Logistic Regression Scores on Scaled Data:

	Train Score	Test Score
C=10^-3	0.777647	0.781690
C=10^-2	0.777647	0.781690
C=10^-1	0.834706	0.859155
C=10^0	0.887647	0.887324
C=10^1	0.890588	0.875587
C=10^2	0.889412	0.873239
C=10^3	0.888235	0.873239

SVC Scores on Scaled Data:

	Train Score	Test Score
C=10^-3	0.777647	0.781690
C=10^-2	0.777647	0.781690
C=10^-1	0.848824	0.877934
C=10^0	0.887647	0.889671
C=10^1	0.890588	0.882629
C=10^2	0.890588	0.884977
C=10^3	0.891176	0.884977

The models shown here appear to be acting roughly the same. For both types of models there is a trend towards better performance when the value of C increases. It should be of note that convergence was an issue for higher values of C, but there still appears to be similar performance. In these high values of C there was an inability to converge, and as such it is likely that overfitting would increase if the model were to converge properly. Underfitting also can be seen in lower values of C for both SVC and Logistic regression, but it is also very slight. Overall a middle value of C=0 appears to work best in these models, with larger values of C coming close and smaller values of C performing worse.

Logistic Regression Scores on Polynomial Data:

	Train Score	Test Score
C=10^-3	0.777647	0.781690
C=10^-2	0.832353	0.852113
C=10^-1	0.895882	0.910798
C=10^0	0.912941	0.910798
C=10^1	0.927647	0.929577
C=10^2	0.942941	0.931925
C=10^3	0.953529	0.931925

SVC Scores on Polynomial Data:

	Train Score	Test Score
C=10^-3	0.777647	0.781690

C=10^-2	0.853529	0.877934
C=10^-1	0.900588	0.913146
C=10^0	0.920000	0.915493
C=10^1	0.932941	0.927230
C=10^2	0.950588	0.927230
C=10^3	0.913529	0.910798

Between all models, the highest performing model is Logistic Regression, with a C value of 10^3. Keep in mind that this model was also not able to converge properly. I believe that this data set has low irreducible error because performance is similar for all models, and that higher complexities are having good performance even when they can not converge. My assumption is based on the fact that the relationship present in the training sample quite easily maps to the testing data.

Decision Tree Scores on Scaled Data:

max_depth=2
min_samples_split=0.01 (0.8694117647058823, 0.8967136150234741)
min_samples_split=0.1   (0.8694117647058823, 0.8967136150234741)
min_samples_split=0.2   (0.8505882352941176, 0.8732394366197183)

max_depth=4
min_samples_split=0.01 (0.9088235294117647, 0.9272300469483568)
min_samples_split=0.1   (0.8994117647058824, 0.9225352112676056)
min_samples_split=0.2   (0.8511764705882353, 0.8708920187793427)

max_depth=8
min_samples_split=0.01 (0.9505882352941176, 0.9225352112676056)
min_samples_split=0.1   (0.9117647058823529, 0.9272300469483568)
min_samples_split=0.2   (0.8511764705882353, 0.8708920187793427)

max_depth=16
min_samples_split=0.01 (0.9564705882352941, 0.9154929577464789)
min_samples_split=0.1   (0.9117647058823529, 0.9272300469483568)
min_samples_split=0.2   (0.8511764705882353, 0.8708920187793427)

max_depth=None
min_samples_split=0.01 (0.9564705882352941, 0.9154929577464789)
min_samples_split=0.1   (0.9117647058823529, 0.9272300469483568)
min_samples_split=0.2   (0.8511764705882353, 0.8708920187793427)

The tree with the max_depth of 4 and min samples split of 0.1 appears to be acting the best. Also, the model appears to work better by increasing the max_depth until the point of 8. In the trees above, there does not appear to be much overfitting or underfitting, similar to the models used in the previous tables.

Decision Tree Scores on Polynomial Data:
max_depth=2
min_samples_split=0.01 (0.861764705882353, 0.8802816901408451)
min_samples_split=0.1 (0.861764705882353, 0.8802816901408451)
min_samples_split=0.2 (0.861764705882353, 0.8802816901408451)

max_depth=4
min_samples_split=0.01 (0.9288235294117647, 0.9131455399061033)
min_samples_split=0.1 (0.9205882352941176, 0.9131455399061033)
min_samples_split=0.2 (0.8670588235294118, 0.8755868544600939)

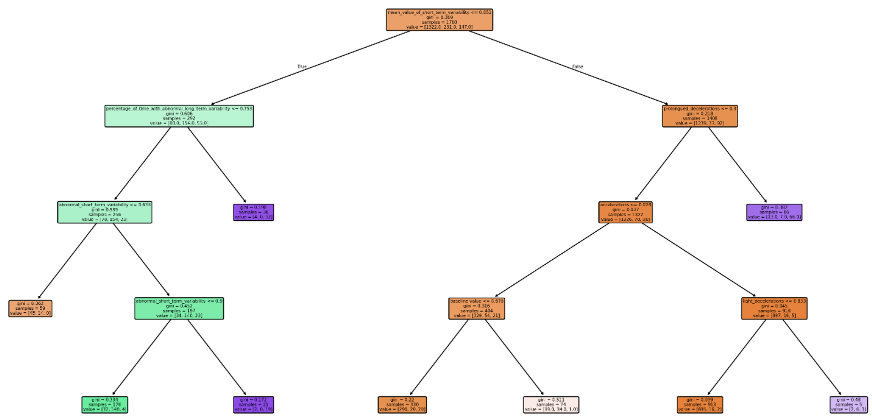
max_depth=8
min_samples_split=0.01 (0.9588235294117647, 0.9084507042253521)
min_samples_split=0.1 (0.9311764705882353, 0.9178403755868545)
min_samples_split=0.2 (0.8717647058823529, 0.8732394366197183)

max_depth=16
min_samples_split=0.01 (0.9688235294117648, 0.9061032863849765)
min_samples_split=0.1 (0.9323529411764706, 0.9154929577464789)
min_samples_split=0.2 (0.8729411764705882, 0.8755868544600939)

max_depth=None
min_samples_split=0.01 (0.97, 0.9014084507042254)
min_samples_split=0.1 (0.9323529411764706, 0.9178403755868545)
min_samples_split=0.2 (0.8723529411764706, 0.8708920187793427)

The polynomial features have had little effect on the performance of the model, except for perhaps a very slight overfitting. The best performing model out of all of these trees is the model training on the normal scaled features, with a max depth of 8 or higher, and a min samples split of 0.1.

Best Decision Tree with max_depth=4 (min_samples_split=0.01)



Best Decision Tree - Overall (min_samples_split=0.01, max_depth=4)