**Abstract**

This report analyzes 5 supervised learning algorithms over two datasets.

GitHub Readme and repo – <https://github.com/Maimoons/7641/blob/master/README.md>

**Introduction**

Two labelled datasets were used to train 5 classifiers. All classifiers had a set of hyperparameters that were tuned with 5-fold cross validation technique. The effect on accuracy with the choice of hyperparameter were plotted in agraph. Cross validation was performed to prevent overfitting and selection bias of the small datasets and the final classifier with the best hyperparameters was trained and then tested on the held-out test set. Time to train the best model and time to test with the best model was recorded and compared across models.

**Datasets and Pre-processing**

Preprocessing is the first step in machine learning algorithms. Raw and unclean data do not offer as well as inferences as clean, standardized data do. Preprocessing was done to remove inconsistencies and noise and fill up the missing values.

1. **Breast Cancer dataset**:

This dataset was taken from the sklearn repository of datasets.

Table

Description automatically generated with medium confidence

*Statistics after preprocessing*

1. **Titanic dataset**:

The titanic dataset is a famous dataset taken from Kaggle’s list of datasets.

Table

Description automatically generated with medium confidence

*Statistics after preprocessing*

**Techniques for preprocessing–**

**Correlation analysis**: The correlation of the features was studied, and features with over 95% correlation were dropped to prevent overfitting. This was necessary since the amount of data was small and suspectable to overfitting and the breast cancer dataset had a lot of highly correlated features.

The columns dropped for breast cancer dataset were ['mean perimeter', 'mean area', 'perimeter error', 'area error', 'worst radius', 'worst perimeter', 'worst area']

**Feature Engineering:** The dataset was preprocessed to remove any null values. The breast cancer did not have any null values. The titanic dataset had two features with null values – Age and Fare. The Age had over 100 missing values whereas the Fare feature had a couple of missing values. They both had their mean filled out in the null rows to extrapolate from the existing data.

**Hot encoding**: Breast cancer had numerical values for all features and did not require non-numerical to numerical encoding. For titanic dataset, since there were not as many features, instead of dropping non numerical feature – sex, it was hot encoded into a 1/0 numerical value.

**Normalization:** For both the datasets, all features were normalized to model the data correctly and use a common scale across all features. This was also helpful in the case of neural networks which prevent the weights from over shooting.

**Oversampling:** Both the datasets were highly skewed in terms of the labels – Malignant and Benign for breast cancer dataset and Survived and not-Survived for titanic. Therefore, the technique of oversampling was performed and more records for the label in minority was extrapolated from the existing data to have equal balance of the two labels. This was again preferred over under sampling to prevent losing the already limited data.

**Experiments**

1. **Decision Trees**

The DecisionTreeClassifier was used from the sklearn library. This allowed us to abstract the core implementation and focus on tuning the hyperparameters for the classifier to perform better.

**Best hyperparameters:**

*BC: {'ccp\_alpha': 0.0, 'criterion': 'entropy', 'max\_depth': 5, 'min\_samples\_leaf': 1}*

*Titanic: 'ccp\_alpha': 0.0, 'criterion': 'entropy', 'max\_depth': 1, 'min\_samples\_leaf': 1}*

The grid search technique was performed on the above hyperparameters to ensure that all possible combinations were tried to pick the best combinations of the parameters.

**Validation Curves**

*Max depth:* This determined how deep and complex the tree was, including the number of minimum leaves at that depth.

*BC Titanic*

Diagram

Description automatically generatedChart

Description automatically generated

For both the datasets the training score increased as the depth increased. For the breast cancer dataset, the accuracy reached 1 close to around a depth of 7 whereas for titanic dataset that was a little later at around a depth of 15. For both the datasets the validation score decreased whereas the training score increased for lower depth suggesting that we were overfitting.

Diagram

Description automatically generatedChart, line chart

Description automatically generated

*Cost of pruning alpha:* The second hyperparameter that was tuned and plotted was the cost of pruning alpha. Pruning is used to prevent overfitting. The effective alphas were selected from the dataset itself using the sklearn’s complexity\_pruning\_path function.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

For the breast cancer dataset, the bigger value of pruning reduced both the training as well as the validation accuracy suggesting the model did worse on all accounts for larger alphas. However, for titanic dataset the highest validation accuracy is around an alpha of 0.03 suggesting it benefitted from pruning around that value and could generalize well. Pruning did not help immensely because we had smaller datasets to begin with.

**Learning Curve**

*BC*

Chart, line chart

Description automatically generatedChart

Description automatically generated

Titanic

Chart, line chart

Description automatically generated Chart

Description automatically generated

The cross-validation curve stayed below the training score throughout for both datasets, suggesting we were not generalizing as well as we were performing on the training set. Naturally, the trend was increased validation accuracy with more training data size. The model overfitting reduced as greater amount of data was available for the model to learn more inferences from and generalize better. For the titanic dataset, the sweet spot was close to 80% of the entire dataset. This could also have been due to the random, skewed split of the 80% dataset that was used to run the experiment.

1. **Boosted Decision Trees**

GradientBoostingClassifier was used from sklearn.ensemble to add boosting to the decision tree above.

**Best hyperparameters:**

*BC: {'learning\_rate': 0.7742636826811278, 'n\_estimators': 100}*

*Titanic: {'learning\_rate': 0.05994842503189409, 'n\_estimators': 20}*

**Validation Curves**

*Number of estimators*: This is the number of weak learner trees where each tree tried to correct the mistake of the previous tree.

Chart, line chart

Description automatically generatedA picture containing line chart

Description automatically generated

Adding a greater number of trees beyond a limit did not improve the performance. This can be observed for the titanic dataset where the highest validation error is around 20 trees beyond which the model generalizes poorly. The training score increased for both the datasets because more trees ensured that we are correcting the mistakes of the previous tree pertaining to the training dataset, so we endded up performing better on the data we are training.

*Learning rate:* This contributes to shrinking of the tree to prevent overfitting, converse of the number of estimators above.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

For the titanic dataset, around a learning rate of 1 is when the model was generalizing the best and the train accuracy is close to perfect too around the same value. For the breast cancer the most optimal learning rate is around the same range.

**Learning Curves**

BC

Chart, line chart

Description automatically generatedChart

Description automatically generated

Titanic

Chart, line chart

Description automatically generatedChart, treemap chart

Description automatically generated

The validation score increased with greater data trained, hence generalizing better. The training score more or less stayed a constant high suggesting the boosting algorithm made use of multiple weak learners trees to perform well on the smaller sized data too.

1. **Neural Networks**

Simple single layer neural network was used because of limited, less complex datasets.

**Best hyperparameters:**

*BC: {'activation': 'relu', 'hidden\_layer\_sizes': 40, 'learning\_rate\_init': 0.00035938136638046257, 'solver': 'adam'}*

*Titanic: : {'activation': 'relu', 'hidden\_layer\_sizes': 60, 'learning\_rate\_init': 0.0001, 'solver': 'adam'}*

*Hidden Units:* As the number of hidden units increased, training accuracy increased because there are more units to learn the train data’s patterns and features better. However, the model’s overfitting which increasing number of units can be seen by the decreasing overall validation error.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

*Learning Rate:* The best learning rate for both seem to be less than one. Small dataset requires small learning rate to prevent penalizing and changing model greatly.

Chart

Description automatically generatedChart, line chart

Description automatically generated

**Epochs**

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generated

The accuracy increased and the loss decreased with increasing epochs suggesting the model was performing better as we trained repeatedly. Within the 300 epochs the model has not overfit because the validation accuracy still appeared to be increasing.

**Learning Curves**

Chart, line chart

Description automatically generated Chart

Description automatically generated

Chart, line chart

Description automatically generated Chart, treemap chart

Description automatically generated

Neural networks need more data to learn better, and it is depicted by the increasing validation accuracy with greater train size. The train accuracy is high because the neural network can overfit and learn the train set well regardless.

1. **SVM**

Both the Regularization parameter C and kernel coefficient gamma were hyper tuned.

**Best hyperparameters:**

*BC: {'C': 0.9258747122872907, 'gamma': 0.0625, 'kernel': 'linear'}*

*Titanic: {'C': 1.3607900001743771, 'gamma': 0.09921256574801246, 'kernel': 'rbf'}*

*C:* Greater regularization led to greater train accuracy. Larger C leads to smaller margin and better classification on train data. The validation accuracy increased for the BC dataset but decreases for titanic dataset suggesting overfitting for the latter.

Chart

Description automatically generatedChart

Description automatically generated

*Gamma:* The train accuracy increased with greater gamma because the model could capture the complexity of the dataset better. This effect is steeper in the titanic dataset than the breast cancer dataset. The ability for the model to generalize however decreased with increasing gamma as the model was more heavily influenced by the train data.

Chart

Description automatically generated with medium confidence Chart

Description automatically generated

1. **KNN**

**Best hyperparameters:**

*BC: {'metric': 'minkowski', 'n\_neighbors': 1, 'p': 1, 'weights': 'uniform'}*

*Titanic: {'metric': 'canberra', 'n\_neighbors': 17, 'p': 1, 'weights': 'distance'}*

*Neighbors:* Increasing the number of neighbors decreased the train accuracy because we are losing more train data behavior by clustering them together, but the model generalized better with more neighbors as it also prevented overfitting.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

*Metric:* The BC dataset had a couple of high accuracy distance metric whereas for titanic the Canberra metric performed the best on the validation set.

Chart, line chart

Description automatically generated Chart

Description automatically generated

**Validation Curves**

Chart, line chart

Description automatically generated Chart, line chart

Description automatically generated

The train accuracy was independent of size, as we could learn well with smaller data too and small neighbors.

The ability for the model to generalize better rightfully increased with more training size.

**Time**

Chart, bar chart

Description automatically generated Chart, waterfall chart

Description automatically generated

Same trend for both datasets - The neural network appeared to have taken the most time to train. The forward and back propagation are costly on top of the low learning rate which results in long training process. The second highest is the boosted decision tree, where the costly operation is the gradient boosting itself and finding multiple weak learners. The greatest time to test was also high for neural network, followed by knn that needs to process the input against all the train data.

**Summary**

Accuracy comparison

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Chart, bar chart

Description automatically generated Chart, bar chart

Description automatically generated

Overall, the classifiers have greater accuracy over the breast cancer dataset than the titanic dataset and that makes sense since the BC dataset has significantly more features than Titanic to learn from even if comparable number of records. The best classifier for breast cancer is the neural network, which can learn better with greater number of features and the best classifier for the Titanic is the decision tree, that could lead to simpler trees and less overfitting with less number of features.

**References**

*[1]* [*https://melaniesoek0120.medium.com/breast-cancer-classification-machine-learning-1150498f18e2*](https://melaniesoek0120.medium.com/breast-cancer-classification-machine-learning-1150498f18e2)

*[2] https://www.geeksforgeeks.org/validation-curve/*

*[3]* [*https://www.analyticsvidhya.com/blog/2020/10/cost-complexity-pruning-decision-trees/*](https://www.analyticsvidhya.com/blog/2020/10/cost-complexity-pruning-decision-trees/)

*[4]* [*https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8*](https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8)

*[5]* [*https://amueller.github.io/aml/01-ml-workflow/03-preprocessing.html*](https://amueller.github.io/aml/01-ml-workflow/03-preprocessing.html)

*[6]* [*https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72*](https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72)

*[7] https://towardsdatascience.com/https-medium-com-janzawadzki-applying-andrew-ngs-1st-deep-neural-network-on-the-titanic-survival-data-set-b77edbc83816*