CS 7641 Assignment 1: Supervised Learning

Steel Ferguson
sferguson42@gatech.edu

1 INTRODUCTION

In this assignment I use 5 different machine learning algorithms to perform binary classification on two different data sets. This analysis covers why I chose the particular data sets, how I arrived at optimized models, the results from the optimized models, and a series of learning related to the algorithms and their parts. Learning and complexity curves are provided for each model on both data sets in the last section of this document in figures 1-22.

2 DATA SELECTION

2.1 Goal and Process

My goal in data selection is to analyze two data sets that provide interesting insights when analyzed together. Specifically, I sought data sets that would be sufficiently different in three ways: 1) relative strength of the five models, 2) one balanced (roughly equal rows in each category) and one unbalanced, and 3) and one set with many more rows than the other.

To find two data sets that would satisfy these, I created small bit of code fit the five models and give the relative accuracies. I compared this along with the balance and length of data sets, to find an "interesting" pair.

The relative accuracies (averaging to 100%) on the test data of the selected data sets on models before any tuning are as follows:

Relative Accuracy Before Tuning:

	Pima Diabetes	Amazon Reviews
Decision Tree	91%	134%
Neural Network	102%	82%
Boosting	102%	91%
SVM	104%	63%
KNN	100%	131%

2.2 Selected Datasets

2.2.1 *Pima Diabetes*

I selected a data set of medical predictor variables and the "outcome" of whether or not the patient has diabetes, from a group of Pima Native Americans. The data

is found on Kaggle[5]. The set is relatively balanced (500 / 768 or 65% with no diabetes) and relatively small (1268).

2.2.2 *Amazon Reviews*

I used Amazon reviews data from 2018[3] in the category of "Industrial and Scientific" in the provided smaller subsets of the full year data. For the target variable I created a "low score" indicator for scores of 1 and 2 (out of 5). I created multiple columns out of the review and title texts. I used positive and negative words lists [1,2] to find the number of each in the title and review texts. I also created variables for the length of the reviews, the number of sentences, and the number of exclamations. Finally, I created indicators for the presence of other key words such as "best", "worst", "absolutely", and "return".
The data is longer than the Pima Diabetes (37k rows instead of <2k) and unbalanced (6% "low score (4,418 / 77,071 which is ~6%).

3 EXPLORATION PROCESS

3.1 Goal and Defining "Better"

After selecting the databases, I set out to tune the models to achieve a better result. I started by defining "better." For the Pima Diabetes dataset, I felt comfortable using accuracy for my objective as the data set is balanced and accuracy provides a clear sense of the result I wanted (correct predictions).
However, with the unbalanced dataset, I choose to use the F1 score which harmonic mean of precision and recall. The reason for this is that using precision and recall as the inputs to the F1 function (instead of the raw number of mismatches) allow it to penalize incorrect prediction in the smaller category more heavily. This adequately accounts for the imbalance.

3.2 Grid Search

In order to find optimal hyperparameters, I leveraged Scikit-Learn's[4] cross validation grid search. I started with a wide grid for most parameters and ran a function I created around the Scikit-Learn grid search to record the rank ordered results. I ran the results over night as the multiple permutations (running brute force instead of random) along with the additional iterations for cross validation took significant time to compute. I used the previous night's results to refine the parameters in each grid. I did nearly a dozen iterations for each model and ended with an improved result. I also checked the most influential parameters (based on the grid search results) in complexity curves. Many of the complexity curves are shown in the figures at the end of this paper.

3.3 Functions

I created functions for initial exploration of the data (mentioned in the data selection section) and functions incorporating the Scikit-Learn's grid search to give results for each model on both data sets in a way that was easily accessed later

(mentioned in the grid search section). I also created function to create and save learning curves and complexity curves for parameters selected.

4 RESULTS

4.1 Change in Score

Tuning increased the score for all five models on both data sets.

Accuracy Increase from Tuning Models for Pima Diabetes Dataset:

	Untuned Accuracy	Tuned Accuracy	Change
Decision Tree	70.31%	71.35%	1.04%
Neural Network	75.69%	76.39%	0.70%
Boosting	75.00%	76.91%	1.91%
SVM	75.17%	78.30%	3.13%
KNN	72.74%	75.00%	2.26%

Accuracy Increase from Tuning Models for Amazon Reviews Dataset:

	Untuned F1	Tuned F1	Change
Decision Tree	30.28%	30.37%	0.09%
Neural Network	18.61%	34.09%	15.48%
Boosting	20.64%	27.94%	7.30%
SVM	14.18%	17.34%	3.16%
KNN	29.57%	33.44%	3.87%

4.1.1 *Overall Scores for The Two Data Sets*

The Pima Diabetes dataset has much more predictive information in its features than does the Amazon Reviews dataset. That is not readily apparent from the tables above as they use different scores, but the average F1 score on the Pima Diabetes dataset is 63% post tuning, compared to the 29% for Amazon Reviews.

After tuning the models, we can confirm that the significant gap in prediction in the Amazon Reviews is likely due to variance in the data that is actually not explained in the features. If I were tasked with improving the score for Amazon Reviews, I would recommend exploring new features.

4.1.2 *The Relative Strength On Each Data Set*

Here is where the datasets show off how they are interesting together. If we rank the tuned models for the two data sets, we get very different orders. SVM is a clear winner in the Pima Diabetes, and it is the clear loser in Amazon Reviews. SVM creates a boundary between subgroups of data. They are especially effective for data with a high number of features relative to the data. They are weaker with

data with high variance (mixing or crossing that boundary) and where data is unbalanced. Both of those weaknesses are present with the Amazon Review data.

The Neural Network performed well relative to other models for both datasets, but it ranks first for the Amazon Reviews and third for the Pima Diabetes dataset. This makes sense as Neural Networks are particularly good at explaining very complex data and correlations.

An interesting aside, not shown here, is that ranking the Pima Diabetes models by their F1 scores also changes their relative ranking. Neural Network become the highest score. That indicates that the Neural Network is better at predicting the smaller category (diabetes).

4.1.3 *The Relative Improvement For Each Model*

Neural Network had a 15% absolute increase in score from tuning making it clearly stand out. This derives largely from the complex nature of the model. The bulk of the difference came from restructuring the layers. Sciket-Learn starts with a defaulted one hidden layer with 100 nodes. The winning structure was 3 hidden layers with 100, 100, and 20 nodes. I was unable to find a similarly beneficial new structure for Pima Diabetes. It ended with one hidden layer of 10 nodes as the optimized structure, but this is not as large of a change.

SVM experience an absolute 3% increase in scores for both data sets. The Pima Diabetes reached this with the default kernel (rbf), but the Amazon Reviews data set reached this by using a polynomial kernel with 5 degrees. The large increase for both make sense as SVM is a more complex model that has more room to tune and optimize. Conversely we saw modest increase in scores from Decision Trees which came primarily from pruning.

4.2 Wall Clock Time

The more complex models (SVM and Neural Networks) took significantly more time to fit than their peers, which explains the very long delays I experienced in grid searching. It is also noteworthy that the Neural Network and SVM models increased significantly for the Amazon Reviews data set as their structures became more complex (polynomial kernel in SVM and three hidden layers in Neural Networks). The same level of increase was not present for Pima Diabetes (relatively) as they did not have the same level of structural complexity increase.

The Boosting model also experienced a large jump in the Amazon Reviews data. The optimized model used decision trees with a max depth of 6 instead of the defaulted max depth of 2.

Computation Time to Fit Models Before and After Tuning (In Seconds):

	Pima Diabetes		Amazon Reviews	
	Untuned	Tuned	Untuned	Tuned
Decision Tree	0.1	0.2	4	4
Neural Network	18	21	1,117	3,896
Boosting	6	14	44	289
SVM	0.2	0.2	393	26,039
KNN	0.1	0.1	18	15

5 LEARNINGS

I will refer to learning curves and complexity curves in this section. These can be found at the end of this document.

5.1 Learning Curves

The Amazon Review learning curves differ from the Pima Diabetes learning curves as the train and test accuracies almost never intersect. This can be attributed to the high amount of variance in the data that is not explained by the features. In an "easier" data set, we will see more convergence.

I can see evidences of overfitting in the Pima Diabetes dataset for the SVM staring about one third of the way through as well as near the end of Neural Networks and Decision Trees. We do not see that same pattern for Boosting. Boosting follows the nice pattern of converging without showing signs of overfitting (for the most part).

In the test score of the Amazon Reviews dataset, I see the pattern of rapid improvement in the beginning and a slowing with more data for each of the models, leading me to believe we are not over fitting the data. This pattern is still visible even though there remains a large gap between the train and test accuracies.

5.2 Complexity Curves

I will highlight some important or interesting parameters for each model.

5.2.1 *Decision Trees*

In grid searches I experimented with information gain (entropy) and impurity (gini index) and found the former to yield better results for the Amazon Review data (30.3% with gini to 31.3% with entropy for F1 on Amazon Review data).

The other large factor for decision trees was the method and severity of pruning. I pruned by placing restriction on the tree build including max depth, minimum

rows per leaf, and minimum to split a node. For the data set, max depth was the most effective.

5.2.2 *Neural Network*

The largest factor in the Neural Network models was the structure. A small change to structure (reducing nodes in the hidden layer) created the most benefit in Pima Diabetes, and a large change (adding two hidden layers) created the most benefit for Amazon Reviews.

Activation equation also had influence. I expected ‘relu’ as this seems to be the most common, but the grid search for the Pima Diabetes data found that combinations with the ‘tanh’ equation could yield slightly higher results. The Amazon Reviews Data did favor the ‘relu’ activation, and I suspect that this was more likely due to the multiple hidden layers. (33.3% ‘tanh’ to 34.1% ‘relu’ for Amazon Reviews.)

The alpha also sloped the score but by a very small amount for this data set as shown in figures 13-14.

5.2.3 *Boosting*

Learning rate was important for Boosting, but the base model used seemed to have a much larger effect for the Amazon review. Optimization landed it on a depth of 8, likely due to the complexity and row number of the data as shown in figure 16.

5.2.4 *SVM*

The regularization coefficient impacted the score for both data sets as seen in figures 19-20. A larger effect for the Amazon Reviews dataset, however, came from swapping out the kernels used. The optimal kernel was a polynomial with 5 degrees. (14.1% with rfb to 17.3% with polynomial)

5.2.5 *K Nearest Neighbors*

In KNN, the most influential parameter was the number of neighbors, although the weights did play a role. Distance was favorable for both data sets (F1 of 29.5% with uniform to 33.4% with distance for Amazon Reviews and accuracy of 74.5% with uniform to 75.0% with distance for Pima Diabetes).

6 REFERENCES

1. Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web conference (WWW-2005), May 10-14.
2. Jianmo Ni, Jiacheng Li, Julian McAuley
   Empirical Methods in Natural Language Processing (EMNLP), 2019. (For the

Amazon Review data set). Accessed through https://nijianmo.github.io/amazon/index.html in Sep 2020. Word lists available at https://gist.github.com/mkulakowski2/4289437, https://gist.github.com/mkulakowski2/4289441, and http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html.

3. Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA
4. Scikit-learn library. https://scikit-learn.org/stable/.
5. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. *In Proceedings of the Symposium on Computer Applications and Medical Care* (pp. 261--265). IEEE Computer Society Press. (For Pima Diabetes data set). Accessed through https://www.kaggle.com/uciml/pima-indians-diabetes-database in Sep 2020.

Learning Curves

Figure 1

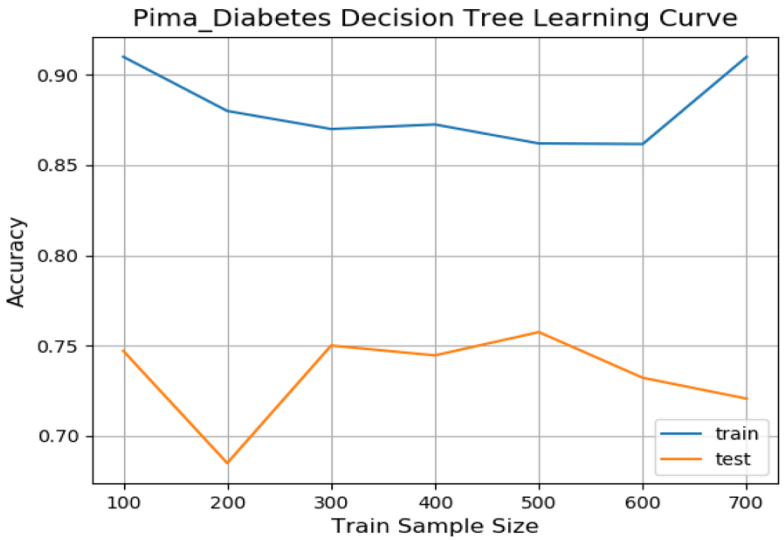


Figure 2

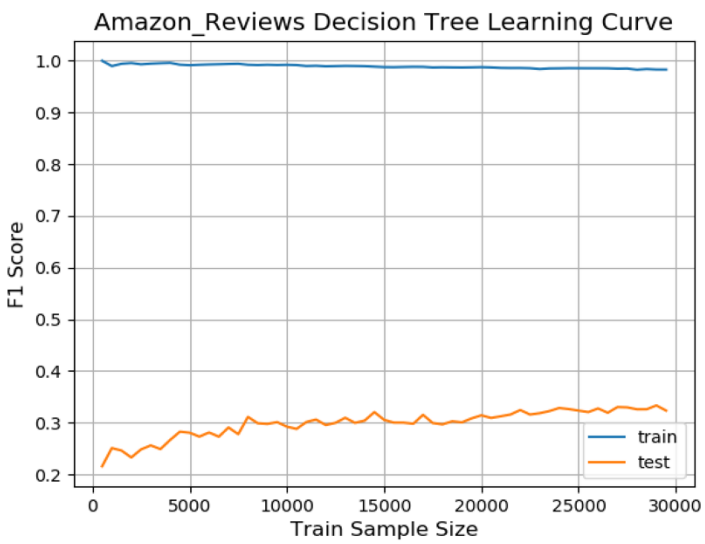


Figure 3

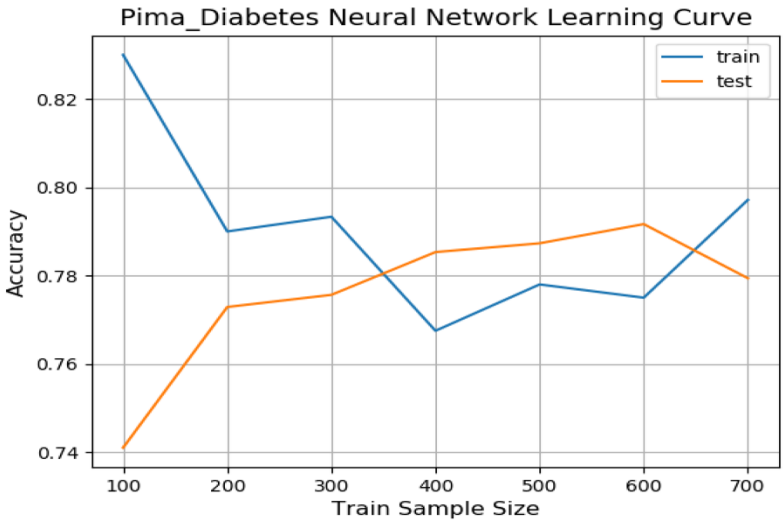


Figure 4

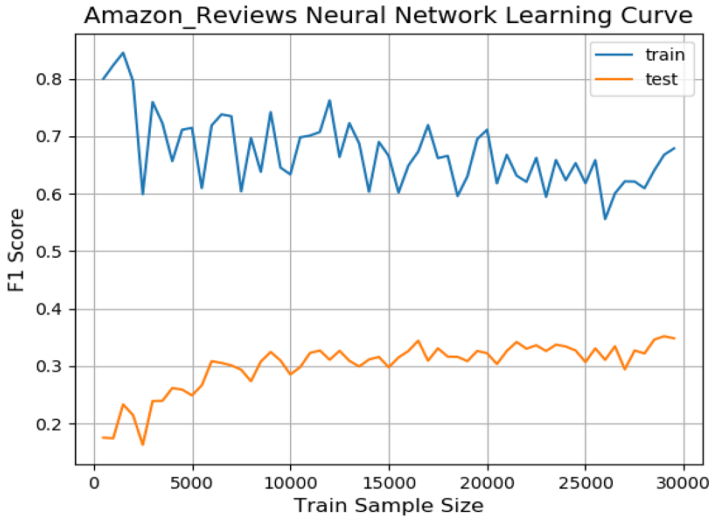


Figure 5

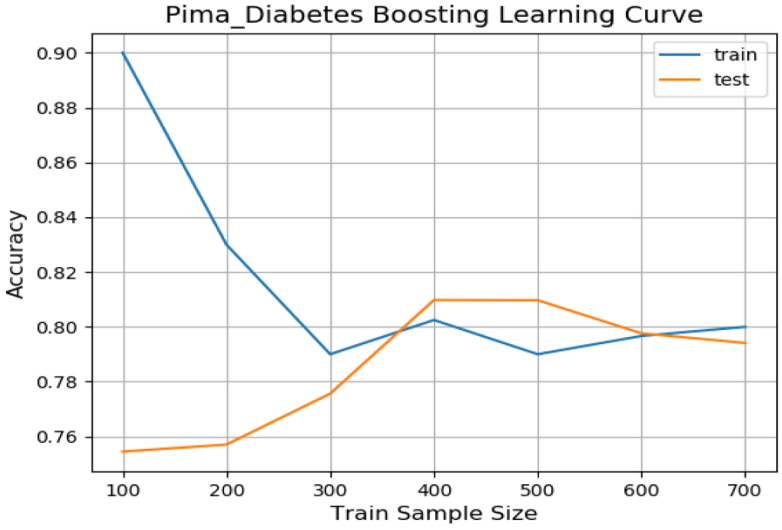
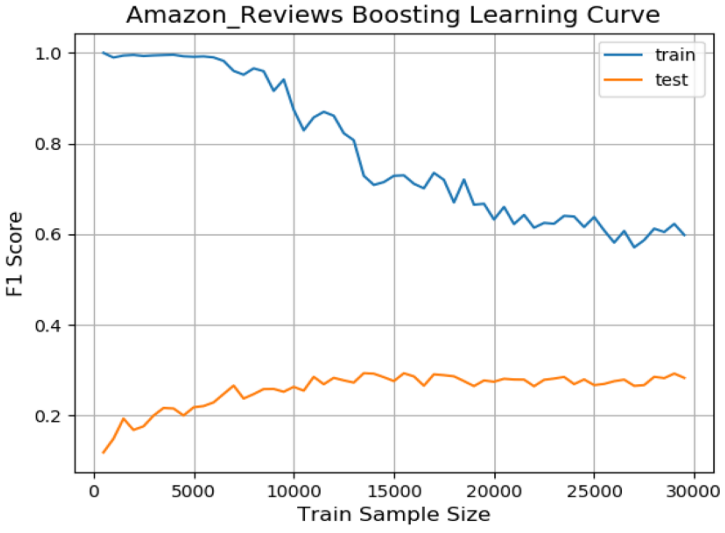


Figure 6



Learning Curves (Continued)

Figure 7

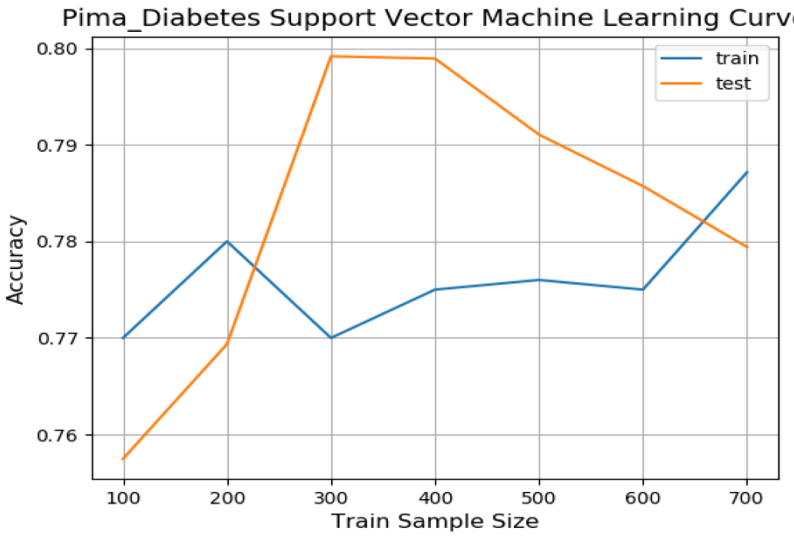


Figure 8

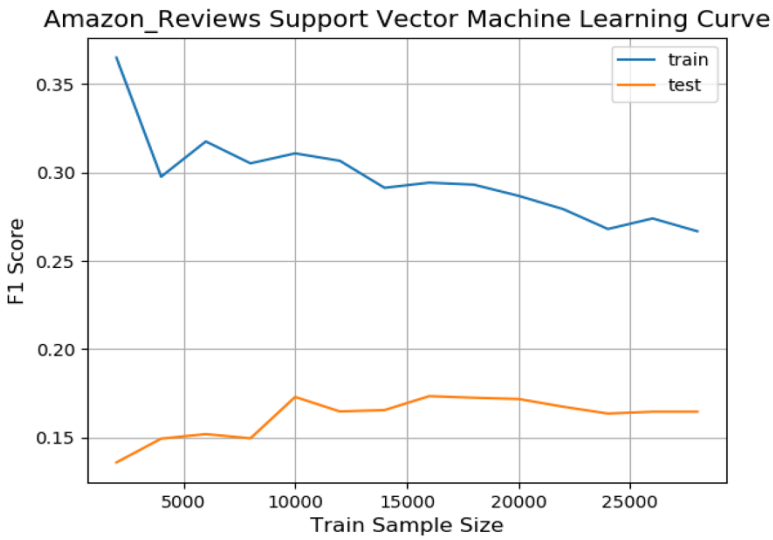


Figure 9

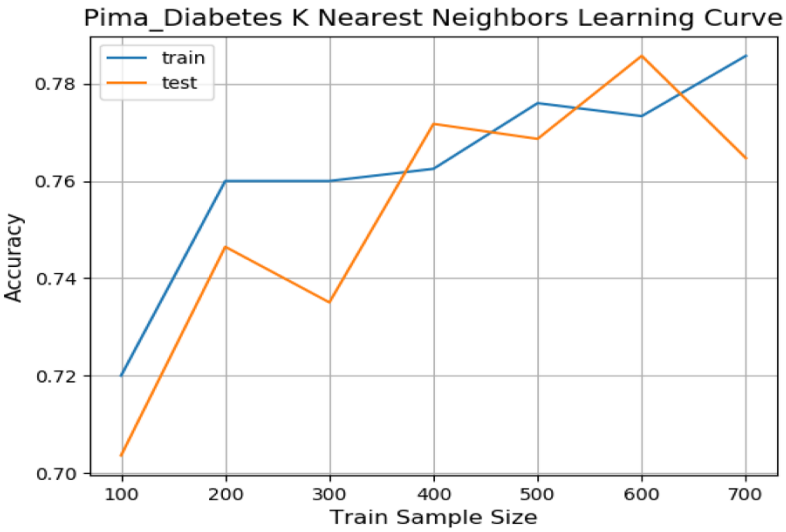
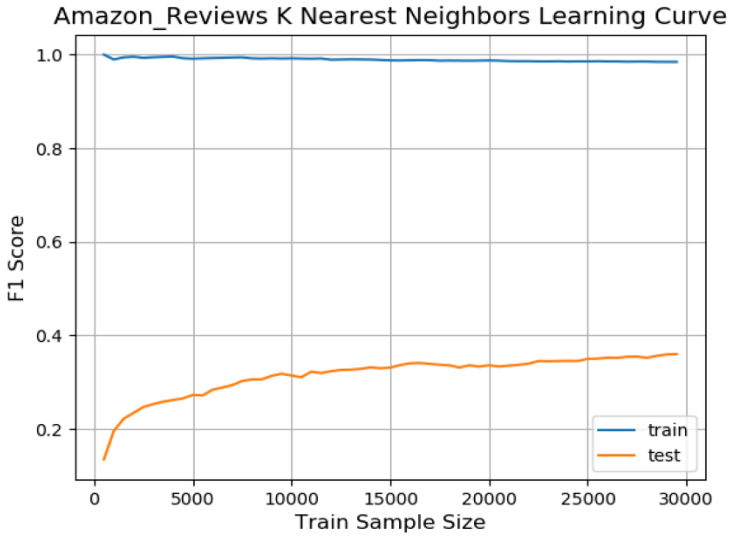


Figure 10



Complexity Curves

Figure 11

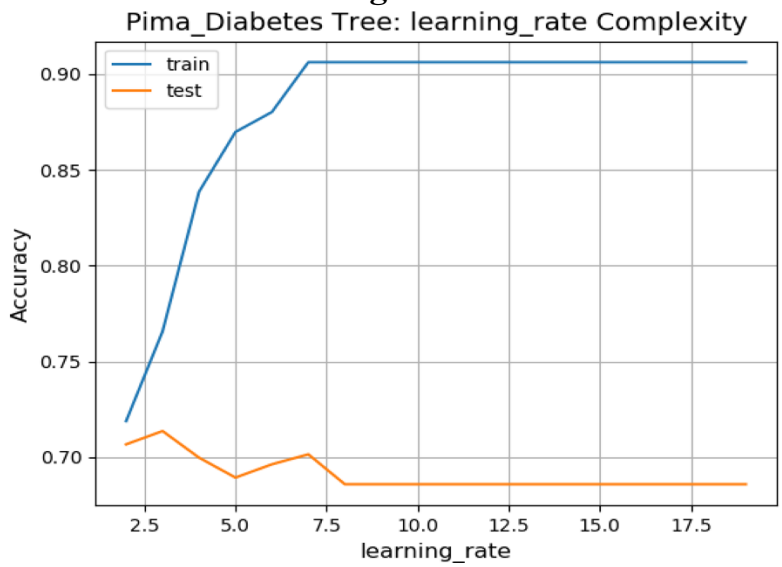


Figure 12

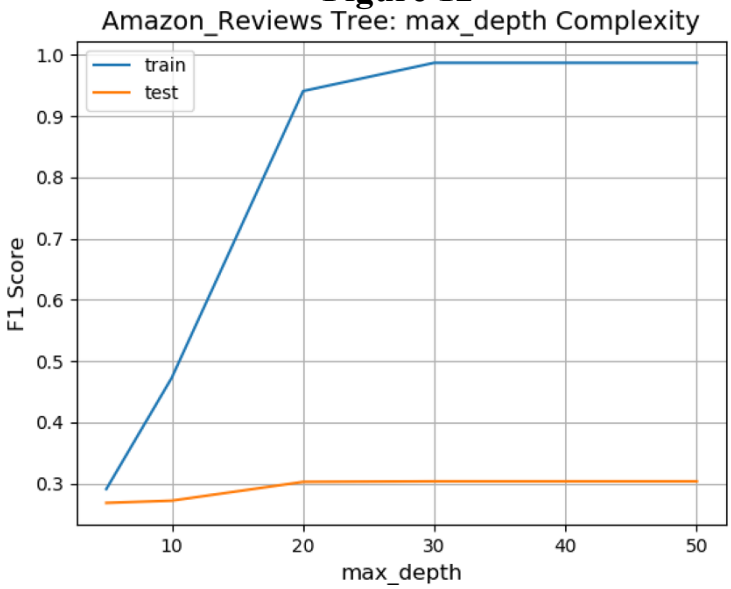


Figure 13

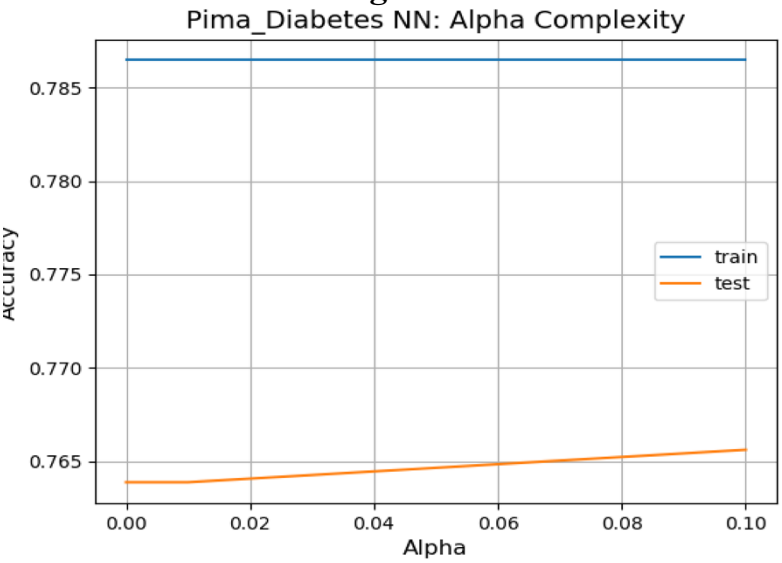


Figure 14

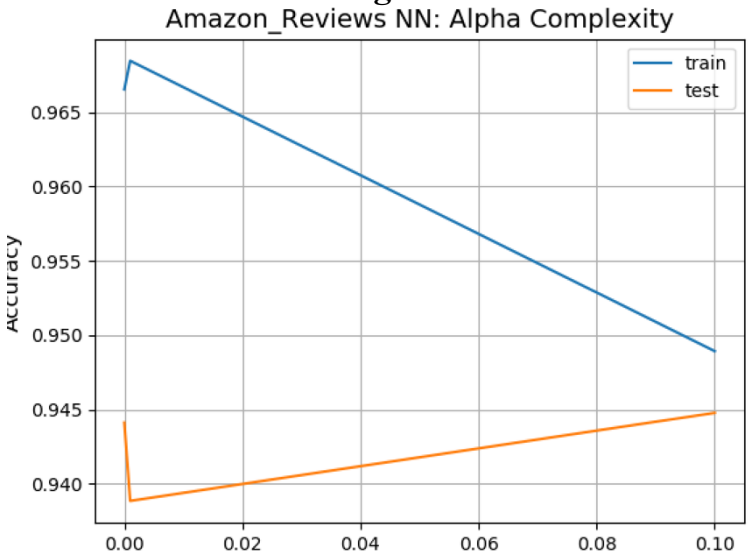


Figure 15

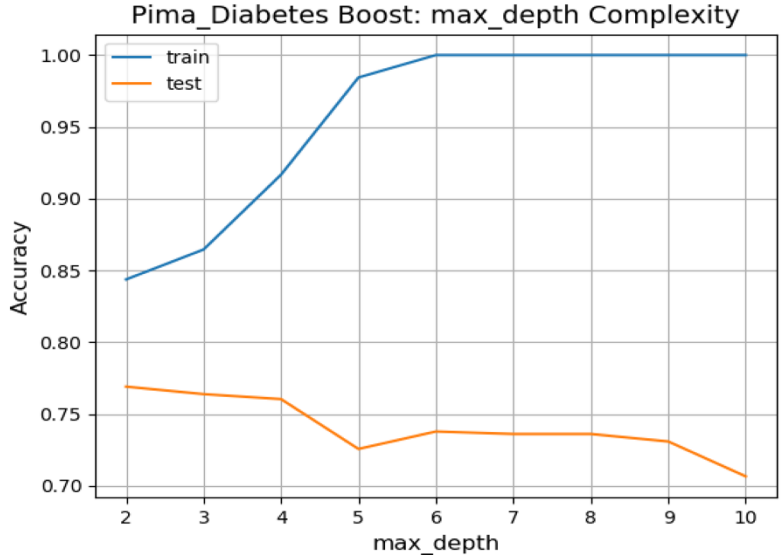
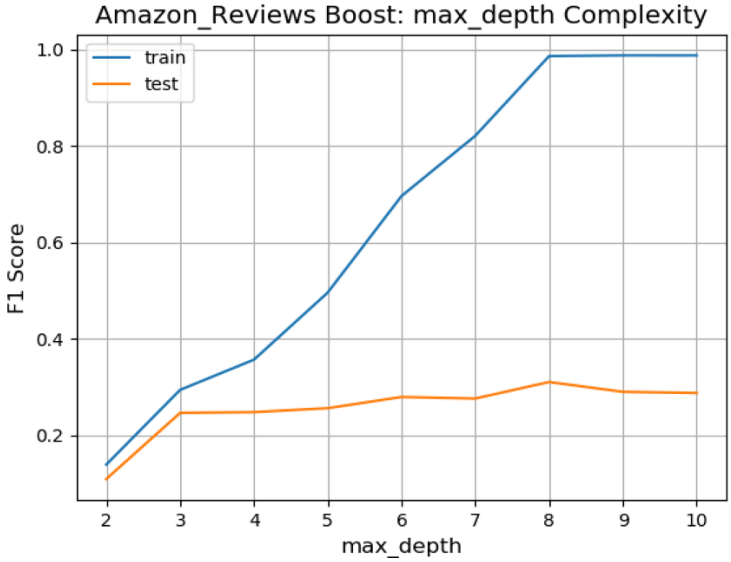


Figure 16



Complexity Curves (Continued)

Figure 17

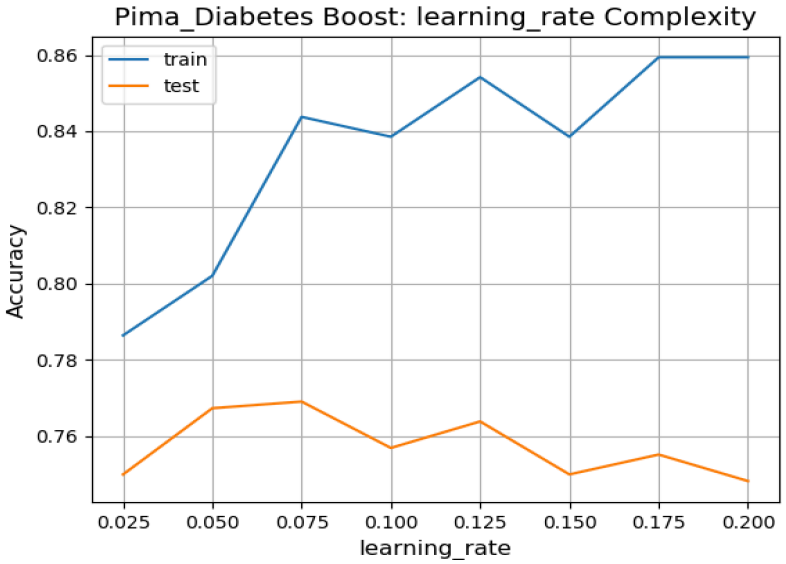


Figure 18

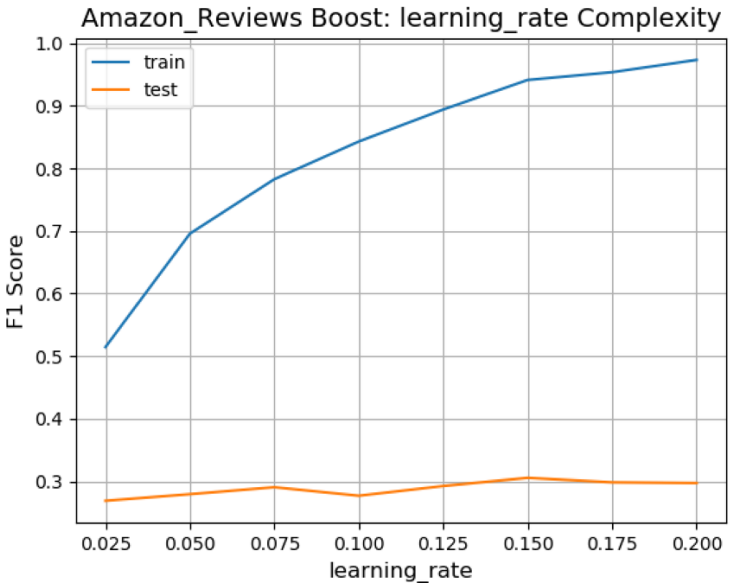


Figure 19

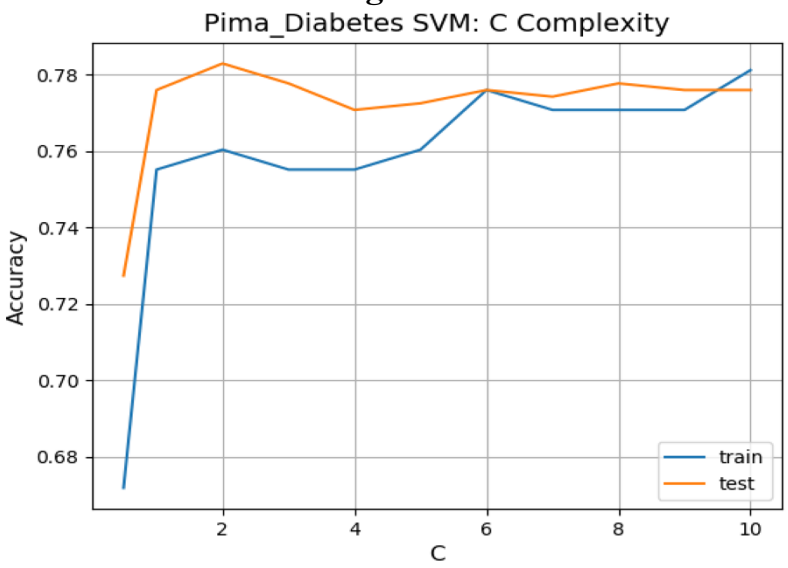


Figure 20

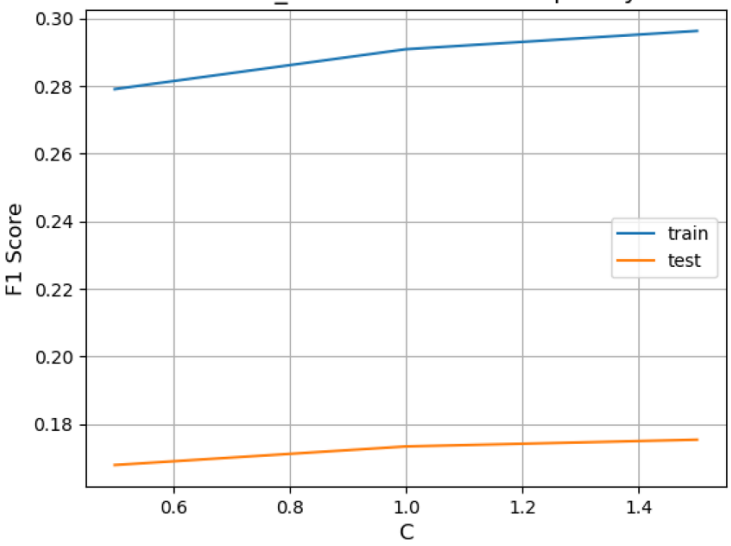


Figure 21

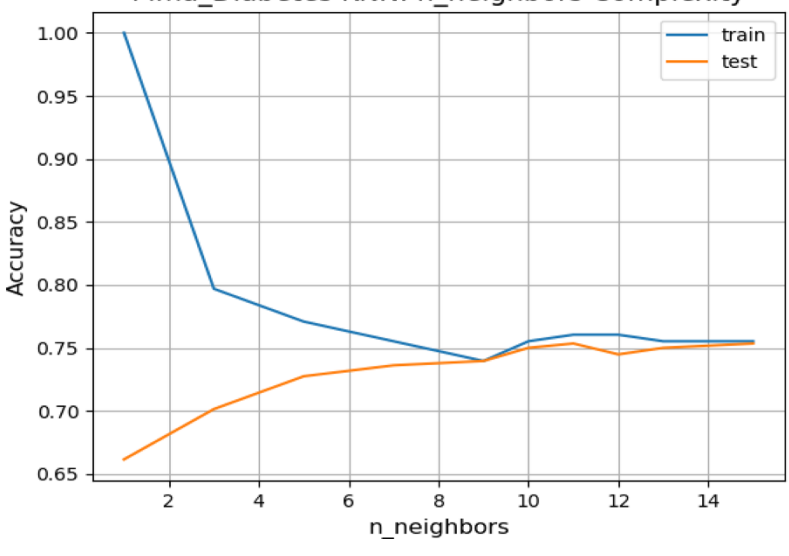


Figure 22