Supervised Learning - Analysis

Younes EL BOUZEKRAOUI
ybouzekraoui3@gatech.edu

1 CLASSIFICATION PROBLEMS DESCRIPTION

1.1 Telecom Churn Dataset

The first classification problem is about Churn Prediction for a Telecommunication company by analyzing all relevant customer data and how they use the service and predicting which customers are most likely to cancel a subscription.

The Churn Dataset includes cleansed customer activity data (features) and a churn label that indicates if a customer has canceled their subscription. (Binary Classification)

From a business standpoint, obtaining this information is critical because attracting new customers is generally more difficult than retaining existing customers. As a result, the information acquired through Churn Prediction allows them to focus more on the customers who are most likely to leave. Many factors influence the reasons for a customer to Churn. It may be the fact that there's a new competitor in the market or something else, and our job is to find such patterns in the data given.

The dataset have the following features (Shown in the Figure) and a target "Churn" wich indicates if the costumer canceled (True) or not (False) his subscription.

State	Account length	Area code	International plan	Voice mail plan	Number vmail messages	Total day minutes	Total day calls	Total day charge	Total eve minutes	Total eve calls	Total eve charge	Total night minutes	Total night calls	Total night charge	Total intl minutes	Total intl calls	Total intl charge	Customer service calls	Churn
KS	128	415	No	Yes	25	265.1	110	45.07	197.4	99	16.78	244.7	91	11.01	10	3	2.7	1	False
OH	107	415	No	Yes	26	161.6	123	27.47	195.5	103	16.62	254.4	103	11.45	13.7	3	3.7	1	False
NJ	137	415	No	No	0	243.4	114	41.38	121.2	110	10.3	162.6	104	7.32	12.2	5	3.29	0	False
OH	84	408	Yes	No	0	299.4	71	50.9	61.9	88	5.26	196.9	89	8.86	6.6	7	1.78	2	False
OK	75	415	Yes	No	0	166.7	113	28.34	148.3	122	12.61	186.9	121	8.41	10.1	3	2.73	3	False
AL	118	510	Yes	No	0	223.4	98	37.98	220.6	101	18.75	203.9	118	9.18	6.3	6	1.7	0	False
MA	121	510	No	Yes	24	218.2	88	37.09	348.5	108	29.62	212.6	118	9.57	7.5	7	2.03	3	False
MO	147	415	Yes	No	0	157	79	26.69	103.1	94	8.76	211.8	96	9.53	7.1	6	1.92	0	False
WV	141	415	Yes	Yes	37	258.6	84	43.96	222	111	18.87	326.4	97	14.69	11.2	5	3.02	0	False
RI	74	415	No	No	0	187.7	127	31.91	163.4	148	13.89	196	94	8.82	9.1	5	2.46	0	False
IA	168	408	No	No	0	128.8	96	21.9	104.9	71	8.92	141.1	128	6.35	11.2	2	3.02	1	False
MT	95	510	No	No	0	156.6	88	26.62	247.6	75	21.05	192.3	115	8.65	12.3	5	3.32	3	False
IA	62	415	No	No	0	120.7	70	20.52	307.2	76	26.11	203	99	9.14	13.1	6	3.54	4	False
ID	85	408	No	Yes	27	196.4	139	33.39	280.9	90	23.88	89.3	75	4.02	13.8	4	3.73	1	False
VT	93	510	No	No	0	190.7	114	32.42	218.2	111	18.55	129.6	121	5.83	8.1	3	2.19	3	False
VA	76	510	No	Yes	33	189.7	66	32.25	212.8	65	18.09	165.7	108	7.46	10	5	2.7	1	False
TX	73	415	No	No	0	224.4	90	38.15	159.5	88	13.56	192.8	74	8.68	13	2	3.51	1	False
FL	147	415	No	No	0	155.1	117	26.37	239.7	93	20.37	208.8	133	9.4	10.6	4	2.86	0	False
CO	77	408	No	No	0	62.4	89	10.61	169.9	121	14.44	209.6	64	9.43	5.7	6	1.54	5	True

Figure 1—Telecom Churn Dataset

The training part of the Data Set Contains 2666 rows and the testing 667 rows with a 80/20 ratio between training and testing data. We can ask if we have enough training Data and if the Data Set size is sufficient, in order to answer this question we need to know the complexity of our classification problem and our learning algorithm, but we can also reason by analogy and look at previous classification problems similar to ours

to have an idea of the data size we need.

1.2 Stars Classification Dataset

The second Classification Problem is Stars Classification

The classification of stars based on their spectral features is known as stellar classification in astronomy. One of the most fundamental classification schemes in astronomy is that of galaxies, quasars, and stars. As more powerful telescopes were created, the early cataloging of stars and their distribution in the sky led to the knowledge that they make up our own galaxy, and after the finding that Andromeda was a separate galaxy from our own, several galaxies began to be examined. The goal of this collection is to categorize stars, galaxies, and quasars based on their spectral properties.

2 ALGORITHMS RESULTS AND ANALYSIS

The algorithms used for this project are taken from the Scikit-learn Python library. The metric used for evaluating the models is the Accuracy metric, which represents the fraction of predictions our model got right.

Accuracy = Number of correct predictions / Total number of Prediction

In order to evaluate the models efficiently the Cross Validation method on the Training data with 5 folds was used and the mean accuracy value was reported as the 'Cross validation Accuracy' , in addition to the accuracy of the model on the Testing data as 'Testing data Accuracy' The Fitting time of the model is also reported in order to evaluate the performance of the model regarding the time.

2.1 Decision Tree

The first approach to predict the target is using a Decission Tree algorithm with it's default parameters (No max depth and no prunning).

The Accuracy results, the Fitting time and the learning curves for the 2 Classification problems are shown in the figure bellow

```
Cross validation Accuracy: 0.908
Testing data Accuracy: 0.918
Fitting time (seconds) = 0.04070754051208496
```

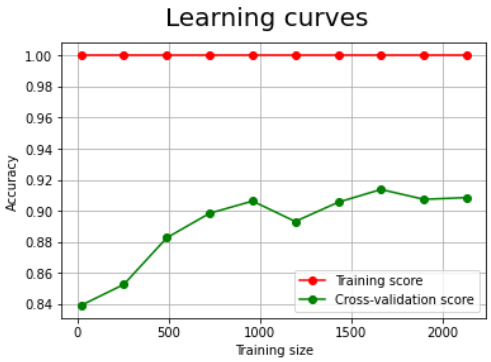

Figure 2—Problem1 : Decision Tree learning curves

```
Cross validation Accuracy: 0.952
Testing data Accuracy: 0.952
Fitting time (seconds) = 0.04885149002075195
```

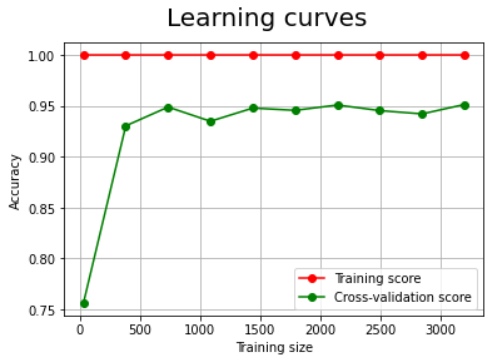

Figure 3—Problem2 : Decision Tree learning curves

One can see on the figure that the train data are at an accuracy of 1 all the time and this means that the bias is low and that our model is too attached to the training data, in addition to that, the significant gap between the two curves means that we have a high variance, and that the model could benefit from more training data. This also means that it is possible to simplify the model, and in our case we can do that by pruning the decision tree

2.2 Decision Tree Prunned

In Order to improve the accuracy and avoid over-fitting , a pruning method of the previous tree was used, the Minimal Cost-Complexity Pruning.

This Type of pruning is used to avoid over-fitting , The more leaf nodes that the tree contains the higher the complexity of the tree, and in our algorithm a parameter "ccp alpha", the Cost-Complexity, is tuned in order of prune the Decision Tree.

The Grid search CV function of Sklearn was used to tune the parameter "ccp alpha" and find the best value that maximizes the cross validation Accuracy score.

The Accuracy results, the Fitting time , the learning curves and a plot of the Cross-validation score with respect to ccp alpha for the 2 Classification problems are shown in the figure bellow

```
Best parameter = {'ccp_alpha': 0.0008563107190753838}
Cross validation Accuracy: 0.933
Testing data Accuracy: 0.952
Fitting time (seconds) = 0.04266057014465332
```

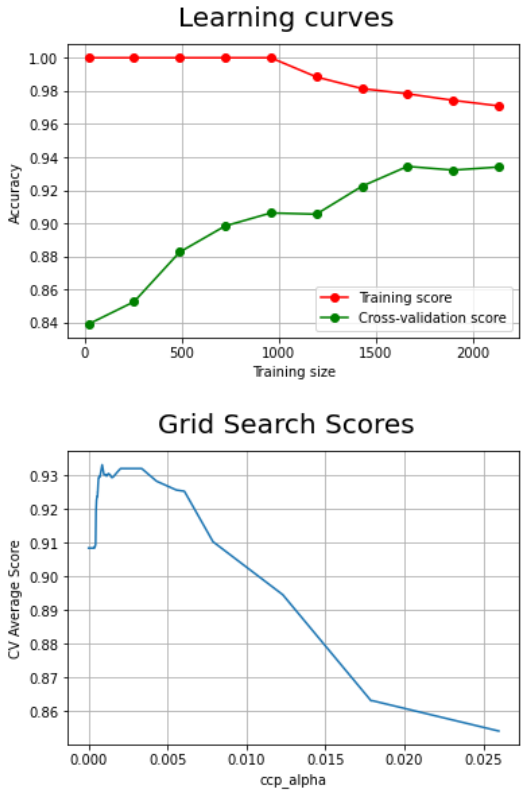

Figure 4—Problem1 : Decision Tree learning curves

```
Best parameter = {'ccp_alpha': 0.0009568055555555549}
Cross validation Accuracy: 0.962
Testing data Accuracy: 0.964
Fitting time (seconds) = 0.04034581184387207
```

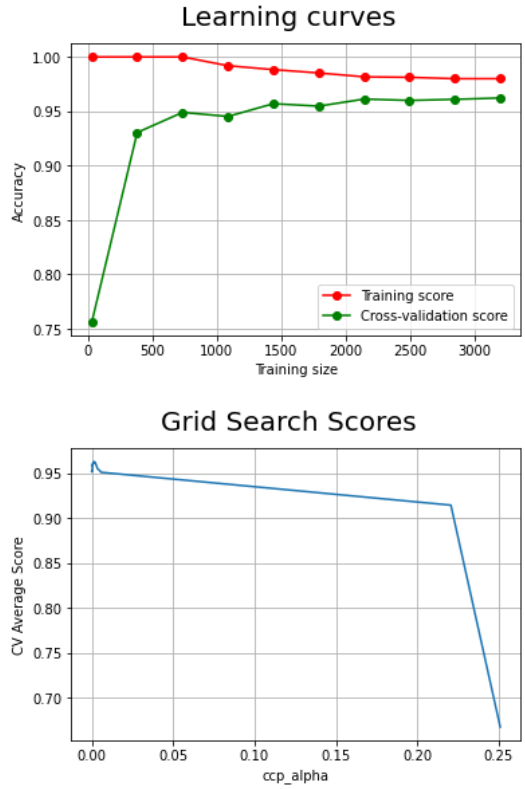

Figure 5—Problem2 : Decision Tree learning curves

One can see that for both classification problems we got a better accuracy score (0.95 and 0.96) , we can also see that the pruned tree models has less gap between the two curves (Train and Cross Val test) thus less variance , and both curves flattens at the end. In terms of time performance the pruned Tree model takes less time to fit the data, which makes sense because it is less complex (having less sub trees) than the previous model.

In general the Pruned Decision Tree model is performing better than the Complex Decision Tree but it needs improvement in order to lower the gap between the red and green curves so we can get a perfect fitting model, in order to do that we can try another type of pruning using the Gini Index or tuning simultaneously the parameters of the Decision tree classifier for example : The minimum number of samples required to be at a leaf or The maximum depth of the tree or The minimum number of samples required to split an internal node.

2.3 Neural network

We used a neural network of 3 dense layers with 64 units each using the relu activation function and the sigmoid activation function for the last layer.

Regarding the time performance the model is taking more time than the other models.

The Accuracy results are not good enough and can be improved by tuning some parameters learning rate, batch size and epochs in order to improve the performance of the model.

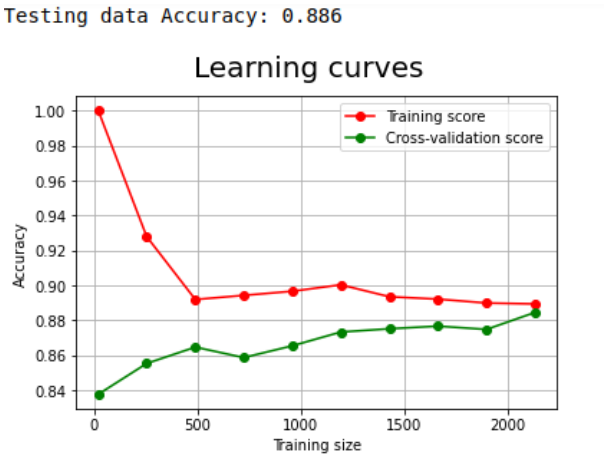

Figure 6—Problem1 : Neural network

2.4 Boosting

In this algorithm the AdaBoost method was used in order to boost our Decision tree. the figures bellow shows the performance of the boosted model and it's learning curves for the Problem 1 (Same shape for Problem 2). The first approach was using a learning rate of 1 which resulted in high training score (equal to 1 all the time) and a large gap between the Train score and the test score curves, which means that we have a low biais but high variance. This problem can be corrected by tuning the learning rate parameter, the second figure bellow shows the results for a learning rate of 0.00001

```
Cross validation Accuracy: 0.926
Testing data Accuracy: 0.933
Fitting time (seconds) = 2.846229076385498
```

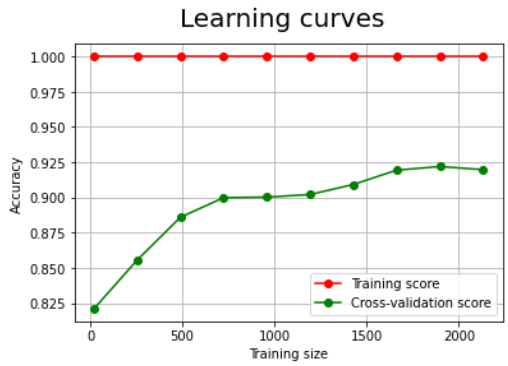

Figure 7—Problem1 : Ada boost learning curves with a learning rate of 1

```
Best parameter = {'learning_rate': 0.0001}
Cross validation Accuracy: 0.937
Testing data Accuracy: 0.949
Fitting time (seconds) = 2.161827278137207
```

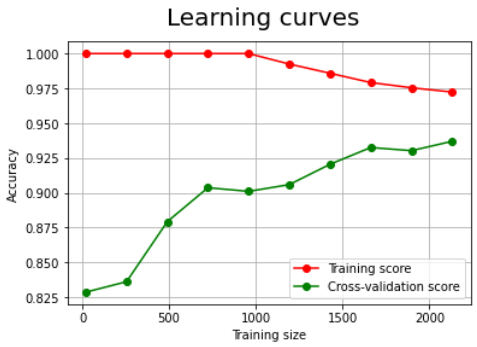

Figure 8—Problem 1 : Ada boost learning curves with a learning rate of 0.00001

One can observe that the performance of the model was improved (Better Accuracy for test data and cross validation) and the gap between the curves is lower. Regarding the fitting time , the model with the learning rate value of 0.00001 has lower fitting time is due to the fact that with small learning rate the algorithm is moving slowly in the direction of the optimal result but when he finds the optimal result it stops , in contract the model with higher learning rate is more likely to miss the optimal result and stop until he reaches the maximum iterations (50 by default in Skitlearn AdaBoost)

2.5 SVM

A Support vector machines model with various kernels and parameters was used to do classification. The first approach was using the Radial basis function kernel, as we can see in the following figures the performance of the model is not good enough (Low Accuracy , and a flat test curve all the time).

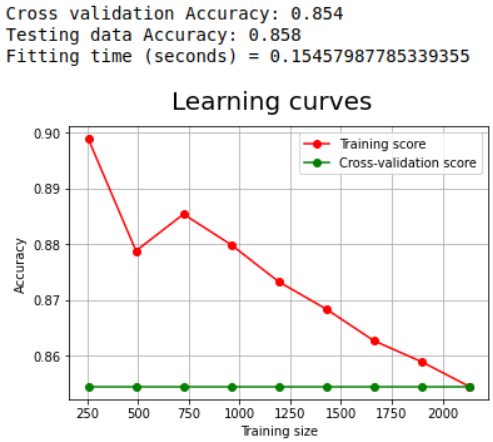

Figure 9—Problem 1 : SVM learning curves Radial basis function kernel

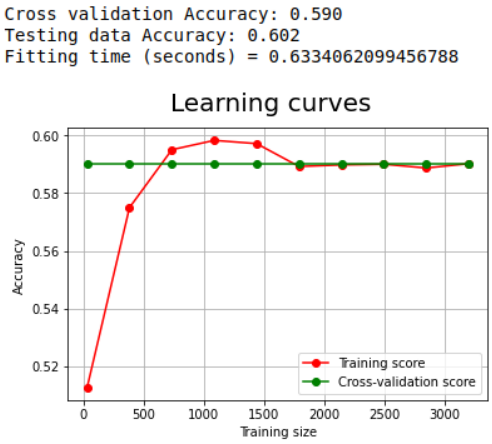

Figure 10—Problem 2 : SVM learning curves Radial basis function kernel

The second Approach was using the polynomial kernel and tuning the degree parameter , the results are better than the previous kernel for the problem 1 but the same for the Problem 2 , this might be due to the data distribution on each problem and that the Problem 2 needs a more complex kernel (Exponential) in order to be able to

separate the classes and classify them correctly A possible good kernel that may do that is the Gaussian Kernel.

```
Best parameter = {'degree': 18}
Cross validation Accuracy: 0.892
Testing data Accuracy: 0.901
Fitting time (seconds) = 0.4561335563659668
```

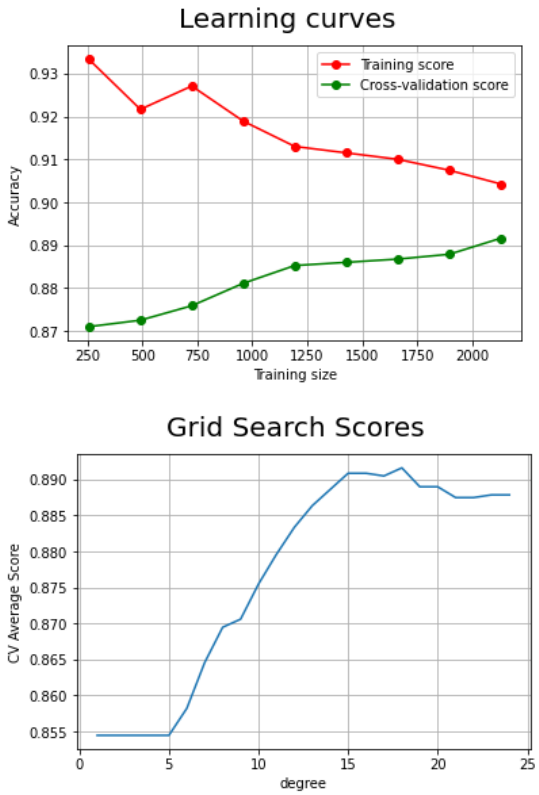

Figure 11—Problem 1 : SVM learning curves Polynomial kernel

```
Cross validation Accuracy: 0.590
Testing data Accuracy: 0.602
Fitting time (seconds) = 0.5224418640136719
```

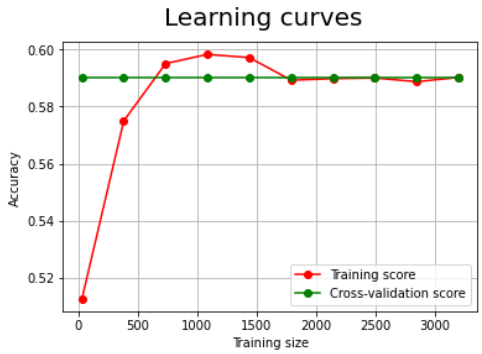

Figure 12—Problem 2 : SVM learning curves Polynomial kernel

2.6 K Nearest neighbors

Another model was used to predict the target , which is K nearest neighbors , the first approach is to use the KNeighborsClassifier from sklearn with it's default parameters K =5. The Accuracy results obtained by cross validation and the learning curves are shown in the figure 4

```
Cross validation Accuracy: 0.854
Testing data Accuracy: 0.865
Fitting time (seconds) = 0.0051883220672607425
```

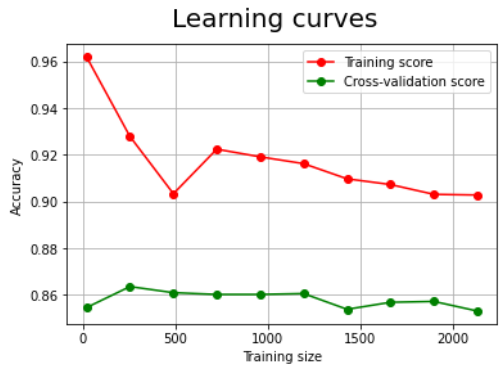

Figure 13—Problem 1 : K =3 nearest neighbors learning curves

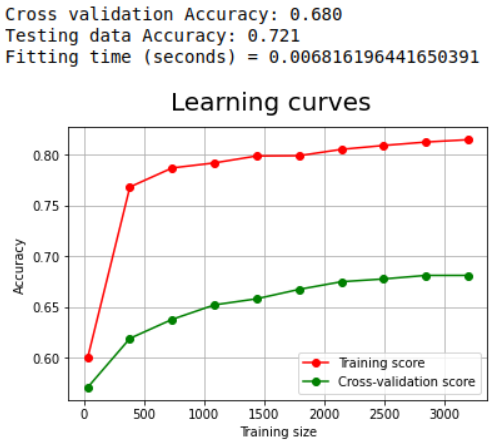

Figure 14—Problem 2 : K =3 nearest neighbors learning curves

2.7 K Nearest neighbors Optimal

In order to perform hyperparameter tuning , the GridSearchCV function of Sklearn was used to iterate over 30 values of K [1 to 30] and find the value of K that maximizes the accuracy using cross validation. The best value of K and the learning curves of the corresponding model are shown on the figure 5

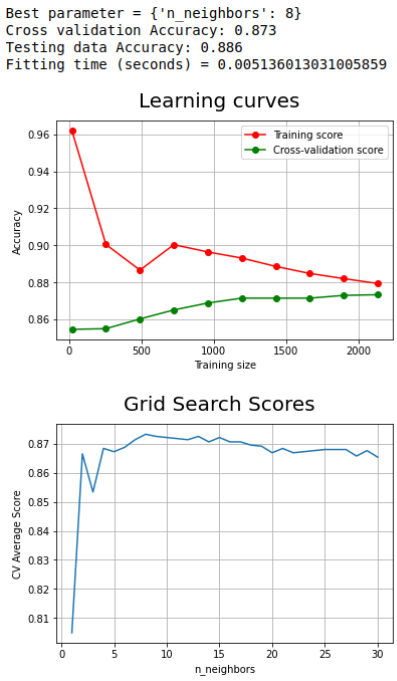

Figure 15—Problem 1 :K =8 nearest neighbors learning curves

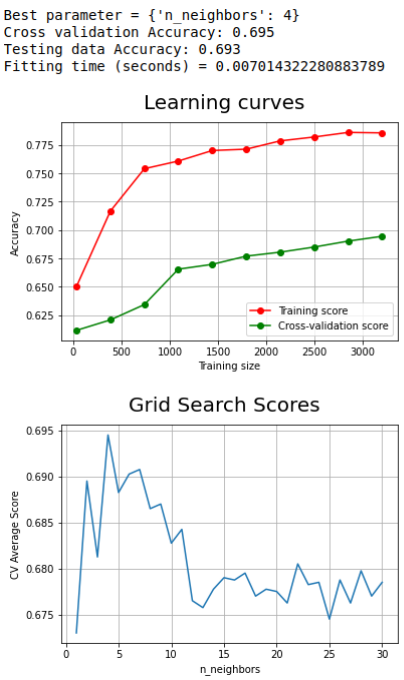

Figure 16—Problem 2 :K =4 nearest neighbors learning curves

3 CONCLUSION

After trying and observing the performance of all the models we can say the the best one that is fitting good our data for both the Classification Problems is the Boosted version of the Decision Tree with an accuracy of 0.95 for the Problem 1 and 0.97 for the Problem 2 .