CS7641 Fall 2022 HW1 Supervised Learning

Fung Yi Yuen
fyuen3@gatech.edu

***Abstract*— Two datasets, Divorce Predictors and Contraceptive Method Choice were chosen from the UCI Machine Learning Repository. Five machine learning algorithms (Decision trees, Neural networks, AdaBoost, Support Vector Machines and k-nearest neighbors) were trained and tested based on the chosen datasets to explore the behavior of the learners under different circumstances.**

I. DATASET DESCRIPTION

A. Divorce Predictors dataset

This dataset contains 170 entries, 53 features and 1 class attribute. The features are interview questions related to the daily lives of married and divorced couples. Their corresponding answers are rated from scale 0 to 4, in which 0=Never, 1=Seldom, 2=Averagely, 3=Frequently, 4=Always. The class attribute is either married or divorced (binary classification). The class distribution is almost balanced (51% married, 49% divorced).

B. Contraceptive Method Choice dataset

This dataset contains 1473 entries, 9 features and 1 class attribute. The features are family's information which contains both binary and categorical attributes while the class has 3 types, no-use, long-term or short-term (multiclass classification). The class distribution is imbalanced (42.7% no-use, 22.6% long-term, 34.7% short-term).

Choosing these two datasets would be an interesting experiment to see if the learning algorithms behave well/poorly under different circumstances such as sample size, balanced vs imbalanced dataset, binary vs multiclass classification etc. The Divorce Predictors dataset contains a small number of entries (only 170) which is nearly 9 times less than the Contraceptive dataset (1473 entries). The Divorce Predictors dataset has a balanced class distribution while the Contraceptive dataset is imbalanced. The Divorce Predictors dataset is a binary classification problem while the Contraceptive dataset is a multiclass classification problem. Due to the above-mentioned difference between the two datasets, the performance and hyperparameters tuning would be different as well.

II. DATA PREPROCESSING

No missing values in both datasets thus all entries are preserved. All the feature and class attributes are numeric in both datasets so no conversion of text-based value to number is required. Both datasets use 80/20 split for training (80%) and testing (20%) sets. The training set is further split into training and validation sets using k-fold cross validation. F1-score is used to evaluate the performance of each training model since the contraceptive dataset is imbalanced. Both precision and recall is important during performance evaluation.

A. Divorce Predictors dataset

This dataset only contains 170 entries. Using 5-fold cross validation would leave the model fewer training samples than that of 10-fold cross validation to learn the pattern. Thus 10-fold cross validation is used for this dataset.

One problem with this dataset is that 53 features is quite a lot to compute during the training phase. Too many features means the model trained could be more complex, which would easily lead to overfitting, especially when the dataset is small. To avoid this "Curse of dimensionality", features which are highly correlated will be removed. For example, features like "I enjoy our holidays with my wife." and "I enjoy traveling with my wife." are highly correlated which means either one of them cannot provide new information during data split but increase compute time. Thus the approach here is to pick the least 10 correlated features among all 53 features:

TABLE 1
LEAST 10 CORRELATED FEATURES

Feature	Correlation
Atr42	0.642307
Atr48	0.633564
Atr53	0.611422
Atr47	0.582693
Atr52	0.575463
Atr45	0.510160
Atr43	0.482223
Atr7	0.427989
Atr46	0.400296
Atr6	0.287140

It is an almost balanced dataset (51% married vs 49% divorced). But to ensure the training and testing sets have similar variance, stratified sampling in train/test split and k-fold cross-validation is used.

B. Contraceptive Method Choice dataset

This dataset contains 1473 records which is a lot more than the divorce dataset. 5-fold cross validation is used to optimize compute time. 10-fold is not used here since this dataset is much larger than the divorce dataset and has enough samples for training.

All 9 features in this dataset are preserved in predicting which contraceptive method is used out of the 3 classes.

The class distribution is imbalanced in this dataset (43% vs 23% vs 34%) so stratified sampling is used in train/test split and k-fold cross validation to preserve similar variance

such that the population in training and testing sets best represent the whole dataset.

III. DECISION TREE

Maximum depth and minimal cost-complexity pruning (aka ccp_alpha), are chosen as the hyperparameters to tune the performance of the decision tree.

A. Divorce Predictors dataset

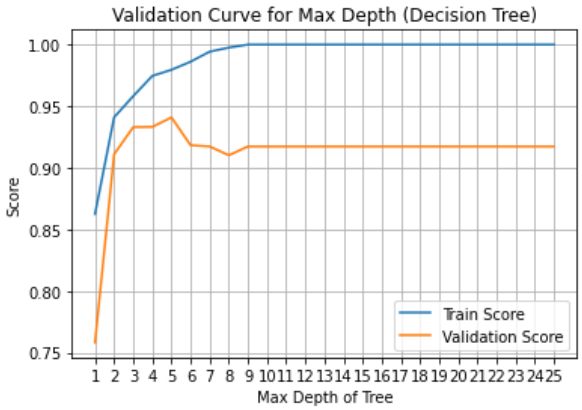

Fig.1 Validation curve for max. depth (divorce dataset)

Observed from fig.1, the optimal value of max. depth should be 5 where the validation score is the highest while the training score is not 1. Both training and validation scores are low when maximum depth is small. It means the model underfits the data and predicts poorly on both training data and unseen data.

As max. depth increases, both training and validation scores increase logarithmically. It means the model predicts better as the decision tree becomes more complex.

However, the validation score drops after max. depth>5 and the training score still keeps rising. It means the model starts to fit training data better but not improve its prediction on unseen data. At max. depth=9, the training score reaches 1 which means the model has high variance and overfits the training set.

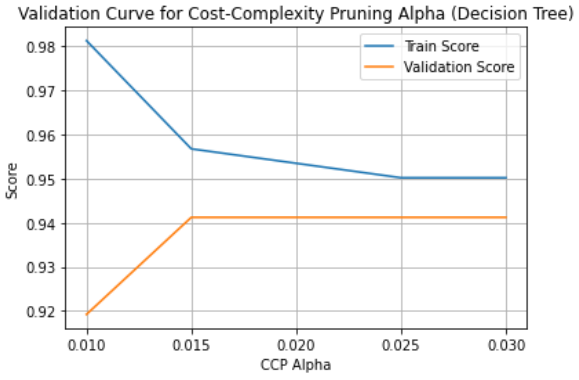

Fig.2 Validation curve for ccp_alpha (divorce dataset)

Observed from fig.2, the optimal value of ccp_alpha should be 0.015 where the validation score is the highest and no more rises with increasing ccp_alpha. When ccp_alpha is small (nearly no pruning), the gap between training and validation score is huge which indicates the model overfits the training data but predicts poorly on unseen data. The decision tree grows so large that every node only contains pure instance(s). Thus it fits perfectly on training data but fails to generalize.

As ccp_alpha increases (i.e. pruning more nodes), the training score drops a little bit but the validation score rises. It means the model predicts better on unseen data. But further pruning after ccp_alpha>0.015 does not improve the model since both training and validation score plateau.

Using the optimals values observed previously (ccp_alpha=0.015, max_depth=5) to plot the learning curve:

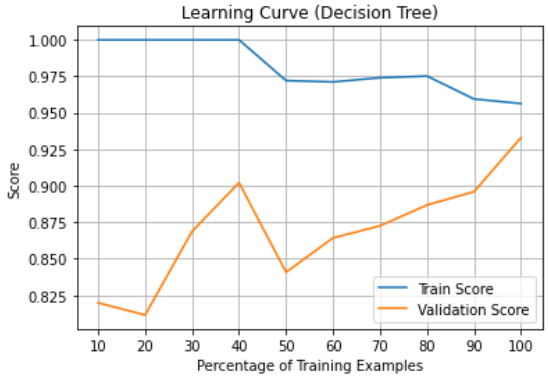

Fig.3 Learning curve for decision tree using optimal values from single hyperparameter tuning (divorce dataset)

Using grid search best params (ccp_alpha=0.01, max_depth=3) to plot the learning curve:

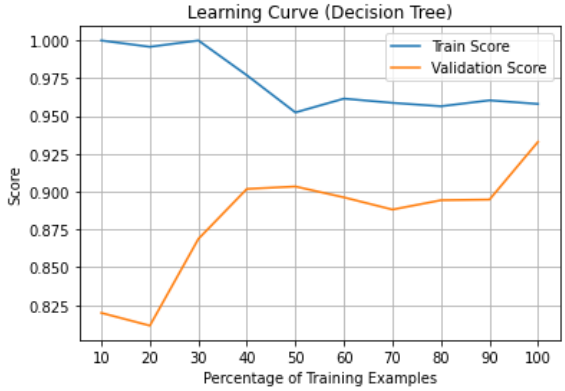

Fig.4 Learning curve for decision tree using grid search best params (divorce dataset)

Turns out f1-score for both the grid search best params and optimals values found from tuning are the same (0.94).

Grid search result:

Best params for decision tree:	{'ccp_alpha': 0.01, 'max_depth': 3}
Time to fit:	5.5501322746276855
Time to predict:	0.006935834884643555
F1_score for best decision tree:	0.9411764705882353

Observed from both fig.3 and fig.4, there is a huge gap between the training and the cross-validation score when sample size is small. This suggests the model has high variance and might need more data to improve its learning performance. As the training samples increase, the gap becomes smaller and smaller (i.e. the two curves converge). It means the variance becomes smaller so that the model is not overfitting to the training data.

Notice that the training score drops slightly as the sample size increases. It is due to generalization when more data is fed to the model. With small samples, the model learns "too well" to fit the data but fails to generalize for unseen data. That is why the training score is 1 while the validation score is much lower. As more data is fed, the training score only drops slightly and then plateau. But the validation score gradually increases which means the model predicts unseen data more accurately.

B. Contraceptive Method Choice dataset

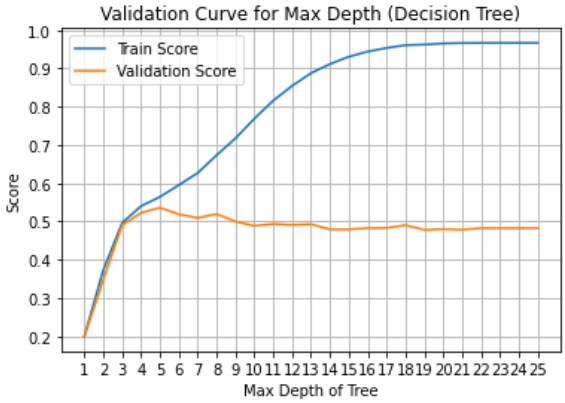

Fig.5 Validation curve for max. depth (contraceptive dataset)

From fig.5, the optimal maximum depth observed should be 5, where the validation score is the highest while the training score is not overfitting the dataset.

This validation curve is similar to the one plotted for divorce dataset. Both training and validation scores are low when the maximum depth of the tree is small. The model has high bias.

As maximum depth increases, both the scores increase which means bias is reduced. The validation score hits its highest at maximum depth=5. Any further increase of maximum depth does not help improve the performance because the validation score plateau but the training score keeps rising. This indicates the model fits training data better than unseen data and would finally become overfits when training score reaches 1 (low bias, high variance).

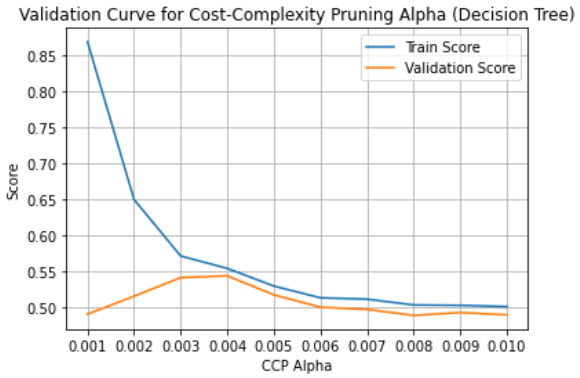

Fig.6 Validation curve for ccp_alpha (contraceptive dataset)

From fig.6, the optimal value for ccp_alpha should be 0.004 where the validation score is the highest. The huge gap between training and validation scores when ccp_alpha is small indicates that the model has low bias but high variance. As ccp_alpha increases to 0.004, the validation score rises to the highest score. But after that both training and validation scores drop, which means the model is underfitting the data and predicts poorly on both training and validation sets.

Grid search finds the best combination of ccp_alpha and maximum depth are 0.003 and 6 respectively, which are very close to the optimal values found from single hyperparameters tuning. F1-score of grid search best param (0.556) is slightly better than that of single hyperparameter tuning (0.554).

Using grid search best params to train the decision tree and plot the learning curve (fig.7):

Best params for decision tree:	{'ccp_alpha': 0.003, 'max_depth': 6}
Time to fit:	6.488149642944336
Time to predict:	0.003972768783569336
F1 score for best decision tree:	0.5564675145420487

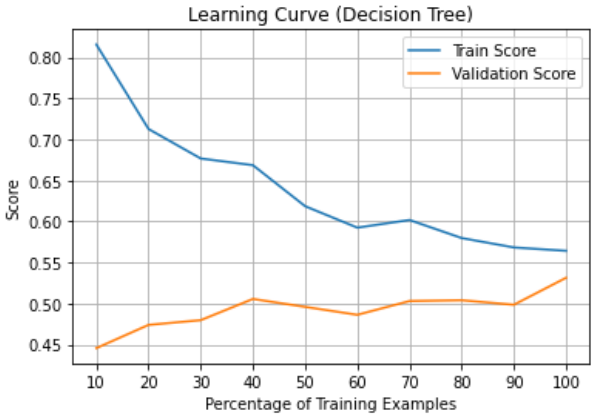

Fig.7 Learning curve for decision tree (contraceptive dataset)

Fig.7 shows there is a huge gap between training and validation scores when 10% of samples are used. This indicates the model has low bias but high variance. As the

sample size increases, the gap becomes smaller and the two scores converge. This indicates the model predicts unseen data better as the sample size increases.

However, the f1-macro score for this dataset is lower than the divorce dataset when looking at the convergence point of the two scores. The best score this model can reach is around 0.55 which is lower than the model trained from divorce dataset (best score: 0.94). The training score of this model keeps decreasing as the sample size increases.

One explanation would be that this dataset is imbalanced (42.7% no-use, 22.6% long-term, 34.7% short-term). The model learns well for the class(s) with abundant samples while it learns poorly for the class(s) with fewer samples. Overall performance thus was affected. Another explanation would be that the features used in this dataset are not relevant in predicting contraceptive methods used. Features such as number of children, husband's occupation or standard-of-living index might not help in predicting contraceptive methods used. Thus taking these features into the algorithm would affect the performance.

IV. NEURAL NETWORKS

Number of nodes in hidden layers and learning rate are chosen as hyperparameters for tuning.

A. Divorce Predictors dataset

Initially **ONE** hidden layer was used to train the neural networks model. But the performance is not good by what the validation curve can tell (fig.8).

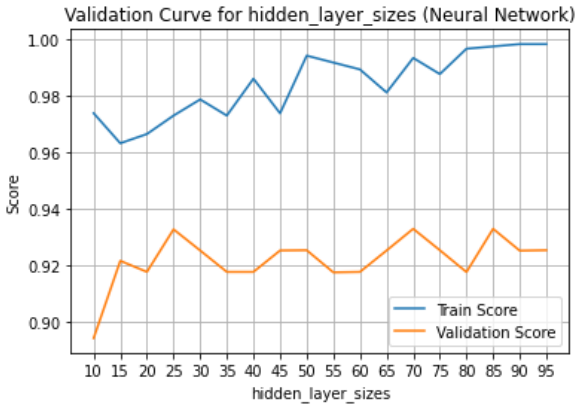

Fig.8 Validation curve for number of nodes in ONE hidden layer (divorce dataset)

There is always a large gap between training and validation scores no matter what number of nodes used. It means the model only fits training data but fails to predict unseen data. Thus **TWO** hidden layers were used and the same validation curve plotted again (fig.9):

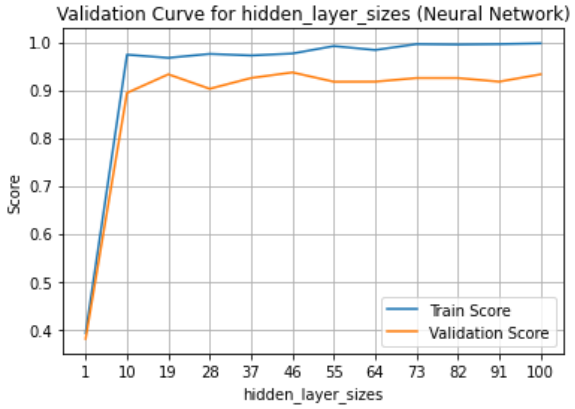

Fig.9 Validation curve for number of nodes in TWO hidden layers (divorce dataset)

The performance is much better since the two scores rise and converge as the number of nodes increases. The model reaches its optimal performance when the number of nodes=19 where both scores are high and the gap is the smallest.

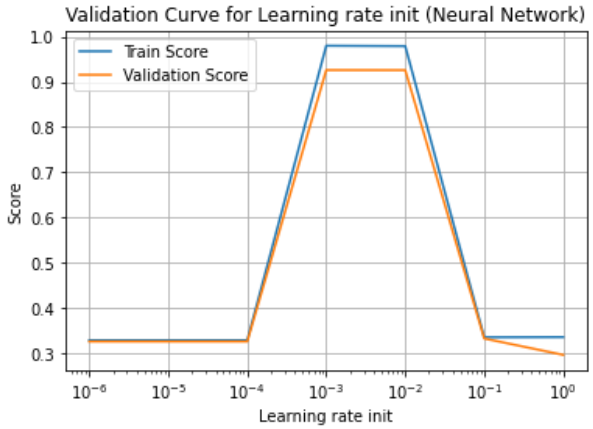

Fig.10 Validation curve for learning rate (divorce dataset)

From fig.10, the optimal learning rate should either be 0.001 or 0.01 since both training and validation scores are at their highest values and the two scores are not diverging. When the learning rate is too small, the model gets stuck so the performance is low and not improving. When the learning rate is close to 1, the model quickly jumps to suboptimal value so its performance drops.

Grid search result shows the best value for learning rate and number of neurons are 0.001 and 19 which are the same as the findings from single hyperparameter tuning.

Using grid search best params to train the neural networks and plot the learning curve (fig.11):

Best params for neural networks:	{'hidden_layer_sizes': 19, 'learning_rate_init': 0.001}
Time to fit:	323.2499055862427
Time to predict:	0.004389762878417969
F1_score for best neural networks:	0.9705627705627706

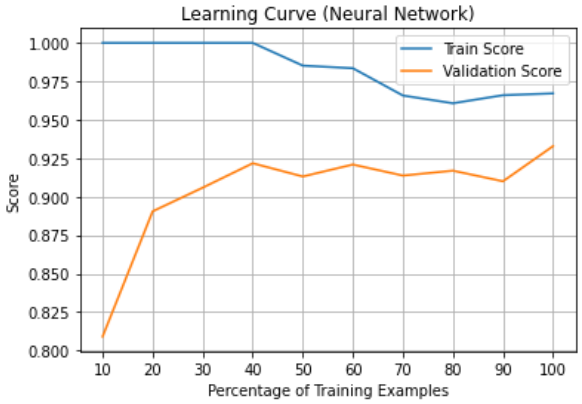

Fig.11 Learning curve for neural networks (divorce dataset)

Fig11 shows there is a huge gap between training and validation scores, which suggests that the model has low bias but high variance. As sample size increases, the two scores converge to a high score near 0.95 which means the model benefits from more training samples. The validation score keeps rising as sample size increases.

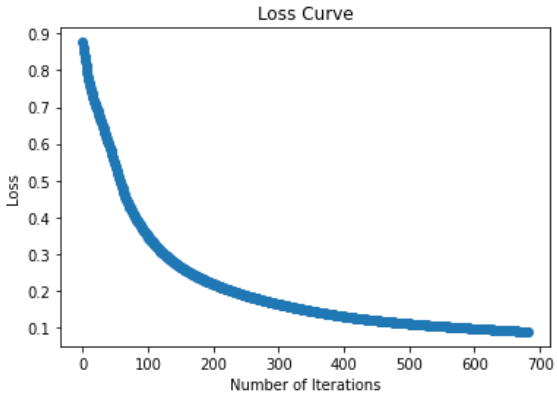

Fig.12 Loss curve for neural networks (divorce dataset)

Fig.12 shows that the training loss function decreases as the number of iterations increases. The model converges to a stable low loss value as the number of iterations reaches 700. But this is a small dataset with only 170 samples. Too many iterations could lead to overfitting thus an early stop might be necessary to avoid the model fitting noise data.

B. Contraceptive Method Choice dataset

Similar to the divorce dataset, the model performs poorly when one hidden layer is used (fig.13). The two scores never converge to a higher point (high variance).

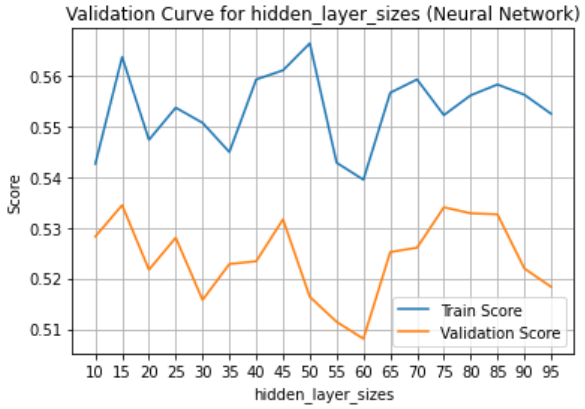

Fig.13 Validation curve for number of nodes in ONE hidden layer (contraceptive dataset)

The model performs better when using two hidden layers (fig.14). Both training and validation scores rise and converge as more nodes are added. The optimal number of nodes should be 13 where both training and validation scores are the highest while the gap between two scores is small.

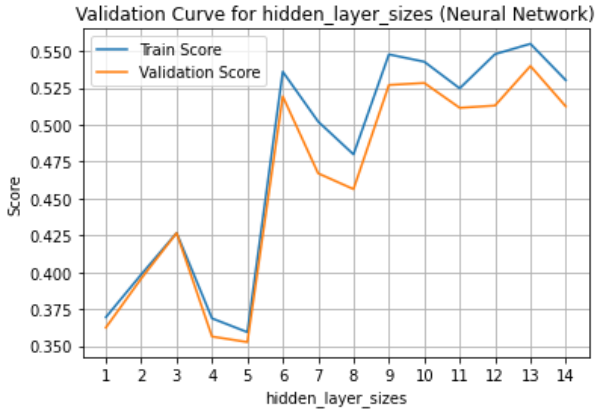

Fig.14 Validation curve for number of nodes in TWO hidden layer (contraceptive dataset)

Fig.15 shows the validation curve for various learning rates using two hidden layers. The optimal learning rate should be 0.001 where both training and validation scores are at their highest points.

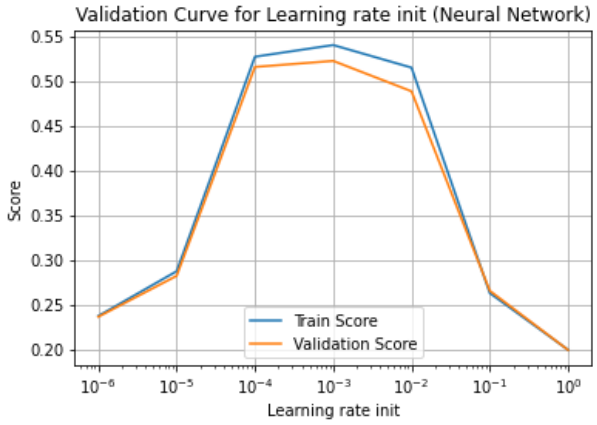

Fig.15 Validation curve for learning rate (contraceptive dataset)

Grid search result shows the best value for learning rate and number of neurons are 0.001 and 13 which are the same as the findings from single hyperparameter tuning.

Using grid search best params to train the neural networks and plot the learning curve (fig.16):

Best params for neural networks:	{'hidden_layer_sizes': 13, 'learning_rate_init': 0.001}
Time to fit:	808.523562669754
Time to predict:	0.0058765411376953125
F1_score for best neural networks:	0.5637511979294003

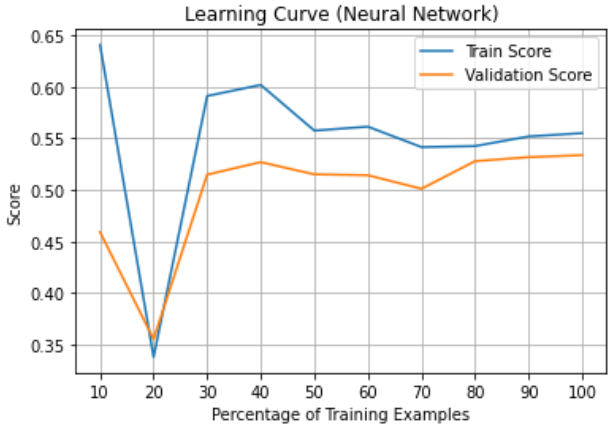

Fig.16 Learning curve for neural networks (contraceptive dataset)

From fig.16, the huge gap at sample size=10% suggests the model has low bias but high variance. Both training and validation scores plummeted at sample size=20%. This could be because this multiclass dataset does not have enough samples for certain minority class(s) when the sample size is small. Thus the prediction for those minority class(s) is poor. As sample size increases, both scores rise again and gradually converge. It means the model predicts both training and unseen data better with more training samples.

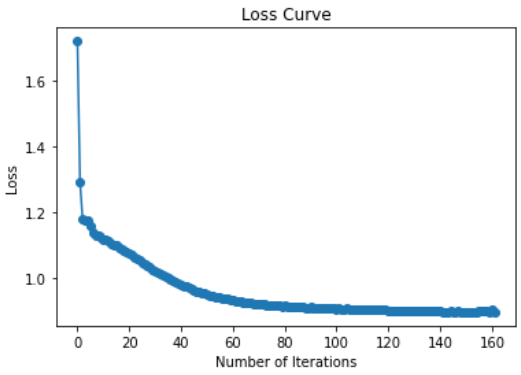

Fig.17 Loss curve for neural networks (contraceptive dataset)

Fig.17 shows the training loss function decreases as the number of iterations increases which is similar to the divorce dataset. The model also converges to a stable low value. Unlike the previous dataset, this loss curve drops sharply at the beginning when the number of iterations is small. It could be because this dataset is much larger than the previous one which means a lot more training data per iteration thus the accuracy improves quickly.

V. ADABOOST

Number of weak learners and learning rate are chosen as hyperparameters for tuning.

A. Divorce Predictors dataset

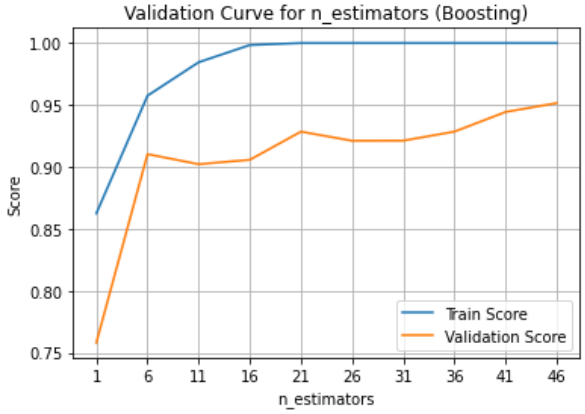

Fig.18 Validation curve for n_estimators (divorce dataset)

Observed from fig.18, the optimal number of weak learners should be 6 where the validation score is the highest but the training score has not yet been overfitted (i.e. not scores 1). Although both training and validation scores increase as the number of weak learners increase, the two scores diverge after n_estimators>6. It means the model starts fitting training data better without improvement in predicting unseen data. Any further increase of n_estimators only makes the training score too perfect but the validation score fluctuates between 0.9 and 0.95.

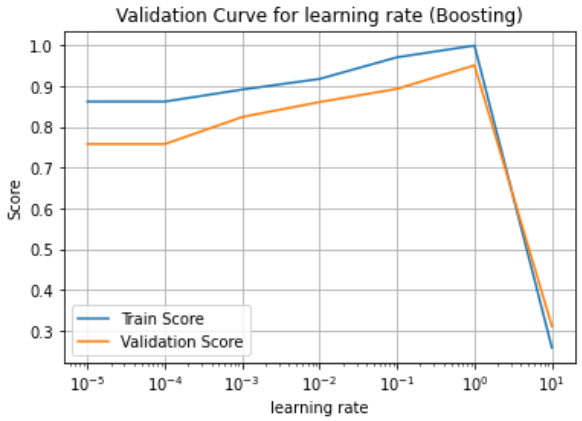

Fig.19 Validation curve for learning rate (divorce dataset)

Observed from fig.19, the optimal value of the learning rate should be 1 where both training and validation score converge and reach their highest points. It means the model

fits both training and unseen data well.

When the learning rate is super small (0.00001), both training and validation scores plateau. It might be because the weight of contribution for each learner is too small that it cannot get a significant reward/penalty for correct/wrong prediction.

As the learning rate increases closer to 1, both scores increase. But further increase for learning rate>1 causes the model to underfit as both scores plummeted quickly.

Grid search result shows the best value for number of weak learners and learning rate are 46 and 1 respectively, in which the number of weak learners is different from the optimal value found from single hyperparameter tuning.

Using grid search best params to train the model and plot the learning curve (fig.20):

Best params for AdaBoost:	{'learning_rate': 1.0, 'n_estimators': 46}
Time to fit:	28.418978452682495
Time to predict:	0.01726841926574707
F1_score for best AdaBoost:	0.0.8807017543859649

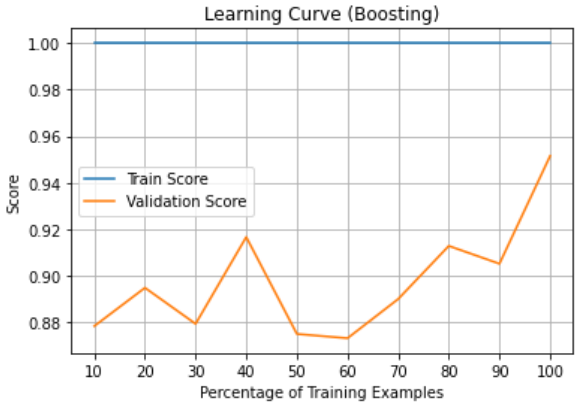

Fig.20 Learning curve for AdaBoost using grid search best params (divorce dataset)

Fig.20 shows the model always overfit the training data no matter what sample size is used. Thus another learning curve is plotted using the optimal values found from single hyperparameters tuning (fig.21):

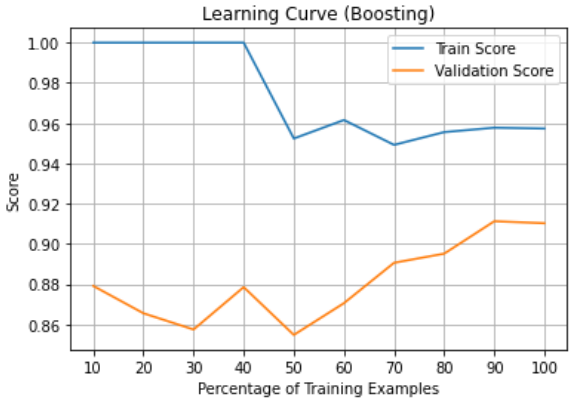

Fig.21 Learning curve for AdaBoost using optimal values from single hyperparameter tuning (divorce dataset)

Fig.21 shows a better AdaBoost model than the one used grid search best params because it generalizes as sample size increases. The F1-score for this model is 0.91 which is also better than the one from grid search best params.

B. Contraceptive Method Choice dataset

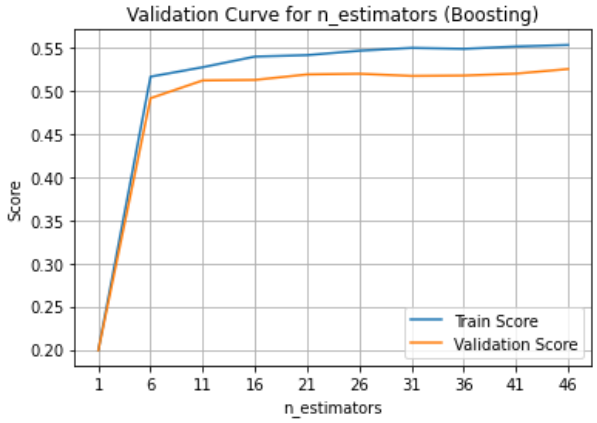

Fig.22 Validation curve for n_estimators (contraceptive dataset)

From fig.22, the optimal number of weak learners should be 11 where the gap between training and validation scores is the smallest while both scores are rising. The model's performance is too low when n_estimators=1 which means it underfits the data. But for n_estimators>11, both testing and validation scores plateau. It means adding more estimators does not help to improve the model's performance.

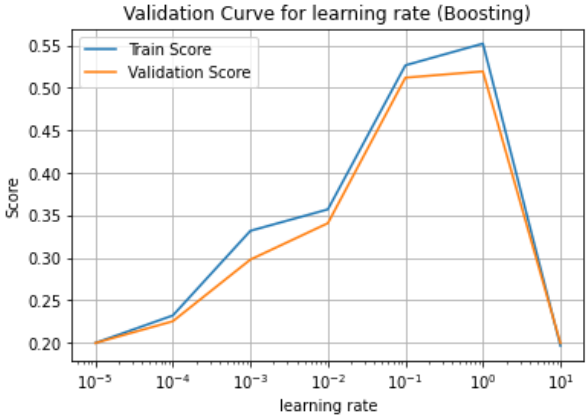

Fig.23 Validation curve for learning rate (contraceptive dataset)

From fig.23, the optimal number of weak learners should be 1 where both training and validation scores are at their highest points. Further increase for learning rate>1 causes the model to underfit as both scores plummeted.

Interestingly, grid search best params for this dataset is the same as the divorce dataset. But the number of weak learners found by grid search is different from the optimal

value found from single hyperparameter tuning. they Using grid search best params to train the model and plot the learning curve (fig.24):

Best params for boosting:	{'learning_rate': 1.0, 'n_estimators': 46}
Time to fit:	18.09411883354187
Time to predict:	0.013281822204589844
F1_score for best boosting:	0.5503947282573236

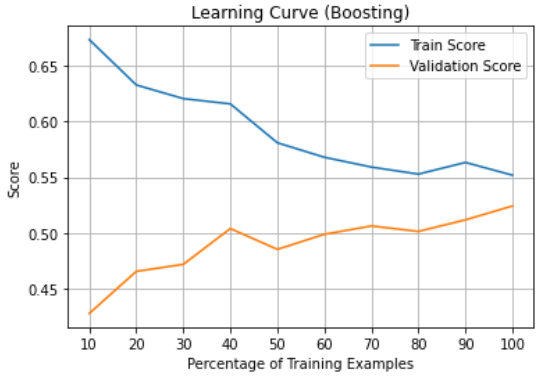

Fig.24 Learning curve for AdaBoost using grid search best params (contraceptive dataset)

Another learning curve is plotted using the optimal values found from single hyperparameters tuning (fig.25). F1-score for this model is 0.555 which is only slightly better than grid search best params (scores 0.550). Both training and validation scores converge to a lower score as sample size increases, which indicates the model gets little benefit from more training data.

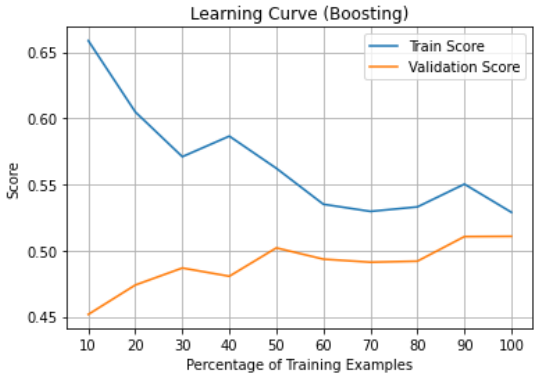

Fig.25 Learning curve for AdaBoost using optimal values from single hyperparameters tuning (contraceptive dataset)

VI. SUPPORT VECTOR MACHINES (SVM)

Kernels and C (regularization) are chosen as hyperparameters for tuning.

A. Divorce Predictors dataset

From fig.26, the optimal value of C should be 1 where both the training and validation scores converge to their highest points. It means the model predicts well on both training and unseen data. If C is too small (<0.001), the model underfits as both the training and validation scores are low. If C is too high (>10), the model overfits as the training score becomes 1 but validation score drops.

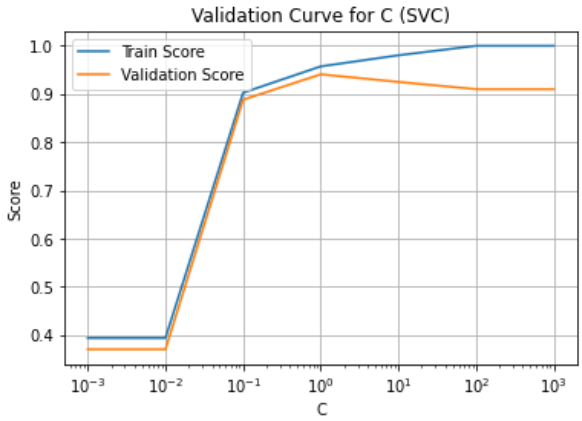

Fig.26 Learning curve for AdaBoost (divorce dataset)

Observed from fig.27, models using linear, polynomial and rbf kernels perform similarly well but the model using the sigmoid kernel performs poorly (underfits) for the divorce dataset.

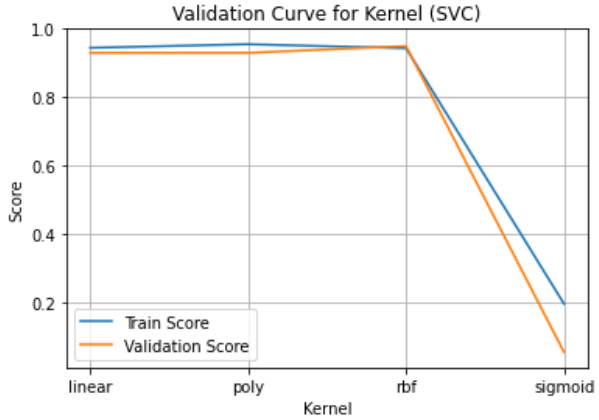

Fig.27 Learning curve for AdaBoost (divorce dataset)

Grid search shows the best params for C and kernel are 1 and rbf respectively, which match with the observation from single hyperparameter tuning.

Using grid search best params to training the model and plot the learning curve (fig.28):

Best params for SVM:	{'C': 1.0, 'kernel': 'rbf'}
Time to fit:	2.5997378826141357
Time to predict:	0.00636601448059082
F1_score for best SVM:	0.9705627705627706

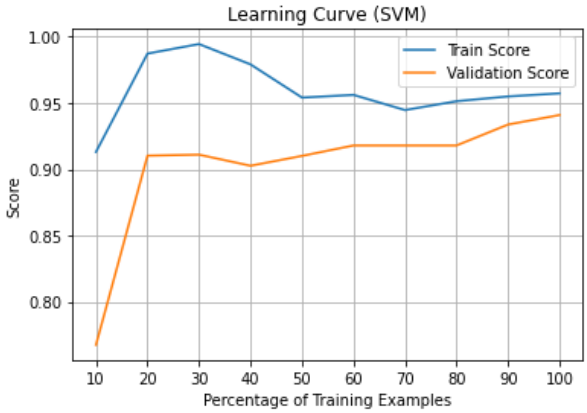

Fig.28 Learning curve for AdaBoost (divorce dataset)

Observed from fig.28, the huge gap at the beginning when sample size is small suggests that the model has high variance. It means the model overfits the training data but fails to predict unseen data (high variance). As more samples are fed to the model, the gap between training and validation scores becomes smaller and the validation score keeps increasing. It means the model predicts better on unseen data while not overfitting the training data.

B. Contraceptive Method Choice dataset

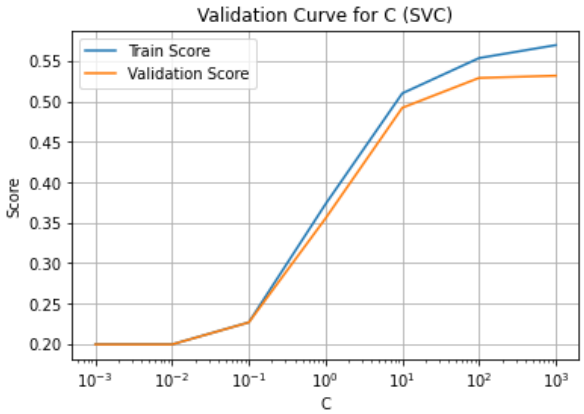

Fig.29 Learning curve for AdaBoost (contraceptive dataset)

Observed from fig.29, the optimal value of C should be 100 where both training and validation scores are still rising but have not diverged yet. For C >100, the validation score plateaus but the gap between two scores becomes larger. It means the model fits training data better than unseen data (starts to overfit).

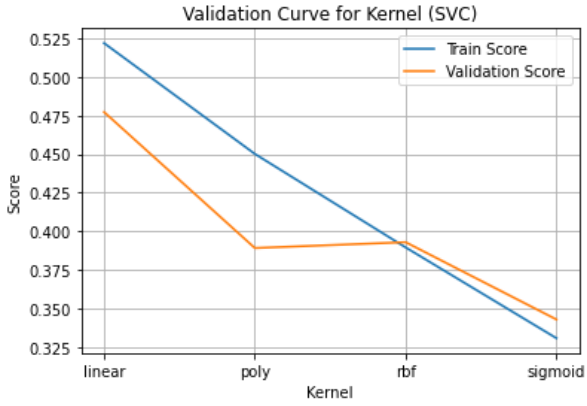

Fig.30 Learning curve for AdaBoost (contraceptive dataset)

Observed from fig.30, linear kernel scores better than the other three kernels. The sigmoid kernel performs the poorest as both training and validation scores are low (underfit). It could be because this dataset is a multiclass dataset while the sigmoid kernel works well on binary classification. Notice that all kernels score below 0.525 which is lower than the scores found from the previous training algorithms. It could mean that SVM is not a good model for this dataset.

Grid search shows the best params for C and kernel are 1000 and 'rbf' respectively, which is different from the observation of single hyperparameter tuning. F1-score of grid search best params model is 0.53, which is slightly better than the model trained using optimal values observed from single hyperparameter tuning (scores 0.51).

Using grid search to train the model and plot the learning curve (fig.31).

Best params from grid search:	{'C': 1000.0, 'kernel': 'rbf'}
Time to fit:	282.7405982017517
Time to predict:	0.02186894416809082
F1 score for best SVM:	0.5319662886272595

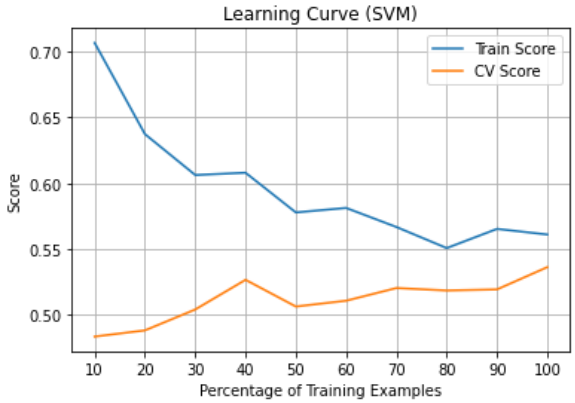

Fig.31 Learning curve for SVM (contraceptive dataset)

The SVM model does not benefit very much from more training samples as the training and the validation scores converge to a low value. It means the model still cannot predict unseen data very well given more training data provided.

VII. K-NEAREST NEIGHBORS (KNN)

Number of k-nearest-neighbors (K) and power parameters (p) for the Minkowski metric are chosen as the hyperparameters for tuning.

A. Divorce Predictors dataset

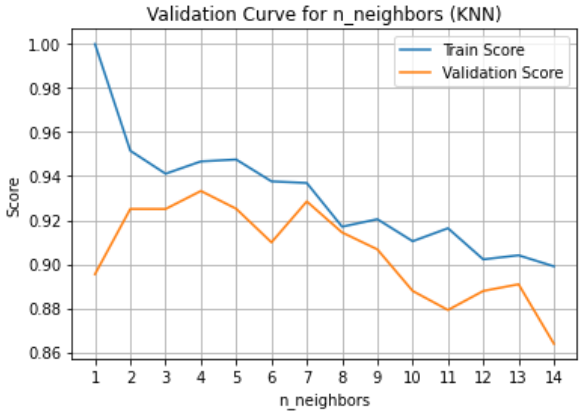

Fig.32 Validation curve for k-nearest-neighbors (divorce dataset)

From fig.32, the optimal number of K should be 4 where the validation score is the highest. K=1 means the learner takes the class of its closest neighbor. Thus it fits the training data too well but fails to predict unseen data. As K increases a bit, the model starts to generalize so that the validation score improves. But further increase for K>4, both validation and training score enter a downtrend which means the model underfits both training and unseen data.

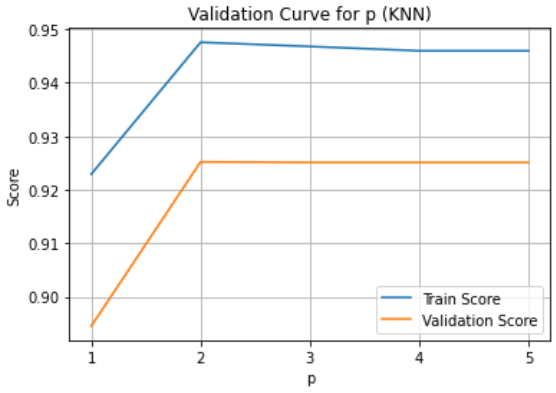

Fig.33 Validation curve for the power (p) of Minkowski distance (divorce dataset)

From fig.33, the KNN model performs similarly well for the powers of 2,3,4,5 except 1, which is the Manhattan distance. It means the model performs better in geometric distance than the distance in a grid.

Grid search shows best params for K and p are 4 and 3 respectively, which matches with the observation found in single hyperparameter tuning. In fact, all the best f1-scores for p=2,3,4 when K=4 are the same.

Using grid search to training the model and plot the learning curve (fig.32):

Best params for KNN:	{'n_neighbors': 4, 'p': 3}
Time to fit:	3.187072277069092
Time to predict:	0.006106376647949219
F1 score for best KNN:	0.9116883116883117

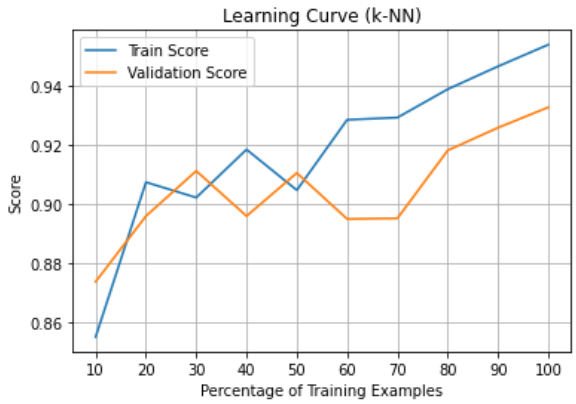

Fig.34 Learning curve for AdaBoost (divorce dataset)

From fig.34, the validation score is sometimes higher than the training score when sample size is small. Different values of p (2,3,4,5) were used and it turns out all the learning curves show higher validation scores than training scores when sample size is small. This dataset only contains 170 entries which leaves not many samples for testing after 80/20 train/test splits and 10-fold cross validation split (one sample left for validation if sample size is 10%). Thus the validation score is highly unreliable when sample size is small.

As the sample size increases, both training and validation scores rise which means the model benefits from more training data.

B. Contraceptive Method Choice dataset

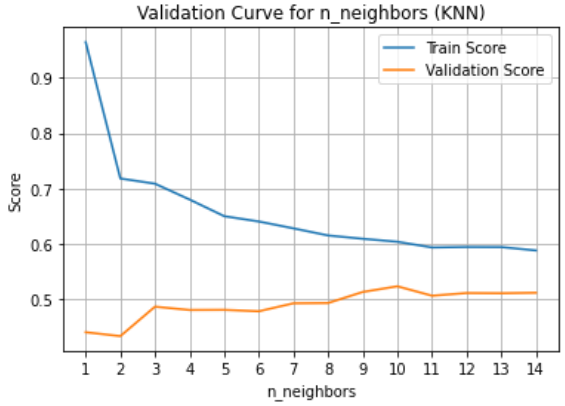

Fig.35 Validation curve for k-nearest-neighbors (contraceptive dataset)

From fig.35, the optimal value of K should be 10 since the validation score is the highest. But this KNN model is not performing well for this dataset because the validation

score stays around 0.5 no matter how many K used. The training score drops quickly as K increases, which means the model underfits as K increases.

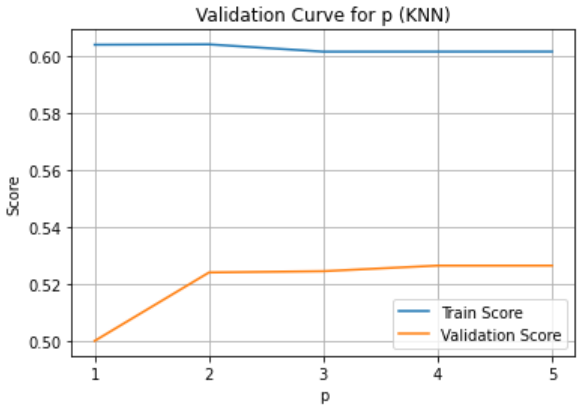

Fig.36 Validation curve for the power (p) of Minkowski distance (contraceptive dataset)

Fig.36 using K=10 to plot the validation curve for various p. Similar to the divorce dataset, powers of 2,3,4,5 except 1 performs similarly well. The reason why Manhattan distance performs poorly for both dataset would be that the dataset's dimension is not that high (i.e. not too many features are taken into account for prediction).

Grid search shows best params for K and p are 10 and 4 respectively, which matches with the findings in single hyperparameter tuning.

Using grid search best params to train the model and plot the learning curve (fig.37)

Best params for KNN:	{'n_neighbors': 10, 'p': 4}
Time to fit:	7.630703926086426
Time to predict:	0.03661656379699707
F1 score for best KNN:	0.5652253238460135

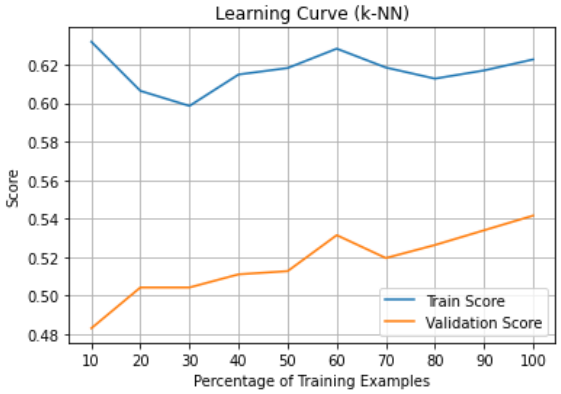

Fig.37 Learning curve for KNN (contraceptive dataset)

Unlike the previous dataset (divorce dataset), the learning curve of this dataset does not have moments where the validation score is higher than the training score. It could be because this dataset contains a lot more samples than the divorce dataset. The validation score rises gradually as the number of samples increases. It means the model benefits from more training samples. However, the gap between training and validation score is still large which indicates bias is still high even when the sample size is 100%.

VIII. WALL CLOCK TIME

A. Training time

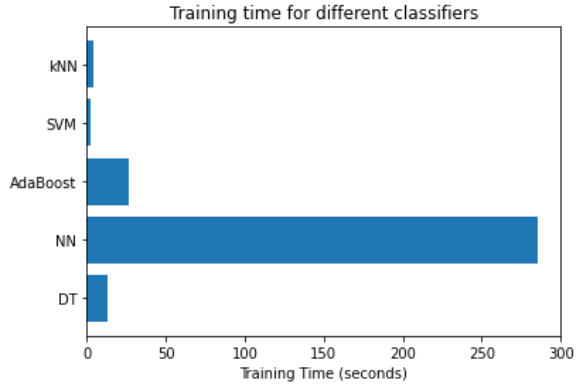

Fig.38 Training time for different classifiers (divorce dataset)

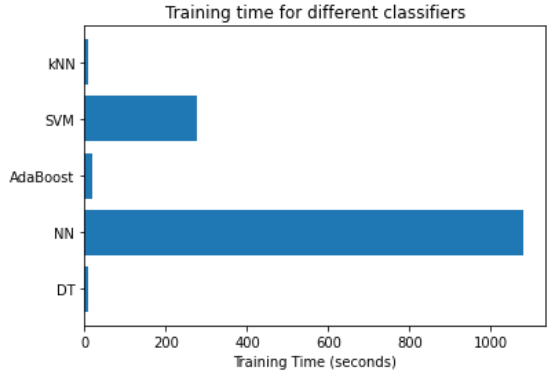

Fig.39 Training time for different classifiers (contraceptive dataset)

For both dataset, the neural networks model takes the longest time to train. As the learning rate for both dataset is small, the time for calculating gradient descent could be huge.

For divorce dataset, adaBoost takes the second highest time to train. It could be due to many weak learners being required to train the model thus more time is needed. SVM is the fastest to train because this dataset is small. The quadratic optimization can be done easily. KNN also trains fast because storing data points in grids is not computationally expensive.

For contraceptive dataset, SVM costs the second highest time to train. It could be due to the large dataset with three classification types. Thus more time needed for quadratic programming. Both the decision tree and KNN require the least training time. The tree is pruned and maximum depth is small so the training time is low. KNN trains fast because the model basically stores data points without heavy calculation.

B. Query time

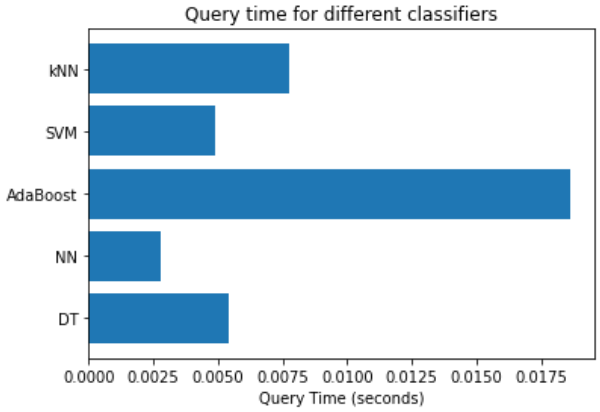

Fig.40 Query time for different classifiers (dirvorce dataset)

For divorce dataset (fig.40), adaBoost takes the longest query time because it contains 46 weak learners which is high enough to slow the query process. KNN takes quite a long time to query because it calculates distance for all the data points in order to find the k-nearest-neighbors. SVM and neural networks query fast because SVM query is solved by linear function while neural networks use a single forward pass for a query. Both computation costs are small.

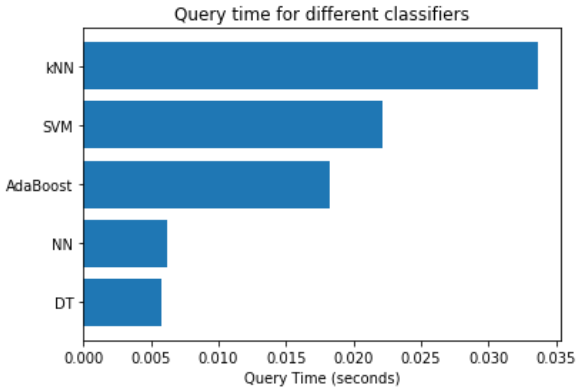

Fig.41 Query time for different classifiers (contraceptive dataset)

For contraceptive dataset (fig.41), KNN costs the highest query time because of the large dataset. Many data points are taken into account to find the closest K-neighbors. Both the neural networks and the decision tree query fast because the forward pass is fast for a query in neural networks and the traverse time is short for a decision tree with small tree depth.

X. CONCLUSION

Fig.42 and fig.43 compare the performance of all 5 algorithms.

For the divorce dataset (fig.42), SVM and neural networks perform the best. It could be due to the complexity of the neural networks and the data is non-linearly separable (rbf kernel is used). Adaboost performs the poorest could because too many weak learners are used (46 learners) which cause overfitting of the model.

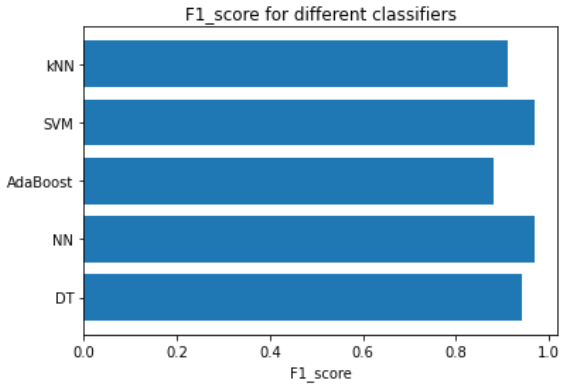

Fig.42 F1-score for different classifiers (divorce dataset)

For the contraceptive dataset (fig.43), KNN performs the best while SVM (rbf kernel) is the worst. It could be due to the multiclass nature of this dataset. The data points could be classified better by the k-nearest-neighbors surrounding them than a hyperplane with maximum margin.

In fact, all 5 algorithms score more or less the same (all scores around 0.5) which means no single model performs exceptionally well. It could be the dataset is using irrelevant features or other problems yet to be discovered.

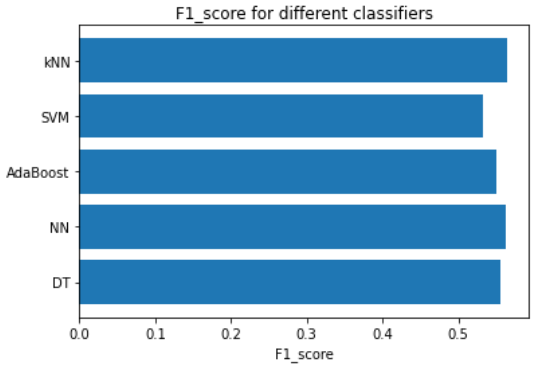

Fig.43 F1-score for different classifiers (contraceptive dataset)

X. REFERENCE

[1]	Mitchell, T. (1997). Machine Learning. McGraw-Hill Education.
[2]	Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly.
[3]	Guido, S., & Müller, A. C. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media, Incorporated.