TESTING

CLASSIFICATION MODEL

Test with Iris Dataset, Wine Dataset and Breast Cancer dataset

**Iris.csv**

1. File upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Not uploaded file – I click save | Save has been disabled as nothing has been added | Save is disabled when no file has been added. |
| Not uploaded file – add config ‘browse files’ | Can add a .ini file that has been saved before and loads correctly | User is successfully able to load a previously saved config file in the ‘saves’ folder. All items load correctly and as expected. |
| Upload a csv file | Only allowed to pick a .csv file, correct errors are shown if the right file isn’t picked | Only CSV files are able to be picked, reducing the need for the error message for different file types. |
| Upload a csv file > 200mb | Correct error indicating max file size is shown to user | A warning shows, stating that the file must be 200mb or smaller. |
| Cross off and delete dataframe | Page refreshes back to original screen, indicating that the user needs to add a .csv file. | Deleting a csv file that has been added refreshes the page back to the original screen as expected. |
|  |  |  |
|  |  |  |

Data Exploration

1. Choose target column

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Pick a target column that is categorical | User only has the option to pick from categorical columns, this includes ‘bool’ columns that can be used for classification | With the iris dataset, the only target available is the only categorical column ‘species’. This is expected. |

1. Feature Scatterplot Analysis

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| X-axis: dropdown shows all numerical columns | Only numerical features are shown, this displays properly as the x-axis on the graph | All numerical columns are shown - all 4 features that are left in the dataset. |
| Y-axis: dropdown shows all numerical columns | Only numerical features are shown, this displays properly as the y-axis on the graph | The same 4 features are shown as expected. Plot axes are labelled as expected. |
| Plot analysis | Plot points are coloured based on the number of classes in the target column, and appropriately labelled | Iris classes are separated by their colour on the graph, and appropriately labelled. |
| Plot analysis – same X and Y axis | Shows a linear line | Using the same X and Y features for each axis shows a linear plot as expected. |
| Plot analysis – dataset with all numerical columns | Plot shows relationship between two features, but no colours are used on the plot | All columns are shown. If no categorical columns are found the code relies on numerical features. |

Model Configuration and Hyper-parameter Tuning

1. Max Depth

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Max-depth: Choose a max-depth value from the slider and observe a change in the resultant decision tree image | Depth of the decision tree visualisation dynamically corresponds to the number on the max-depth slider | Decision tree visualisation changes along with new updated values in the max depth slider. No errors occur on the Iris dataset. |
| Max-depth: max-depth value of 1 on slider | Underfitting warning shown, a tree with only 1 max depth is shown. | Warning is displayed underneath the slider indicating that low max depth levels can cause underfitting. The decision tree still updates and works to show the results as expected. |
| Max-depth: high max-depth value of 10 + | Overfitting warning is shown, full tree is shown with a warning explaining high rendering times | Warning is displayed underneath the slider indicating that high max depth levels can cause overfitting. The decision tree still updates and works to show the results as expected. |

1. Criterion Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Gini’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved gini value from the dropdown to fit the decision tree model |
| Choose ‘Entropy’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved entropy value from the dropdown to fit the decision tree model |
| Choose ‘Log\_loss’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved log\_loss value from the dropdown to fit the decision tree model |

1. Split Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Best’ | Tree and score updates using the relevant dropdown choice. ‘Best’ would generally have a higher test/train score. | The decision tree updates and uses the ‘best split’ criteria for the split from the dropdown to fit the decision tree model |
| Choose ‘Random’ | Tree and score updates using the relevant dropdown choice. | The decision tree updates and uses the ‘random split’ criteria for the split from the dropdown to fit the decision tree model |

1. Min Samples Split

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | When choosing 15 as the minimum split, the tree updates the decision tree with the new min samples per split value. In many cases this number does not make a difference in the scoring, but no error arises. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per split can lead to overfitting. Decision tree is still updated to reflect min sample split value. |

1. Min Samples Leaf

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Choosing 20 as the min samples per split. The decision tree and scores update to reflect new value. In many cases this number does not make a difference as much as max-depth value updates do. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per leaf can lead to overfitting. Decision tree is still updated to reflect min sample leaf value. |

1. Test Size

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Mid range number is chosen as 0.25. The scores and decision tree visualisation update to reflect the new chosen test split size as expected. |
| Choose small test size sample (0.05) | A warning is displayed, explaining why a small test size can lead to bad generalisation. The tree and scores are still updated to reflect these changes. | Small test split sizes display a warning underneath the slider. This message warns users that choosing small sample sizes may lead to weak generalisation of the model. |
| Choose large test size sample (0.5) | A warning is displayed, explaining why large training sets can lead to a bad model. The tree and scores are still updated to reflect these changes. | 0.5 test split size. A warning is displayed underneath the slider to indicate to the user that large test sizes will not provide enough training data to make good generalisations on the model. The visualisations and scores still update to reflect these changes. |

Model Training and Evaluation

Is the overfitting warning actually valid?

1. Decision Tree

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose different Max depth levels | Number of levels in the tree should match the max depth levels in the slider | Each change for the number in the max depth slider is successfully rendered in the decision tree visualisation. |
| Choose different min\_samples per leaf and split values | The decision tree should update and change to reflect any changes | Each change in both the sliders updates the decision tree visualisations. However the scores or tree do not change – these parameters may not change anything due to the size and simplicity of the dataset. |

1. Scores

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration criteria | Scores for training, test and generalisation should change and update | Each change in the parameter configuration causes an update for the training, test and generalisation scores. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). Check decision tree and visualisation to check this is accurate – is it really overfitting? | High training score, lower test score. Warning is displayed stating that the scores suggest overfitting | Since the dataset is so small, the training score and test scores are very easily very good. Generalisation scores are under <0.05. Training and test scores are high, no warning display is shown as it is hard to show overfitting within this model through scoring. |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). Check decision tree and visualisation to check this is accurate – is it really underfitting? | Low training and low test scores. Warning is displayed stating that the scores suggest underfitting | A warning is displayed under the scores, showing that the scores suggest underfitting. Both training and test scores are fairly low (0.667, 0.571). The generalisation score is over 0.095 suggesting that the scores do show a level of underfitting. The confusion matrix supports this. |
| Configure good model (check parameters through grid search CV). Check decision tree and visualisations to check this is accurate. | High training and high test score. Displays message suggesting that the scores show the model is a good fit. | High training score: 0.972, high test score: 0.978. A message is displayed underneath the scores suggesting this is a good model fit. This is supported by the confusion matrix. |

1. Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). | Confusion matrix displays correctly and updates dynamically to match the score, most predictions are incorrect. | A conventional overfitting model cannot be achieved with this data (high training, low test). Both high scores – the confusion matrix updates to show accurate predictions for each of the class labels. |
| Configure good model (grid search CV) | Confusion matrix displays correctly and updates dynamically to match the score, some predictions are incorrect. | The confusion matrix reflects the high scores accurately, with only one incorrect prediction. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). | Confusion matrix displays correctly and updates dynamically to match the score, little to no predictions are incorrect. | Overfitting model cannot be created with this dataset, but both values high (1.0) show all predictions are correct as expected. |

1. Cross Validation

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose all K values on the slider and test with balanced\_accuracy scoring | Mean, standard deviations and lists are dynamically updated to represent the number of K-Folds. | Cross validation with all k values on the slider (2-10) work, and update scorings and lists as expected. The mean and standard deviation re-renders and updates. The list length of cross validation scorings matches the k-fold value. |
| Check each scoring type for validity | Mean, standard deviations and lists are dynamically updated to represent the correct scoring type. | ‘f1’, ‘precision’, ‘recall’ show nan values in the list. This could be due to class imbalance. The rest of the scorings update appropriately and as expected. |

1. Cost Complexity Pruning

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure good model (grid search CV). Check CCP Alphas and graphs | If a better solution is found, the training and test scores will be higher for this decision tree. Decision tree visualisation is different from the configurated one (if improved scores). | The cpp alphas improve, and the training score improves. The test score however stays the same. The model has improved in this case, but generalisability stays the same. End decision tree is slightly more complex. |

Feature Selection and Importance

1. Feature Importance

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a decision tree model – repeat to see updates | Feature Importance plot should show in ranking value which feature is most used. Values in the list should correspond with graph. These should update with changes to the decision tree. | Change the test split size, and configuration parameters updates the feature importance list and plot. Both the graph and list and correctly ranked, with the right features displaying in both. |

1. Feature Selection

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check each value for slider to pick top-most important | The features chosen should match those in order of importance on the feature importance graph. Retrain classifier with all slider values should work | The feature selection process does not pick based on the correct ranking order, but the features chosen have been chosen as they have the most impact. |
| Feature selection plot | Should show correct number of numerical features available. | Shows the correct number and feature labels in the plot |

Analysis

1. GridSearchCV

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Insert letters, negative numbers, float numbers to the Max depth and Min samples per leaf inputs. | This should return an error message to the user indicating only valid natural numbers are allowed. | Any invalid datatypes such as strings ‘a’, float values ‘2.64’ and negative numbers ‘-1’ are not added to the list as expected. A message arises stating these aren’t valid and are excluded from the resultant lists. |
| Values for the inputs section in the list | These values should match those added into the inputs by the user. If many are added, the user must be warned about training times. | If the datatypes are valid natural numbers, the values in the list are shown as expected. |
| Scores and Visualisations | After a valid grid search a visualisation of the best model tree, the best model parameters and scores are displayed correctly. | After clicking the ‘perform grid search’ button, the correct parameters, scores and decision tree visualisation of the best model is shown as expected. |

Save

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a changed model with complete inputs. | The save successfully adds the correct values to an ini file on the desktop. | Correct values are added to the new saved .ini file in the ‘saves’ directory. |
| Save a model with no changes (file has been added) | The default model is saved successfully to an ini file on the desktop | Opening the iris dataset, all the default values are created and is saved as a .ini file with all the correct info. |

Upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Click on upload button, directs the user to .ini files. The user can upload a previously saved file. | If a user has saved a .ini file, once the file has been uploaded all details are the same as when saved as expected. The user can also make any changes and save again to update an changes in the model. | The user can upload and browse previously saved files. All info is shown as correct, ‘perform grid search’ and ‘retrain with selected features’ are not fully shown due to the session states. Information saved prior are retained to create the same results. |
| Replace dataframe with another .csv file (after a config file has been loaded) | Refreshes to display new dataframe, previous dataframe info from uploaded config is not saved. | Page does not refresh. States that a new dataframe has been uploaded, but the uploaded config file data has overridden CSV file upload. |

**Winequality-red.csv / wine.csv**

Data Exploration

1. Choose target column

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Pick a target column that is categorical | User only has the option to pick from categorical columns, this includes ‘bool’ columns that can be used for classification. If there are no categorical, allow choice of ‘numerical’ features. | There are no defined categorical features for the dataset. Because of this the chosen features default to the last feature – quality, and all numerical features are shown. At this point It is up to the user to define the correct target. |

1. Feature Scatterplot Analysis

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| X-axis: dropdown shows all numerical columns | Only numerical features are shown, this displays properly as the x-axis on the graph | All numerical features for the dataset are shown, they are displayed correctly on the scatterplot graph |
| Y-axis: dropdown shows all numerical columns | Only numerical features are shown, this displays properly as the y-axis on the graph | All numerical features for the dataset are shown, they are displayed correctly on the scatterplot graph |
| Plot analysis | Plot points are coloured based on the number of classes in the target column, and appropriately labelled | Plots are coloured based on the different classes in the ‘quality’ feature. These are appropriately labelled. |
| Plot analysis – same X and Y axis | Shows a linear line | Correctly show a linear plot |

Model Configuration and Hyper-parameter Tuning

1. Max Depth

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Max-depth: Choose a max-depth value from the slider and observe a change in the resultant decision tree image | Depth of the decision tree visualisation dynamically corresponds to the number on the max-depth slider | Each max depth value change dynamically changes the decision tree visualisations. Max depth values of 6 + break the code – this seems to be a Dtreeviz library problem. |
| Max-depth: max-depth value of 1 on slider | Underfitting warning shown, a tree with only 1 max depth is shown. | An overfitting warning is shown, and the decision tree visualisation updates to match the max-depth number. |
| Max-depth: high max-depth value of 10 + | Overfitting warning is shown, full tree is shown with a warning explaining high rendering times | Max-depth values of 6+ break the decision tree visualisation |

1. Criterion Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Gini’ | Tree and score updates using the relevant dropdown choice | Tree and score updates to the ‘gini’ criteria successfully. |
| Choose ‘Entropy’ | Tree and score updates using the relevant dropdown choice | The tree and score visualisations refresh to match the ‘entropy’ criteria |
| Choose ‘Log\_loss’ | Tree and score updates using the relevant dropdown choice | Tree and scores successfully update to match the ‘log\_loss’ criteria |

1. Split Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Best’ | Tree and score updates using the relevant dropdown choice. ‘Best’ would generally have a higher test/train score. | The decision tree updates and uses the ‘best split’ criteria for the split from the dropdown to fit the decision tree model |
| Choose ‘Random’ | Tree and score updates using the relevant dropdown choice. | The decision tree updates and uses the ‘random split’ criteria for the split from the dropdown to fit the decision tree model |

1. Min Samples Split

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | The mid-range value (20 and 50) for minimum samples align with the decision tree and scores. No change is seen at node or score level. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Overfitting warning is shown for low min\_samples\_split values under the slider. No change for min\_samples\_split is seen at score or node level. |

1. Min Samples Leaf

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Choosing 20 as the min samples per split. The decision tree and scores update to reflect new value. In many cases this number does not make a difference as much as max-depth changes do. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per leaf can lead to overfitting. Decision tree is still updated to reflect min sample leaf value. |

1. Test Size

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Tree visualisations and scores align with the new test size given and are updated dynamically as expected. |
| Choose small test size sample (0.05) | A warning is displayed, explaining why a small test size can lead to bad generalisation. The tree and scores are still updated to reflect these changes. | Small test split sizes display a warning underneath the slider. This message warns users that choosing small sample sizes may lead to weak generalisation of the model. |
| Choose large test size sample (0.5) | A warning is displayed, explaining why large training sets can lead to a bad model. The tree and scores are still updated to reflect these changes. | Large test split sizes display a warning underneath the slider. This message warns users that choosing large test split sizes may lead to lack of training for generalisability. |

Model Training and Evaluation

Is the overfitting warning actually valid?

1. Decision Tree

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose different Max depth levels | Number of levels in the tree should match the max depth levels in the slider | Each max depth value change dynamically changes the decision tree visualisations. Max depth values of 6 + break the code – this seems to be a Dtreeviz library problem. |
| Choose different min\_samples per leaf and split values | The decision tree should update and change to reflect any changes | Each change in both the sliders updates the decision tree visualisations. However the scores or tree do not change – these parameters may not change anything due to the size and simplicity of the dataset. |

1. Scores

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration criteria | Scores for training, test and generalisation should change and update | Changes in the parameter configuration causes an update for the training, test and generalisation scores. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). Check decision tree and visualisation to check this is accurate – is it really overfitting? | High training score, lower test score. Warning is displayed stating that the scores suggest overfitting | Overfitting model (training score: 1.0, testing score: 0.327). From looking at the decision tree visualisation it is clear the tree if overfitting as the tree is large and over 10 max depths. The warning is displayed beneath the scores stating that the scores suggest overfitting. |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). Check decision tree and visualisation to check this is accurate – is it really underfitting? | Low training and low test scores. Warning is displayed stating that the scores suggest underfitting | Underfitting model, both scores are low. Training: 0.26, test: 0.209. From looking at the confusion matrix the model is underfitting. A warning is displayed under the scores stating the scores suggest underfitting. |
| Configure good model (check parameters through grid search CV). Check decision tree and visualisations to check this is accurate. | High training and high test score. Displays message suggesting that the scores show the model is a good fit. | A different preprocessed wine dataset was used for the good model. Subset the ‘quality’ feature into classes for good and bad. No scores for both training and test were over 0.8, ones with 0.7 displayed a message stating that the model may need some improving. |

1. Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). | Confusion matrix displays correctly and updates dynamically to match the score, most predictions are incorrect. | Confusion matrix updates, and low scores align with the visualisation, where many predictions are mislabelled. |
| Configure good model (grid search CV) | Confusion matrix displays correctly and updates dynamically to match the score, some predictions are incorrect. | A decent model, with scores of 0.7 for both training and test aligns with the confusion matrix. Some predictions are wrong, more are correct. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). | Confusion matrix displays correctly and updates dynamically to match the score, little to no predictions are incorrect. | An overfitting model, some predictions in the confusion matrix are still wrong as expected. The percentage correct matches that of the test score. |

1. Cross Validation

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose all K values on the slider and test with balanced\_accuracy scoring | Mean, standard deviations and lists are dynamically updated to represent the number of K-Folds. | Cross validation with all k values on the slider (2-10) work, and update scorings and lists as expected. The mean and standard deviation re-renders and updates. The list length of cross validation scorings matches the k-fold value. |
| Check each scoring type for validity | Mean, standard deviations and lists are dynamically updated to represent the correct scoring type. | All scoring types work for the cross validation. The mean, cross validation scores and standard deviations update dynamically. |

1. Cost Complexity Pruning

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure good model (grid search CV). Check CCP Alphas and graphs | If a better solution is found, the training and test scores will be higher for this decision tree. Decision tree visualisation is different from the configurated one (if improved scores). | A decent model is configured (0.7 scores for both). The CCP model and decision tree improves the score for the training at 1.0 with test as 0.76. This suggests the CCP model has found a better solution but is prone to overfitting. The decision tree visualisation is large and complex, supporting this. |

Feature Selection and Importance

1. Feature Importance

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a decision tree model – repeat to see updates | Feature Importance plot should show in ranking value which feature is most used. Values in the list should correspond with graph. These should update with changes to the decision tree. | Change the test split size, and configuration parameters updates the feature importance list and plot. Both the graph and list and correctly ranked, with the right features displaying in both. |

1. Feature Selection

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check each value for slider to pick top most important | The features chosen should match those in order of importance on the feature importance graph. Retrain classifier with all slider values should work | After changing the slider values for features to select, they match the feature importance plots in order of ranking as expected. |
| Feature selection plot | Should show correct number of numerical features available. | The feature selection plot shows the correct number of numerical features available in the dataset. |

Analysis

1. GridSearchCV

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Insert letters, negative numbers, float numbers to the Max depth and Min samples per leaf inputs. | This should return an error message to the user indicating only valid natural numbers are allowed. | Any invalid datatypes such as strings ‘a’, float values ‘2.64’ and negative numbers ‘-1’ are not added to the list as expected. A message arises stating these aren’t valid and are excluded from the resultant lists. |
| Values for the inputs section in the list | These values should match those added into the inputs by the user. If many are added, the user must be warned about training times. | If the datatypes are valid natural numbers, the values in the list are shown as expected. |
| Scores and Visualisations | After a valid grid search a visualisation of the best model tree, the best model parameters and scores are displayed correctly. | After clicking the ‘perform grid search’ button, the correct parameters, scores and decision tree visualisation of the best model is shown as expected. |

Save

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a changed model with complete inputs. Click save. | The save successfully adds the correct values to an ini file on the desktop. | Correct values are added to the new saved .ini file in the ‘saves’ directory. |
| Save a model with no changes (file has been added) | The default model is saved successfully to an ini file on the desktop | Opening the wine dataset, all the default values are created and is saved as a .ini file with all the correct info. |

Upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Click on upload button, directs the user to .ini files. The user can upload a previously saved file. | If a user has saved a .ini file, once the file has been uploaded all details are the same as when saved as expected. The user can also make any changes and save again to update an changes in the model. | The user can upload and browse previously saved files. All info is shown as correct, ‘perform grid search’ and ‘retrain with selected features’ are not fully shown due to the session states. Information saved prior are retained to create the same results. |

**Breast-cancer.csv**

Data Exploration

1. Choose target column

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Pick a target column that is categorical | User only has the option to pick from categorical columns, this includes ‘bool’ columns that can be used for classification | The dataset loading successfully opens the target column in the dropdown ‘diagnosis’ |

1. Feature Scatterplot Analysis

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| X-axis: dropdown shows all numerical columns | Only numerical features are shown, this displays properly as the x-axis on the graph | X-axis shows all numerical features available in the dataset and are displayed appropriately in the resultant scatterplot graph. |
| Y-axis: dropdown shows all numerical columns | Only numerical features are shown, this displays properly as the y-axis on the graph | Y-axis shows all numerical features available in the dataset and are displayed appropriately in the resultant scatterplot graph. |
| Plot analysis | Plot points are coloured based on the number of classes in the target column, and appropriately labelled | Plot points are correctly coloured under the labels ‘M’ and ‘B’ – the classes shown in the diagnosis feature. |
| Plot analysis – same X and Y axis | Shows a linear line | The same X and Y axis features correctly show a linear plot |

Model Configuration and Hyper-parameter Tuning

1. Max Depth

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Max-depth: Choose a max-depth value from the slider and observe a change in the resultant decision tree image | Depth of the decision tree visualisation dynamically corresponds to the number on the max-depth slider | Decision tree visualisation changes along with new updated values in the max depth slider. No errors occur on the breast-cancer dataset. |
| Max-depth: max-depth value of 1 on slider | Underfitting warning shown, a tree with only 1 max depth is shown. | Warning is displayed underneath the slider indicating that low max depth levels can cause underfitting. The decision tree still updates and works to show the results as expected. |
| Max-depth: high max-depth value of 10 + | Overfitting warning is shown, full tree is shown with a warning explaining high rendering times | Warning is displayed underneath the slider indicating that high max depth levels can cause overfitting. The decision tree still updates and works to show the results as expected. |

1. Criterion Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Gini’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved gini value from the dropdown to fit the decision tree model |
| Choose ‘Entropy’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved entropy value from the dropdown to fit the decision tree model |
| Choose ‘Log\_loss’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved log\_loss value from the dropdown to fit the decision tree model |

1. Split Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Best’ | Tree and score updates using the relevant dropdown choice. ‘Best’ would generally have a higher test/train score. | The decision tree updates and uses the ‘best split’ criteria for the split from the dropdown to fit the decision tree model |
| Choose ‘Random’ | Tree and score updates using the relevant dropdown choice. | The decision tree updates and uses the ‘random split’ criteria for the split from the dropdown to fit the decision tree model |

1. Min Samples Split

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | When choosing 15 as the minimum split, the tree updates the decision tree with the new min samples per split value. In many cases this number does not make a difference in the scoring, but no error arises. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per split can lead to overfitting. Decision tree is still updated to reflect min sample split value. |

1. Min Samples Leaf

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Choosing 20 as the min samples per split. The decision tree and scores update to reflect new value. In many cases this number does not make a difference as much as max-depth changes do. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per leaf can lead to overfitting. Decision tree is still updated to reflect min sample leaf value. |

1. Test Size

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Tree visualisations and scores align with the new test size given and are updated dynamically as expected. |
| Choose small test size sample (0.05) | A warning is displayed, explaining why a small test size can lead to bad generalisation. The tree and scores are still updated to reflect these changes. | Small test split sizes display a warning underneath the slider. This message warns users that choosing small sample sizes may lead to weak generalisation of the model. |
| Choose large test size sample (0.5) | A warning is displayed, explaining why large training sets can lead to a bad model. The tree and scores are still updated to reflect these changes. | Large test split sizes display a warning underneath the slider. This message warns users that choosing large test split sizes may lead to lack of training for generalisability. |

Model Training and Evaluation

Is the overfitting warning actually valid?

1. Decision Tree

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose different Max depth levels | Number of levels in the tree should match the max depth levels in the slider | Each change for the number in the max depth slider is successfully rendered in the decision tree visualisation. |
| Choose different min\_samples per leaf and split values | The decision tree should update and change to reflect any changes | Each change in both the sliders updates the decision tree visualisations. However the scores or tree do not change – these parameters may not change anything due to the size or simplicity of the dataset. |

1. Scores

|  |  |  |
| --- | --- | --- |
| **Scores** | | |
| Action | Expected Outcome | Outcome |
| Change configuration criteria | Scores for training, test and generalisation should change and update | Each change in the parameter configuration causes an update for the training, test and generalisation scores. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). Check decision tree and visualisation to check this is accurate – is it really overfitting? | High training score, lower test score. Warning is displayed stating that the scores suggest overfitting | Training score is 0.98, test score is 0.86. Since the generalisation score is higher than expected, the user receives a warning displaying that the scores suggest overfitting in the model. Other instances where the training score is high e.g. 0.96 and the test is 0.88 suggest there could be some element of overfitting. Instead the model displays a message stating the model may need some improving. Could this be seen as overfitting instead? |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). Check decision tree and visualisation to check this is accurate – is it really underfitting? | Low training and low test scores. Warning is displayed stating that the scores suggest underfitting | When trying to purposely underfit the model, the scores are still relatively high at 0.7. Instead of displaying an underfitting warning from looking at the scores, it instead suggests that the model may need improving. |
| Configure good model (check parameters through grid search CV). Check decision tree and visualisations to check this is accurate. | High training and high test score. Displays message suggesting that the scores show the model is a good fit. | High training score: 0.913, high test score: 0.897. A message is displayed underneath the scores suggesting this is a good model fit. This is supported by the confusion matrix, few are misdiagnosed without overfitting. |

1. Confusion Matrix

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). | Confusion matrix displays correctly and updates dynamically to match the score, most predictions are incorrect. | A conventional underfitting model cannot be achieved with this data (lower than 0.5). At 0.7, score predictions are depicted correctly– the confusion matrix updates to show accurate predictions for each of the class labels. |
| Configure good model (grid search CV) | Confusion matrix displays correctly and updates dynamically to match the score, some predictions are incorrect. | The confusion matrix reflects the high scores accurately, with fewer incorrect predictions. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). | Confusion matrix displays correctly and updates dynamically to match the score, little to no predictions are incorrect. | Overfitting model cannot be created with this dataset, but both values high (0.9+) show most predictions are correct as expected. |

1. Cross Validation

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose all K values on the slider and test with balanced\_accuracy scoring | Mean, standard deviations and lists are dynamically updated to represent the number of K-Folds. | Cross validation with all k values on the slider (2-10) work, and update scorings and lists as expected. The mean and standard deviation re-renders and updates. The list length of cross validation scorings matches the k-fold value. |
| Check each scoring type for validity | Mean, standard deviations and lists are dynamically updated to represent the correct scoring type. | All scoring types work for the cross validation. The mean, cross validation scores and standard deviations update dynamically. |

1. Cost Complexity Pruning

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure good model (grid search CV). Check CCP Alphas and graphs | If a better solution is found, the training and test scores will be higher for this decision tree. Decision tree visualisation is different from the configurated one (if improved scores). | A good model is configured (0.9+ scores for both). The CCP model reduces the max depth value and increases the overall training score and test score values (0.991, 0.903). This suggests the CCP model has found a better solution but could prone to overfitting. |

Feature Selection and Importance

1. Feature Importance

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a decision tree model – repeat to see updates | Feature Importance plot should show in ranking value which feature is most used. Values in the list should correspond with graph. These should update with changes to the decision tree. | Change the test split size, and configuration parameters updates the feature importance list and plot. Both the graph and list and correctly ranked, with the right features displaying in both. |

1. Feature Selection

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check each value for slider to pick top most important | The features chosen should match those in order of importance on the feature importance graph. Retrain classifier with all slider values should work | After changing the slider values for features to select, the most important features are recognised and added first to the feature selection. At times, those not indicated as the most important in the model are important in the RFE and used for the feature selection. Shows there are discrepancies between the plot and the RFE algorithm for identifying best features. |
| Feature selection plot | Should show correct number of numerical features available. | The feature selection plot shows the correct number of numerical features available in the dataset. |

Analysis

1. GridSearchCV

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Insert letters, negative numbers, float numbers to the Max depth and Min samples per leaf inputs. | This should return an error message to the user indicating only valid natural numbers are allowed. | Any invalid datatypes such as strings ‘a’, float values ‘2.64’ and negative numbers ‘-1’ are not added to the list, as is expected. A message arises stating these aren’t valid and are excluded from the resultant lists. |
| Values for the inputs section in the list | These values should match those added into the inputs by the user. If many are added, the user must be warned about training times. | If the datatypes are valid natural numbers, the values in the list are shown as expected. |
| Scores and Visualisations | After a valid grid search a visualisation of the best model tree, the best model parameters and scores are displayed correctly. | After clicking the ‘perform grid search’ button, the correct parameters, scores and decision tree visualisation of the best model is shown as expected. |

Save

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a changed model with complete inputs. Save as ‘test\_1.ini’ | The save successfully adds the correct values to an ini file on the desktop. | Correct values are added to the new saved .ini file in the ‘saves’ directory. |
| Save a model with no changes (file has been added) | The default model is saved successfully to an ini file on the desktop | Opening the wine dataset, all the default values are created and is saved as a .ini file with all the correct info. |

Upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Click on upload button, directs the user to .ini files. The user can upload a previously saved file. | If a user has saved a .ini file, once the file has been uploaded all details are the same as when saved as expected. The user can also make any changes and save again to update an changes in the model. | The user can upload and browse previously saved files. All info is shown as correct, ‘perform grid search’ and ‘retrain with selected features’ are not fully shown due to the session states. Information saved prior are retained to create the same results. |

Tests on Classification Datasets:

Iris.csv – why? evaluate how well the decision tree classifier can handle a simple, multi-class classification problem.

Check that all features work:

*Target column*

*Feature Scatterplot Analysis*

*Decision Tree Visualisation*

*Scores*

*Confusion Matrix*

*Cross Validation*

*Feature Importance*

*Feature Selection*

*Cross Complexity Pruning*

*Grid Search CV*

Displaying correctly

Overfitting and Underfitting scores

Graphs display as expected

Wine.csv – why? This dataset is perfect for testing how well decision trees can handle multi-class classification with continuous features. It has more features and a bit more complexity than the Iris dataset, but still manageable for testing classification algorithms like decision trees.

BreastCancer.csv – why? You can use this dataset to test how well the decision tree classifier performs in a healthcare-related binary classification problem, especially with feature importance analysis. It is an example of a binary classification problem with real-world medical data, making it a bit more complex and suitable for evaluating the robustness of your decision tree model.

PERFORMANCE METRICS

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | CPU | Memory | Load Time |
| Iris.csv |  |  |  |
| Winequality-red.csv |  |  |  |
| Breast\_cancer.csv |  |  |  |

OR

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Max\_depth | Max\_depth | Load Time |
| Iris.csv | 3 | 10 |  |
| Winequality-red.csv/wine.csv |  |  |  |
| Breast\_cancer.csv |  |  |  |

Regression Testing

 **Boston Housing Dataset**: Ideal for testing the performance of your decision tree regressor on real estate price predictions with mixed feature types.

 **Diabetes Dataset**: Useful for testing a smaller dataset with physiological data and a continuous target variable.

 **California Housing Dataset**: Provides a large and complex dataset to evaluate the scalability and performance of your regression model.

**Diabetes-cleaned.csv**

1. File upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Not uploaded file – I click save | Save has been disabled as nothing has been added | Save is disabled when no file has been added, until a file has been added and the page fully loaded. |
| Not uploaded file – add config ‘browse files’ | Can add a .ini file that has been saved before and loads correctly | User is successfully able to load a previously saved config file in the ‘saves’ folder. All items load correctly and as expected. |
| Upload a csv file | Only allowed to pick a .csv file, correct errors are shown if the right file isn’t picked | Only CSV files are able to be picked, reducing the need for the error message for different file types. |
| Upload a csv file > 200mb | Correct error indicating max file size is shown to user | A warning shows, stating that the file must be 200mb or smaller. |
| Cross off and delete dataframe | Page refreshes back to original screen, indicating that the user needs to add a .csv file. | Deleting a csv file that has been added refreshes the page back to the original screen as expected. |

Data Exploration

1. Choose target column

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Pick a target column that is numerical | User only has the option to pick from numerical columns for the target. | With the diabetes-cleaned dataset, all features are shown, this includes the ‘outcome’ feature. This is treated as numerical since the feature is an ‘int64’ feature type. |

1. Boxplot

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check feature labels | All feature labels are labelled correctly. | All features are labelled correctly on the plot. |

1. Correlation Matrix

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check feature labels and plot | All feature labels are labelled correctly, and plot is displayed clearly. | All features are labelled correctly on the plot, and the plot can be enlarged to be seen clearly. |

Model Configuration and Hyper-parameter Tuning

1. Max Depth

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Max-depth: Choose a max-depth value from the slider and observe a change in the resultant decision tree image | Depth of the decision tree visualisation dynamically corresponds to the number on the max-depth slider | Decision tree visualisation changes along with new updated values in the max depth slider. No errors occur on the diabetes dataset. |
| Max-depth: max-depth value of 1 on slider | Underfitting warning shown, a tree with only 1 max depth is shown. | Warning is displayed underneath the slider indicating that low max depth levels can cause underfitting. The decision tree still updates and works to show the results as expected. |
| Max-depth: high max-depth value of 10 + | Overfitting warning is shown, full tree is shown with a warning explaining high rendering times | Warning is displayed underneath the slider indicating that high max depth levels can cause overfitting. The decision tree still updates and works to show the results as expected. |

1. Criterion Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘squared\_error’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘friedman\_mse’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘absolute\_error’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘poisson’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |

1. Split Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Best’ | Tree and score updates using the relevant dropdown choice. ‘Best’ would generally have a higher test/train score. | The decision tree updates and uses the ‘best split’ criteria for the split from the dropdown to fit the decision tree model |
| Choose ‘Random’ | Tree and score updates using the relevant dropdown choice. | The decision tree updates and uses the ‘random split’ criteria for the split from the dropdown to fit the decision tree model |

1. Min Samples Split

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | When choosing 15 as the minimum split, the tree updates the decision tree with the new min samples per split value. In many cases this number does not make a difference in the scoring, but no error arises. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per split can lead to overfitting. Decision tree is still updated to reflect min sample split value. |

1. Min Samples Leaf

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid-range’ number on the slider | The tree visualisation and scores align with the selected number | Choosing 20 as the min samples per split. The decision tree and scores update to reflect new value. In many cases this number does not make a difference as much as max-depth value updates do. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per leaf can lead to overfitting. Decision tree is still updated to reflect min sample leaf value. |

1. Test Size

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Mid range number is chosen as 0.25. The scores and decision tree visualisation update to reflect the new chosen test split size as expected. |
| Choose small test size sample (0.05) | A warning is displayed, explaining why a small test size can lead to bad generalisation. The tree and scores are still updated to reflect these changes. | Small test split sizes display a warning underneath the slider. This message warns users that choosing small sample sizes may lead to weak generalisation of the model. The decision tree updates to take on the new test size value. |
| Choose large test size sample (0.5) | A warning is displayed, explaining why large training sets can lead to a bad model. The tree and scores are still updated to reflect these changes. | 0.5 test split size. A warning is displayed underneath the slider to indicate to the user that large test sizes will not provide enough training data to make good generalisations on the model. The visualisations and scores still update to reflect these changes. |

Model Training and Evaluation

Is the overfitting warning actually valid?

1. Decision Tree

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose different Max depth levels | Number of levels in the tree should match the max depth levels in the slider | Each change for the number in the max depth slider is successfully rendered in the decision tree visualisation. |
| Choose different min\_samples per leaf and split values | The decision tree should update and change to reflect any changes | Each change in both the sliders updates the decision tree visualisations. However the scores or tree do not change – these parameters may not change anything due to the size or simplicity of the dataset. |

1. Scores

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration criteria | Scores for training, test and generalisation should change and update | Each change in the parameter configuration causes an update for the training, test and generalisation scores. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). Check decision tree and visualisation to check this is accurate – is it really overfitting? | High training score, lower test score. Warning is displayed stating that the scores suggest overfitting | The training mae score is significantly lower than the test score. A warning is correctly displayed stating that the scores suggest overfitting. |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). Check decision tree and visualisation to check this is accurate – is it really underfitting? | Low training and low test scores. Warning is displayed stating that the scores suggest underfitting | A warning is displayed under the scores, showing that the scores suggest underfitting. Both training and test errors are high (0.31, 0.35), with both surpassing the threshold. The low max depth shown in the decision tree supports this. |
| Configure good model (check parameters through grid search CV). Check decision tree and visualisations to check this is accurate. | High training and high test score. Displays message suggesting that the scores show the model is a good fit. | Low training error: 0.235, low test error: 0.251. Both are under the over/underfitting threshold. A message is displayed underneath the scores suggesting this is a good model fit. |

1. True vs Predicted Values

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). | Plot displays correctly and updates dynamically to correlate with test errors. The correlation is weak. | The plots update to match new configurations. Underfitting – both plots don’t have fewer points close to the diagonal line as expected. |
| Configure good model (grid search CV) | Plot displays correctly and updates dynamically to correlate with test errors. The correlation is good. | The plot updates – both plots are very similar and have a good number of points close to the line. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). | Plot displays correctly and updates dynamically to correlate with test errors. The correlation for test is not as good as expected. | Overfitting model. The plot updates to show that the training plot has a much better result than the test plots. This is as expected. |

1. Cross Validation

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose all K values on the slider and test with balanced\_accuracy scoring | Mean, standard deviations and lists are dynamically updated to represent the number of K-Folds. | Cross validation with all k values on the slider (2-10) work, and update scorings and lists as expected. The mean and standard deviation re-renders and updates. The list length of cross validation scorings matches the k-fold value. |
| Check each scoring type for validity | Mean, standard deviations and lists are dynamically updated to represent the correct scoring type. | All scoring types for regression work as expected, each update in the dropdown dynamically updates the scores, mean, standard deviations and cross validation list. |

1. SHAP Analysis

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration of model | Any new configuration added to the model will update the SHAP summary and waterfall plot. | Each update changes the SHAP summary plot and matches the feature importance plot. All labels are correct and all features are shown. |

1. Partial Dependency Plot

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration of model | Any new configuration added to the model will update the partial dependency plot. | Each update changes the partial dependency plots. Plots are shown for all features in the dataset. |

1. Cost Complexity Pruning

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure good model (grid search CV). Check CCP Alphas and graphs | If a better solution is found, the training and test scores will be higher for this decision tree. Decision tree visualisation is different from the configurated one (if improved scores). | The cpp alphas improve, and the training and test scores improve. The model has improved in this case, and shows that the model can still be tuned for further improvement. End decision tree is more complex. |

Feature Selection and Importance

1. Feature Importance

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a decision tree model – repeat to see updates | Feature Importance plot should show in ranking value which feature is most used. Values in the list should correspond with graph. These should update with changes to the decision tree. | Change the test split size, and configuration parameters updates the feature importance list and plot. Both the graph and list and correctly ranked, with the right features displaying in both. |

1. Feature Selection

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check each value for slider to pick top-most important | The features chosen should match those in order of importance on the feature importance graph. Retrain classifier with all slider values should work | The feature selection process updates to match the feature importance plot. Adding more features until feature limit is reached works as expected. Decision tree displays correctly. |
| Feature selection plot | Should show correct number of numerical features available. | Shows the correct number and feature labels in the plot |

Analysis

1. GridSearchCV

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Insert letters, negative numbers, float numbers to the Max depth and Min samples per leaf inputs. | This should return an error message to the user indicating only valid natural numbers are allowed. | Any invalid datatypes such as strings ‘a’, float values ‘2.64’ and negative numbers ‘-1’ are not added to the list as expected. A message arises stating these aren’t valid and are excluded from the resultant lists. |
| Values for the inputs section in the list | These values should match those added into the inputs by the user. If many are added, the user must be warned about training times. | If the datatypes are valid natural numbers, the values in the list are shown as expected. |
| Scores and Visualisations | After a valid grid search a visualisation of the best model tree, the best model parameters and scores are displayed correctly. | After clicking the ‘perform grid search’ button, the correct parameters, scores and decision tree visualisation of the best model is shown as expected. |

Save

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a changed model with complete inputs. | The save successfully adds the correct values to an ini file on the desktop. | Correct values are added to the new saved .ini file in the ‘saves’ directory. |
| Save a model with no changes (file has been added) | The default model is saved successfully to an ini file on the desktop | Opening the iris dataset, all the default values are created and is saved as a .ini file with all the correct info. |

Upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Click on upload button, directs the user to .ini files. The user can upload a previously saved file. | If a user has saved a .ini file, once the file has been uploaded all details are the same as when saved as expected. The user can also make any changes and save again to update an changes in the model. | The user can upload and browse previously saved files. All info is shown as correct, ‘perform grid search’ and ‘retrain with selected features’ are not fully shown due to the session states. Information saved prior are retained to create the same results. |
| Replace dataframe with another .csv file (after a config file has been loaded) | Refreshes to display new dataframe, previous dataframe info from uploaded config is not saved. | Page does not refresh. States that a new dataframe has been uploaded, but the uploaded config file data has overridden CSV file upload. |

**housing-cleaned.csv**

Data Exploration

1. Choose target column

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Pick a target column that is numerical | User only has the option to pick from numerical columns for the target. | All numerical features are shown in the target dropdown, as expected. |

1. Boxplot

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check feature labels | All feature labels are labelled correctly. | All features are labelled correctly on the plot. Outliers remaining in the dataset can be seen unscaled. |

1. Correlation Matrix

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check feature labels and plot | All feature labels are labelled correctly, and plot is displayed clearly. | All features are labelled correctly on the plot, and the plot can be enlarged to be seen clearly. |

Model Configuration and Hyper-parameter Tuning

1. Max Depth

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Max-depth: Choose a max-depth value from the slider and observe a change in the resultant decision tree image | Depth of the decision tree visualisation dynamically corresponds to the number on the max-depth slider | Decision tree visualisation changes along with new updated values in the max depth slider. Scalability might be an issue; load times are very long with max depths > 6. Probably due to the size of the dataset. |
| Max-depth: max-depth value of 1 on slider | Underfitting warning shown, a tree with only 1 max depth is shown. | Warning is displayed underneath the slider indicating that low max depth levels can cause underfitting. The decision tree still updates and works to show the results as expected. |
| Max-depth: high max-depth value of 10 + | Overfitting warning is shown, full tree is shown with a warning explaining high rendering times | Warning is displayed underneath the slider indicating that high max depth levels can cause overfitting. The decision tree still updates and works to show the results as expected, but load times are very long. |

1. Criterion Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘squared\_error’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘friedman\_mse’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘absolute\_error’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘poisson’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |

1. Split Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Best’ | Tree and score updates using the relevant dropdown choice. ‘Best’ would generally have a higher test/train score. | The decision tree updates and uses the ‘best split’ criteria for the split from the dropdown to fit the decision tree model |
| Choose ‘Random’ | Tree and score updates using the relevant dropdown choice. | The decision tree updates and uses the ‘random split’ criteria for the split from the dropdown to fit the decision tree model |

1. Min Samples Split

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | When choosing 15 as the minimum split, the tree updates the decision tree with the new min samples per split value. In many cases this number does not make a difference in the scoring, but no error arises. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per split can lead to overfitting. Decision tree is still updated to reflect min sample split value. |

1. Min Samples Leaf

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid-range’ number on the slider | The tree visualisation and scores align with the selected number | Choosing 20 as the min samples per split. The decision tree and scores update to reflect new value. In many cases this number does not make a difference as much as max-depth value updates do. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per leaf can lead to overfitting. Decision tree is still updated to reflect min sample leaf value. |

1. Test Size

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Mid range number is chosen as 0.25. The scores and decision tree visualisation update to reflect the new chosen test split size as expected. |
| Choose small test size sample (0.05) | A warning is displayed, explaining why a small test size can lead to bad generalisation. The tree and scores are still updated to reflect these changes. | Small test split sizes display a warning underneath the slider. This message warns users that choosing small sample sizes may lead to weak generalisation of the model. The decision tree updates to take on the new test size value. |
| Choose large test size sample (0.5) | A warning is displayed, explaining why large training sets can lead to a bad model. The tree and scores are still updated to reflect these changes. | 0.5 test split size. A warning is displayed underneath the slider to indicate to the user that large test sizes will not provide enough training data to make good generalisations on the model. The visualisations and scores still update to reflect these changes. |

Model Training and Evaluation

Is the overfitting warning actually valid?

1. Decision Tree

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose different Max depth levels | Number of levels in the tree should match the max depth levels in the slider | Each change for the number in the max depth slider is successfully rendered in the decision tree visualisation. Page loading times increases when max depth increases – application becomes unstable with larger datasets. Issues with scalability. |
| Choose different min\_samples per leaf and split values | The decision tree should update and change to reflect any changes | Each change in both the sliders updates the decision tree visualisations. However the scores or tree do not change – these parameters may not change anything due to the size or simplicity of the dataset. |

1. Scores

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration criteria | Scores for training, test and generalisation should change and update | Each change in the parameter configuration causes an update for the training, test and generalisation scores. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). Check decision tree and visualisation to check this is accurate – is it really overfitting? | High training score, lower test score. Warning is displayed stating that the scores suggest overfitting | The training mae score is significantly lower than the test score. A warning is correctly displayed stating that the scores suggest overfitting.  In one case the difference between the scores are not very different, but the test score is greater than the baseline threshold (mean + 1.5\*std). This could be a point of disagreement on whether this is overfitting or not |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). Check decision tree and visualisation to check this is accurate – is it really underfitting? | Low training and low test scores. Warning is displayed stating that the scores suggest underfitting | With this dataset it is difficult to obtain an ‘underfitting’ model. could also mean that the method to establish any over/underfitting data needs to be different for this dataset. One underfitting model was found, as both scores were close to 0 but surpassed the threshold for underfit. An appropriate underfit warning was displayed under the model scores. |
| Configure good model (check parameters through grid search CV). Check decision tree and visualisations to check this is accurate. | High training and high test score. Displays message suggesting that the scores show the model is a good fit. | Very low mae errors for both training and test scores. They are also very close scores suggesting they have generalised well. A message is displayed underneath the scores suggesting this is a good model fit. |

1. True vs Predicted Values

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). | Plot displays correctly and updates dynamically to correlate with test errors. The correlation is weak. | The plots update to match new configurations. Underfitting – both plots don’t have fewer points close to the diagonal line as expected. |
| Configure good model (grid search CV) | Plot displays correctly and updates dynamically to correlate with test errors. The correlation is good. | The plot updates – both plots are very similar and have a good number of points close to the line. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). | Plot displays correctly and updates dynamically to correlate with test errors. The correlation for test is not as good as expected. | Overfitting model. The plot updates to show that the training plot has a much better result than the test plots. This is as expected. |

1. Cross Validation

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose all K values on the slider and test with balanced\_accuracy scoring | Mean, standard deviations and lists are dynamically updated to represent the number of K-Folds. | Cross validation with all k values on the slider (2-10) work, and update scorings and lists as expected. The mean and standard deviation re-renders and updates. The list length of cross validation scorings matches the k-fold value. |
| Check each scoring type for validity | Mean, standard deviations and lists are dynamically updated to represent the correct scoring type. | All scoring types for regression work as expected, each update in the dropdown dynamically updates the scores, mean, standard deviations and cross validation list. |

1. SHAP Analysis

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration of model | Any new configuration added to the model will update the SHAP summary and waterfall plot. | Each update changes the SHAP summary plot and matches the feature importance plot. All labels are correct and all features are shown. |

1. Partial Dependency Plot

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration of model | Any new configuration added to the model will update the partial dependency plot. | Each update changes the partial dependency plots. Plots are shown for all features in the dataset. |

1. Cost Complexity Pruning

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure good model (grid search CV). Check CCP Alphas and graphs | If a better solution is found, the training and test scores will be higher for this decision tree. Decision tree visualisation is different from the configurated one (if improved scores). | The cpp alphas improve, and the training and test scores improve. End ones do not show and take too long. No scores provided as this would probably return a very large, complex decision tree. Pruned tree visualisation for this dataset takes a very long time to load!! |

Feature Selection and Importance

1. Feature Importance

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a decision tree model – repeat to see updates | Feature Importance plot should show in ranking value which feature is most used. Values in the list should correspond with graph. These should update with changes to the decision tree. | Change the test split size, and configuration parameters updates the feature importance list and plot. Both the graph and list and correctly ranked, with the right features displaying in both. |

1. Feature Selection

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check each value for slider to pick top-most important | The features chosen should match those in order of importance on the feature importance graph. Retrain classifier with all slider values should work | The feature selection process updates to match the feature importance plot. Adding more features until feature limit is reached works as expected. |
| Feature selection plot | Should show correct number of numerical features available. | Shows the correct number and feature labels in the plot |

Analysis

1. GridSearchCV

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Insert letters, negative numbers, float numbers to the Max depth and Min samples per leaf inputs. | This should return an error message to the user indicating only valid natural numbers are allowed. | Columns load in order. GridSearchCV section does not load, as pruned tree is still running. Big scalability problem. |
| Values for the inputs section in the list | These values should match those added into the inputs by the user. If many are added, the user must be warned about training times. | - |
| Scores and Visualisations | After a valid grid search a visualisation of the best model tree, the best model parameters and scores are displayed correctly. | - |

Save

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a changed model with complete inputs. | The save successfully adds the correct values to an ini file on the desktop. | Correct values are added to the new saved .ini file in the ‘saves’ directory. |
| Save a model with no changes (file has been added) | The default model is saved successfully to an ini file on the desktop | Opening the iris dataset, all the default values are created and is saved as a .ini file with all the correct info. |

Upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Click on upload button, directs the user to .ini files. The user can upload a previously saved file. | If a user has saved a .ini file, once the file has been uploaded all details are the same as when saved as expected. The user can also make any changes and save again to update an changes in the model. | The user can upload and browse previously saved files. All info is shown as correct, ‘perform grid search’ and ‘retrain with selected features’ are not fully shown due to the session states. Information saved prior are retained to create the same results. |
| Replace dataframe with another .csv file (after a config file has been loaded) | Refreshes to display new dataframe, previous dataframe info from uploaded config is not saved. | Page does not refresh. States that a new dataframe has been uploaded, but the uploaded config file data has overridden CSV file upload. |

Potential fix for scalability: only establish running on certain sections until needed.

Allow user to work on a subset of the data

**Winequality-red.csv**

Data Exploration

1. Choose target column

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Pick a target column that is numerical | User only has the option to pick from numerical columns for the target. | All numerical features are shown in the target dropdown, as expected. |

1. Boxplot

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check feature labels | All feature labels are labelled correctly. | All features are labelled correctly on the plot. Outliers remaining in the dataset can be seen before data has been scaled |

1. Correlation Matrix

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check feature labels and plot | All feature labels are labelled correctly, and plot is displayed clearly. | All features are labelled correctly on the plot, and the plot can be enlarged to be seen clearly. |

Model Configuration and Hyper-parameter Tuning

1. Max Depth

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Max-depth: Choose a max-depth value from the slider and observe a change in the resultant decision tree image | Depth of the decision tree visualisation dynamically corresponds to the number on the max-depth slider | Decision tree visualisation changes along with new updated values in the max depth slider. All values change the decision tree depth as expected without any errors. |
| Max-depth: max-depth value of 1 on slider | Underfitting warning shown, a tree with only 1 max depth is shown. | Warning is displayed underneath the slider indicating that low max depth levels can cause underfitting. The decision tree still updates and works to show the results as expected. |
| Max-depth: high max-depth value of 10 + | Overfitting warning is shown, full tree is shown with a warning explaining high rendering times | Warning is displayed underneath the slider indicating that high max depth levels can cause overfitting. The decision tree still updates and works to show the results as expected, but load times are very long. |

1. Criterion Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘squared\_error’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘friedman\_mse’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘absolute\_error’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |
| Choose ‘poisson’ | Tree and score updates using the relevant dropdown choice | The decision tree updates and uses the saved scoring value from the dropdown to fit the decision tree model |

1. Split Dropdown

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose ‘Best’ | Tree and score updates using the relevant dropdown choice. ‘Best’ would generally have a higher test/train score. | The decision tree updates and uses the ‘best split’ criteria for the split from the dropdown to fit the decision tree model |
| Choose ‘Random’ | Tree and score updates using the relevant dropdown choice. | The decision tree updates and uses the ‘random split’ criteria for the split from the dropdown to fit the decision tree model |

1. Min Samples Split

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | When choosing 15 as the minimum split, the tree updates the decision tree with the new min samples per split value. In many cases this number does not make a difference in the scoring, but no error arises. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per split can lead to overfitting. Decision tree is still updated to reflect min sample split value. |

1. Min Samples Leaf

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid-range’ number on the slider | The tree visualisation and scores align with the selected number | Choosing 20 as the min samples per split. The decision tree and scores update to reflect new value. In many cases this number does not make a difference as much as max-depth value updates do. |
| Choose 1 on the slider | An overfitting warning is displayed, but the tree visualisation and scores are still updated | Warning is displayed indicating to the user that low samples per leaf can lead to overfitting. Decision tree is still updated to reflect min sample leaf value. |

1. Test Size

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Enter a valid ‘mid range’ number on the slider | The tree visualisation and scores align with the selected number | Mid range number is chosen as 0.25. The scores and decision tree visualisation update to reflect the new chosen test split size as expected. |
| Choose small test size sample (0.05) | A warning is displayed, explaining why a small test size can lead to bad generalisation. The tree and scores are still updated to reflect these changes. | Small test split sizes display a warning underneath the slider. This message warns users that choosing small sample sizes may lead to weak generalisation of the model. The decision tree updates to take on the new test size value. |
| Choose large test size sample (0.5) | A warning is displayed, explaining why large training sets can lead to a bad model. The tree and scores are still updated to reflect these changes. | 0.5 test split size. A warning is displayed underneath the slider to indicate to the user that large test sizes will not provide enough training data to make good generalisations on the model. The visualisations and scores still update to reflect these changes. |

Model Training and Evaluation

Is the overfitting warning actually valid?

1. Decision Tree

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose different Max depth levels | Number of levels in the tree should match the max depth levels in the slider | Each change for the number in the max depth slider is successfully rendered in the decision tree visualisation. Page loading times increases when max depth increases – application becomes unstable with larger datasets. Issues with scalability. Max depths 9-10 do not work, receive a dtreeviz error. |
| Choose different min\_samples per leaf and split values | The decision tree should update and change to reflect any changes | Each change in both the sliders updates the decision tree visualisations. However the scores or tree do not change – these parameters may not change anything due to the size or simplicity of the dataset. |

1. Scores

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration criteria | Scores for training, test and generalisation should change and update | Each change in the parameter configuration causes an update for the training, test and threshold scores. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). Check decision tree and visualisation to check this is accurate – is it really overfitting? | High training score, lower test score. Warning is displayed stating that the scores suggest overfitting | The training mae score is significantly lower than the test score. (Lower than training and above the threshold). A warning is correctly displayed stating that the scores suggest overfitting. |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). Check decision tree and visualisation to check this is accurate – is it really underfitting? | Low training and low test scores. Warning is displayed stating that the scores suggest underfitting | Purposely underfitting the data (df has been sampled to 50 points). Score message display doesn’t equate with the true vs. predicted plot. Plot looks as if it is underfitting, instead the message displays saying it could be further improved.  With this dataset it is difficult to obtain an ‘underfitting’ model. could also mean that the method to establish any over/underfitting data needs to be different for this dataset. |
| Configure good model (check parameters through grid search CV). Check decision tree and visualisations to check this is accurate. | High training and high test score. Displays message suggesting that the scores show the model is a good fit. | Very low mae errors for both training and test scores. They are also very close scores suggesting they have generalised well. A message is displayed underneath the scores suggesting this is a good model fit. |

1. True vs Predicted Values

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure underfitting model (low max depth, high min samples split, high min samples leaf). | Plot displays correctly and updates dynamically to correlate with test errors. The correlation is weak. | The plots update to match new configurations. Underfitting – both plots don’t have fewer points close to the diagonal line as expected. |
| Configure good model (grid search CV) | Plot displays correctly and updates dynamically to correlate with test errors. The correlation is good. | The plot updates – both plots are very similar and have a good number of points close to the line. |
| Configure overfitting model (High max depth, low min samples split, low min samples leaf). | Plot displays correctly and updates dynamically to correlate with test errors. The correlation for test is not as good as expected. | Overfitting model. The plot updates to show that the training plot has a much better result than the test plots. This is as expected. |

1. Cross Validation

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Choose all K values on the slider and test with balanced\_accuracy scoring | Mean, standard deviations and lists are dynamically updated to represent the number of K-Folds. | Cross validation with all k values on the slider (2-10) work, and update scorings and lists as expected. The mean and standard deviation re-renders and updates. The list length of cross validation scorings matches the k-fold value. |
| Check each scoring type for validity | Mean, standard deviations and lists are dynamically updated to represent the correct scoring type. | All scoring types for regression work as expected, each update in the dropdown dynamically updates the scores, mean, standard deviations and cross validation list. |

1. SHAP Analysis

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration of model | Any new configuration added to the model will update the SHAP summary and waterfall plot. | Each update changes the SHAP summary plot and matches the feature importance plot. All labels are correct and all features are shown. |

1. Partial Dependency Plot

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Change configuration of model | Any new configuration added to the model will update the partial dependency plot. | Each update changes the partial dependency plots. Plots are shown for all features in the dataset. |

1. Cost Complexity Pruning

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure good model (grid search CV). Check CCP Alphas and graphs | If a better solution is found, the training and test scores will be higher for this decision tree. Decision tree visualisation is different from the configurated one (if improved scores). | The cpp alphas improve, and the training and test scores improve. End ones do not show and take too long. No scores provided as this would probably return a very large, complex decision tree. This has been capped at max\_depth now works. |

Feature Selection and Importance

1. Feature Importance

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a decision tree model – repeat to see updates | Feature Importance plot should show in ranking value which feature is most used. Values in the list should correspond with graph. These should update with changes to the decision tree. | Change the test split size, and configuration parameters updates the feature importance list and plot. Both the graph and list and correctly ranked, with the right features displaying in both. |

1. Feature Selection

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Check each value for slider to pick top-most important | The features chosen should match those in order of importance on the feature importance graph. Retrain classifier with all slider values should work | The feature selection process updates to match the feature importance plot. Adding more features until feature limit is reached works as expected. |
| Feature selection plot | Should show correct number of numerical features available. | Shows the correct number and feature labels in the plot |

Analysis

1. GridSearchCV

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Insert letters, negative numbers, float numbers to the Max depth and Min samples per leaf inputs. | This should return an error message to the user indicating only valid natural numbers are allowed. | Any invalid datatypes such as strings ‘a’, float values ‘2.64’ and negative numbers ‘-1’ are not added to the list as expected. A message arises stating these aren’t valid and are excluded from the resultant lists. |
| Values for the inputs section in the list | These values should match those added into the inputs by the user. If many are added, the user must be warned about training times. | If the datatypes are valid natural numbers, the values in the list are shown as expected. |
| Scores and Visualisations | After a valid grid search a visualisation of the best model tree, the best model parameters and scores are displayed correctly. | After clicking the ‘perform grid search’ button, the correct parameters, scores and decision tree visualisation of the best model is shown as expected. |

Save

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Configure a changed model with complete inputs. | The save successfully adds the correct values to an ini file on the desktop. | Correct values are added to the new saved .ini file in the ‘saves’ directory. |
| Save a model with no changes (file has been added) | The default model is saved successfully to an ini file on the desktop | Opening the iris dataset, all the default values are created and is saved as a .ini file with all the correct info. |

Upload

|  |  |  |
| --- | --- | --- |
| Action | Expected Outcome | Outcome |
| Click on upload button, directs the user to .ini files. The user can upload a previously saved file. | If a user has saved a .ini file, once the file has been uploaded all details are the same as when saved as expected. The user can also make any changes and save again to update an changes in the model. | The user can upload and browse previously saved files. All info is shown as correct, ‘perform grid search’ and ‘retrain with selected features’ are not fully shown due to the session states. Information saved prior are retained to create the same results. |
| Replace dataframe with another .csv file (after a config file has been loaded) | Refreshes to display new dataframe, previous dataframe info from uploaded config is not saved. | Page does not refresh. States that a new dataframe has been uploaded, but the uploaded config file data has overridden CSV file upload. |

**Performance testing**

**Classification**

|  |  |  |  |
| --- | --- | --- | --- |
| **Classification Model – Performance Testing** | | | |
| **Dataset name** | **Load\_times (s): Max\_depth = 1, Min\_samples\_split = 2, Min\_samples\_leaf = 1** | **Load\_times(s):**  **Max\_depth=4**  **Min\_samples\_split = 2,**  **Min\_samples\_leaf = 1** | **Load\_times(s):**  **Max\_depth = 7**  **Min\_samples\_split = 2,**  **Min\_samples\_leaf = 1** |
| **Iris.csv** | 20.759237051010132 | 58.206010818481445 | 62.7831158638000 |
| **Wine.csv** | 5.467482805252075 | 18.824524641036987 | **---** |
| **Breast-cancer.csv** | 5.28674578666687 | 12.457592010498047 | 14.989876985549927 |

**Regression**

|  |  |  |  |
| --- | --- | --- | --- |
| **Regression Model – Performance Testing** | | | |
| **Dataset name** | **Load\_times: Max\_depth = 1, Min\_samples\_split = 2, Min\_samples\_leaf = 1** | **Load\_times:**  **Max\_depth=4**  **Min\_samples\_split = 2,**  **Min\_samples\_leaf = 1** | **Load\_times:**  **Max\_depth = 7**  **Min\_samples\_split = 2,**  **Min\_samples\_leaf = 1** |
| **Diabetes.csv** | 28.91278600692749 | 89.90188717842102 | 286.8029680252075 |
| **Winequality-red.csv** | 37.40577006340027 | 113.86705088615417 | 375.6883478164673 |
| **Housing-cleaned.csv** | 53.99760103225708 | 147.77776980400085 | 707.8037519454956 |