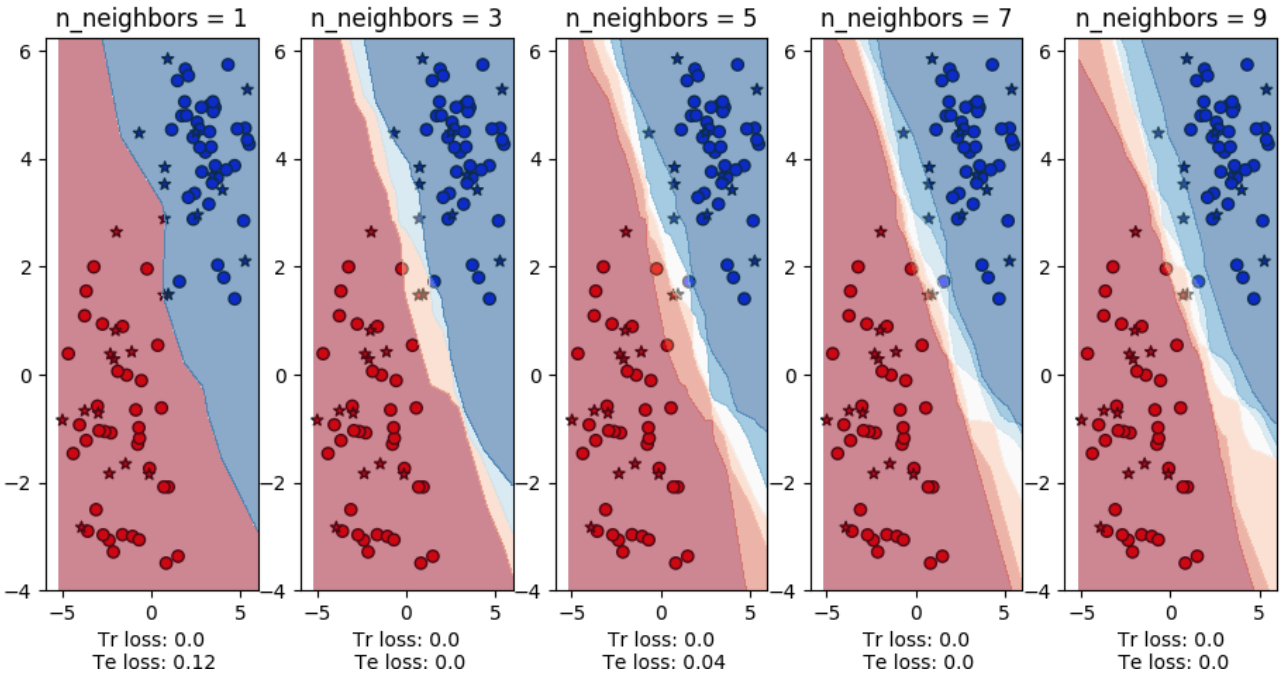
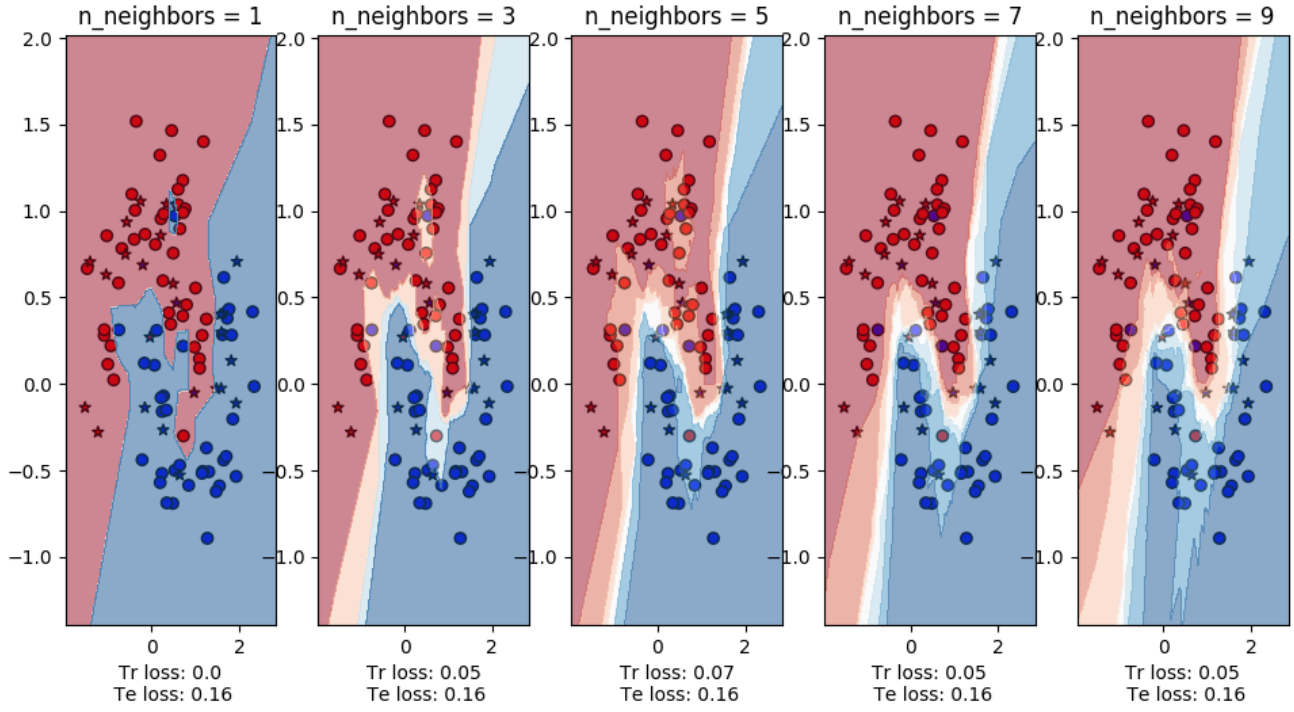
Rob Schwartz
Project 3 PDF

Part 1:

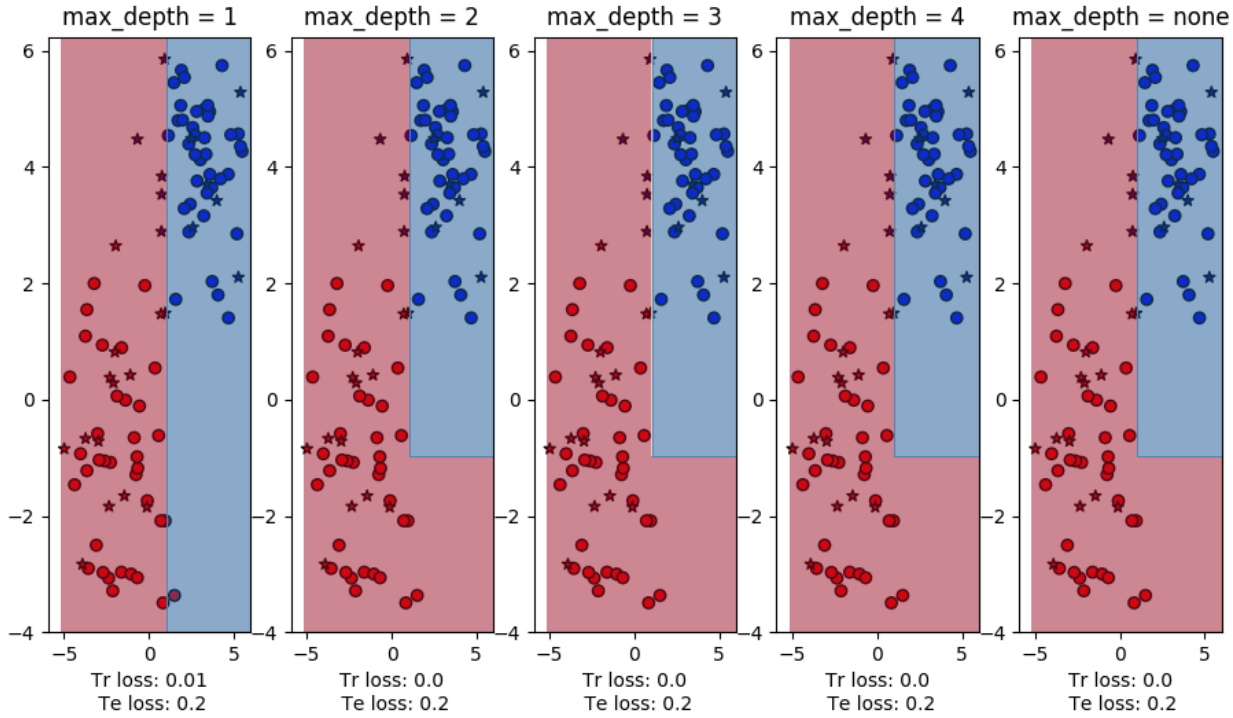
Simple Task kNN



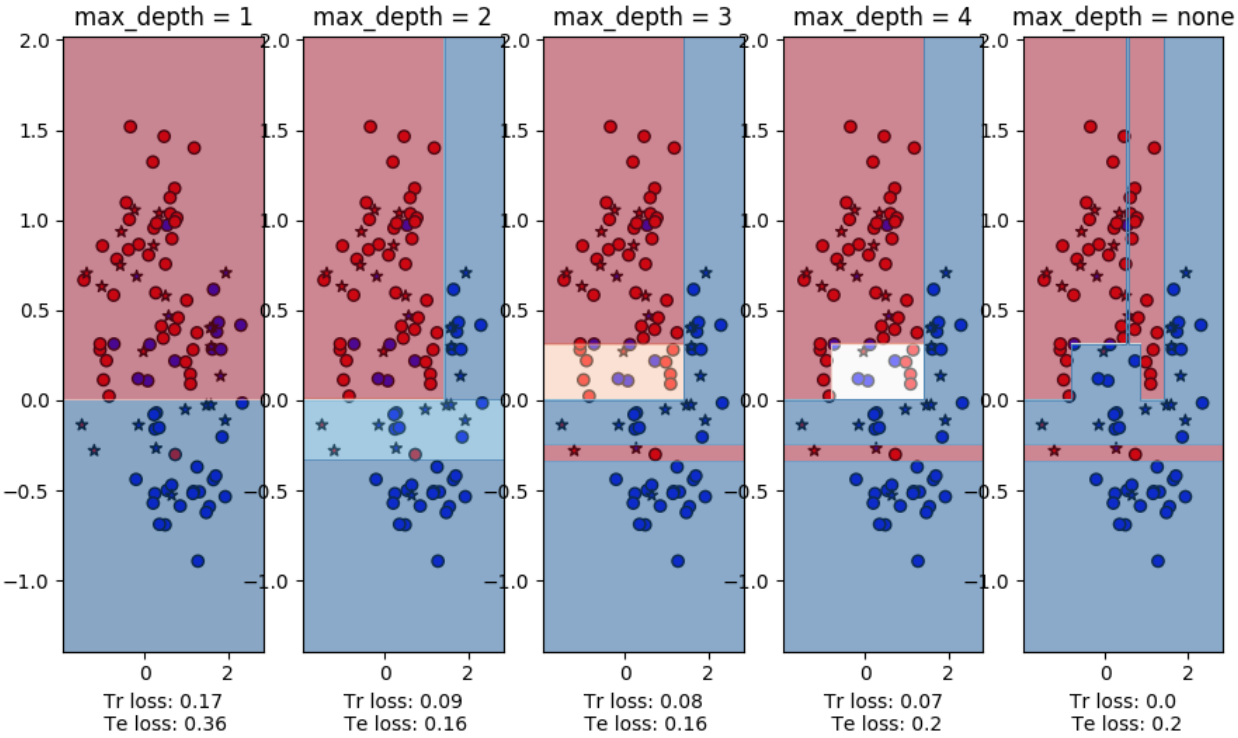
Moons kNN



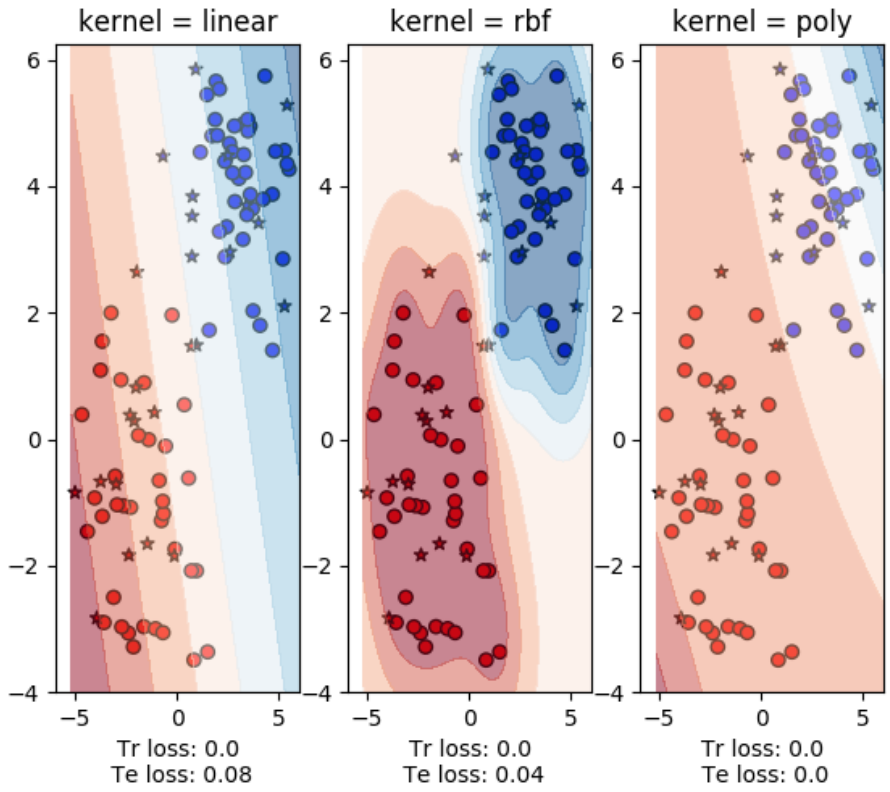
Simple Task Decision Trees



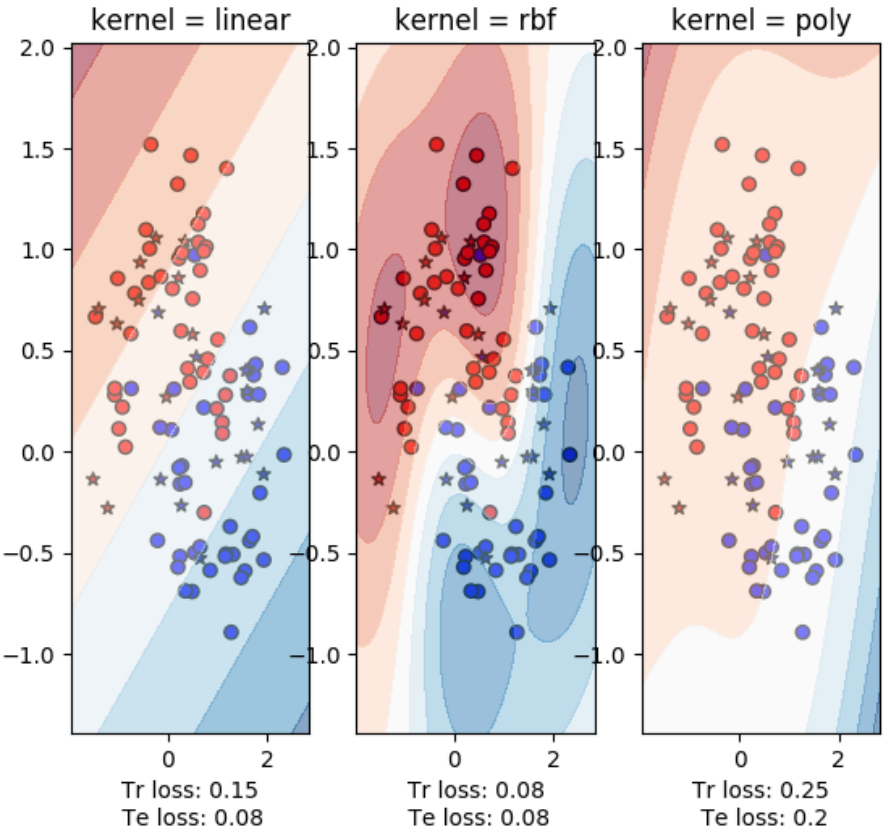
Moons Decision Trees



Simple Task SVM



Moons SVM



Q4:

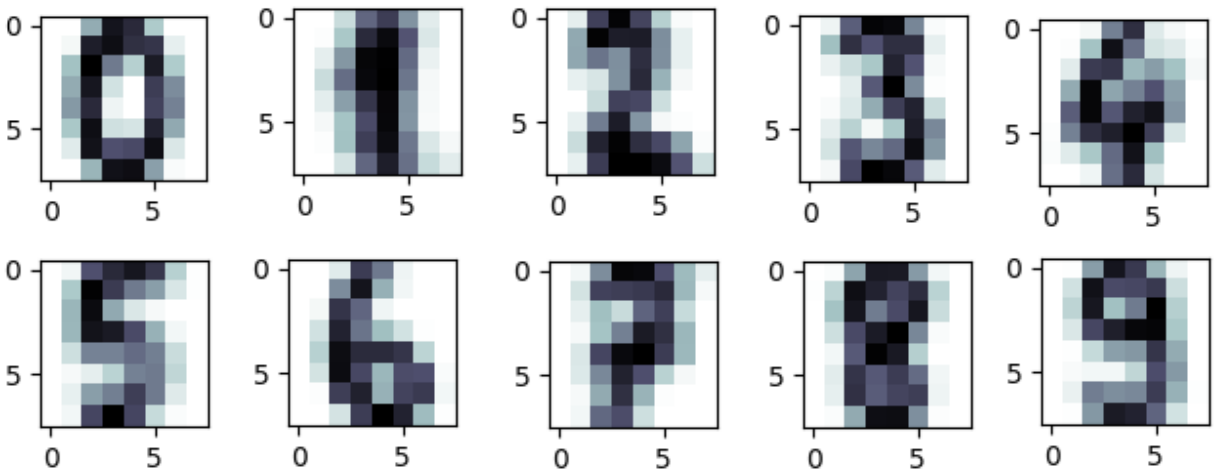
Unfortunately, I wasn't able to determine a data set, as I was unable to perfect my data-gathering technique. However, I intended to cluster all 1 points in one corner and all 0 points outside of that one corner, which would have worked well with an unlimited-depth decision tree model. I imagine that if a few features pointed to the data being in that "1" corner, the model would have easily classified the data as such.

I ran into issues when trying to add data around certain coordinates. I have kept your model so that the code runs properly. You can find my tests included in the file.

Part 2:

Q1. a. There are 21 0's.
There are 35 1's.
There are 31 2's.
There are 28 3's.
There are 39 4's.
There are 33 5's.
There are 28 6's.
There are 30 7's.
There are 28 8's.
There are 27 9's.

b.



c. Only features that do not help the model discern one image from the rest might be useless. Therefore, I expect that the features that represent corners, considering they are largely identical, would be useless for the model.

Q2:

Confusion Matrix:

```
[[155   0   1   0   0   0   0   0   1   0]
 [  0 140   0   0   1   2   0   0   4   0]
 [  0   2 135   7   0   0   0   0   2   0]
 [  0   0   0 133   0   3   1   2  10   6]
 [  0   5   0   0 134   0   0   1   2   0]
 [  0   1   0   2   1 142   1   1   0   1]
 [  0   1   0   0   1   0 142   0   9   0]
 [  0   0   0   0   0   0   0 146   1   2]
 [  1  11   0   1   3   5  12   1 106   6]
 [  0   7   0   9   2   1   0   0   5 129]]
```

Precision/Recall Scores

	precision	recall	f1-score	support
0	0.99	0.99	0.99	157
1	0.84	0.95	0.89	147
2	0.99	0.92	0.96	146
3	0.88	0.86	0.87	155
4	0.94	0.94	0.94	142
5	0.93	0.95	0.94	149
6	0.91	0.93	0.92	153
7	0.97	0.98	0.97	149
8	0.76	0.73	0.74	146
9	0.90	0.84	0.87	153
micro avg	0.91	0.91	0.91	1497
macro avg	0.91	0.91	0.91	1497
weighted avg	0.91	0.91	0.91	1497

Q3:

<u>Untransformed Data Confusion Matrix</u>

```
[ [ 91 11]
  [  5  162] ]
```

<u>Transformed Data Confusion Matrix</u>

```
[ [ 100 2 ]
  [   6  161] ]
```

1. Train kNN classifier with training data.
2. Predict test results using test data and trained kNN classifier.
3. Output confusion matrix with inputs of: true test results and predicted test results.
4. Fit training data to MinMaxScaler.
5. Transform training and testing data using scaler.
6. Repeat steps 1-3, using transformed training and test data instead of initial training and test data.

Q4:

The best learner was:

kernel - poly
degree - 2
C - 1000

Confusion Matrix:

```
[ [ 100  2]
  [   1  166] ]
```

Scores:

	precision	recall	f1-score	support
0	0.99	0.98	0.99	102
1	0.99	0.99	0.99	167
micro avg	0.99	0.99	0.99	269
macro avg	0.99	0.99	0.99	269
weighted avg	0.99	0.99	0.99	269