**Datasets**

The datasets are downloaded from the link <https://archive.ics.uci.edu/ml/index.php> which was posted by a student in Ed. The two datasets chosen are Car Evaluation and Adult. The car evaluation dataset was chosen as it was interesting to see the effects of the various parameters across the general population. However, the data size was a little unsatisfactory. The adult dataset was chosen to look at the impact of race and sex on income earned for the same occupation. This dataset also had a relatively large number of records to play around with. For both datasets, the features were limited to be able to run effective experiments.

1. Car Evaluation [1]

The dataset is derived from a simple hierarchical decision model originally developed for the demonstration of DEX, M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.).

This dataset will be referred to as dataset 1.

1. Adult [1]

The dataset extraction was done by Barry Becker from the 1994 Census database. A set of reasonably clean records was extracted using the following conditions: ((AAGE>16) && (AGI>100) && (AFNLWGT>1)&& (HRSWK>0)).

This dataset will be referred to as dataset 2.

Dataset 1 does not have as good results in the experiments as dataset 2 due to the lesser source data available for processing.

**Decision Tree**

The following links were used as references to code for the decision tree model:

1. <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>
2. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html>
3. <https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html>
4. <https://www.projectpro.io/recipes/plot-learning-curve-in-python>
5. <https://www.dataquest.io/blog/learning-curves-machine-learning/>
6. <https://ai.plainenglish.io/hyperparameter-tuning-of-decision-tree-classifier-using-gridsearchcv-2a6ebcaffeda>
7. <https://towardsdatascience.com/how-to-find-decision-tree-depth-via-cross-validation-2bf143f0f3d6>

As the training sample increases, there is marked improvement in the accuracy of the model and a reduction in bias. This is captured in the graphs depicting the Learning Curves of the decision tree model. This corresponds to a lower mean squared error as captured in the middle two graphs with MSE in the y-axis. The impact of overfitting along with the quality of split by adding gini (gini impurity) and entropy (information gain) as the hyper parameter in the decision tree is captured in the Validation Curves.

1. For dataset 1, the best two parameters are max depth of 7 and entropy as the criterion.
2. For dataset 2, the best two parameters are max depth of 7 and gini as the criterion.
3. For tree depth < 7, the model underfits.
4. Tree depth > 7 it starts to overfit. The training accuracy increases a lot for greater tree depths.

**Dataset 2 Dataset 1**

Chart

Description automatically generatedChart, scatter chart

Description automatically generatedChart

Description automatically generatedChart, scatter chart

Description automatically generatedChart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**AdaBoost**

The following links were used as references to code for the adaboost implementation:

1. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>
2. <https://www.datacamp.com/community/tutorials/adaboost-classifier-python>

As the training sample increases, the accuracy of the training set decreases and that of the validation set increases along with a reduction in bias. Gradually, they converge. This is captured in the graphs depicting the Learning Curves of the decision tree model after being boosted via the Adaboost classifier. This corresponds to the lower mean squared error in validation and increasing mean squared error in training as captured in the middle two graphs with MSE in the y-axis.

The two parameters with respect to hyper tuning the model considered here are the number of estimators and the learning rate. The number of estimators determines the number of weak learners and the learning rate determines the contribution of each classifier. The base estimator is defaulted to the decision tree.

As the number of estimators increase, the model is prone to overfitting. This can be seen in the validation curve of dataset 1. It is interesting to note that in dataset 2, the accuracy starts to drop at estimators around 130. The graph does not reflect that of dataset 1 and am unable to plot the downward trend of accuracy for even higher number of estimators similar to dataset 1.

1. For dataset 1, the best two parameters are learning rate = 0.25 and number of estimators = 11.
2. For dataset 2, the best two parameters are learning rate = 1.5 and number of estimators = 128.

**Dataset 2 Dataset 1**

Chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, scatter chart

Description automatically generatedChart, line chart

Description automatically generatedChart

Description automatically generatedChart, scatter chart

Description automatically generated

**Neural Network**

The following links were used as references to code for the neural network MLP model:

1. <https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html>
2. <https://www.tensorflow.org/api_docs/python/tf/keras/wrappers/scikit_learn>
3. <https://www.kaggle.com/residentmario/using-keras-models-with-scikit-learn-pipelines>
4. <https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/>
5. <https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/>

As the training sample increases, there is marked improvement in the accuracy of the model. This is captured in the graphs depicting the Learning Curves of the decision tree model. This corresponds to a lower mean squared error as captured in the middle two graphs with MSE in the y-axis. The impact of

Chart

Description automatically generatedGraphical user interface

Description automatically generated with low confidenceChart, scatter chart

Description automatically generatedChart

Description automatically generated

**References**

1. Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.