Titanic AI project

Session 4 FINAL

Team:

Yuliia Nikolaenko
Eduardo Bonnefemne
Gabriel Pérez García
Elena-Mihaela Grigore

Part 1: Decision Trees

Quick Recap

Survived	Pclass	SibSp	Parch	FareType	SexCode	Age_cat	Embarked_code
0	549	549	549	549	549	549	549
1	342	342	342	342	342	342	342

Ticked and Name were dropped

Age

- imputed as mean of the respective pclass
- binned as 5 bands of equal width in the min_age -> max_age space

Fare - binned as the 4 quartiles of the Fare frequencies

1.1 Decision Trees - default params

- We fit a Decision Tree with default parameters to observe accuracy:
  - Train
  - Test
- Train test split (70-30)

Q: What to expect ?

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=None, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')
Training accuracy: 0.897
[[133  21]
 [ 42  72]]


              precision    recall  f1-score   support

           0       0.76      0.86      0.81       154
           1       0.77      0.63      0.70       114

    accuracy                           0.76       268
   macro avg       0.77      0.75      0.75       268
weighted avg       0.77      0.76      0.76       268
```

1.2 Test accuracy 5-fold cross-validation ?

- Decision Tree with default param
- 5-fold cross-validation
- Test accuracy only
- Q: low accuracy on test by chance only ?

1.2 Test accuracy 5-fold cross-validation ?

- Decision Tree with default param
- 5-fold cross-validation
- Test accuracy only
- Q: low accuracy on test by chance only ?

```
Accuracy for all folds: [0.77653631 0.76404494 0.78089888 0.80337079 0.78089888]
Mean accuracy: 0.78
Standard deviation: 0.01
```

1.2 Test accuracy 5-fold cross-validation ?

- Decision Tree with default param
- 5-fold cross-validation
- Test accuracy only
- Q: low accuracy on test by chance only ?

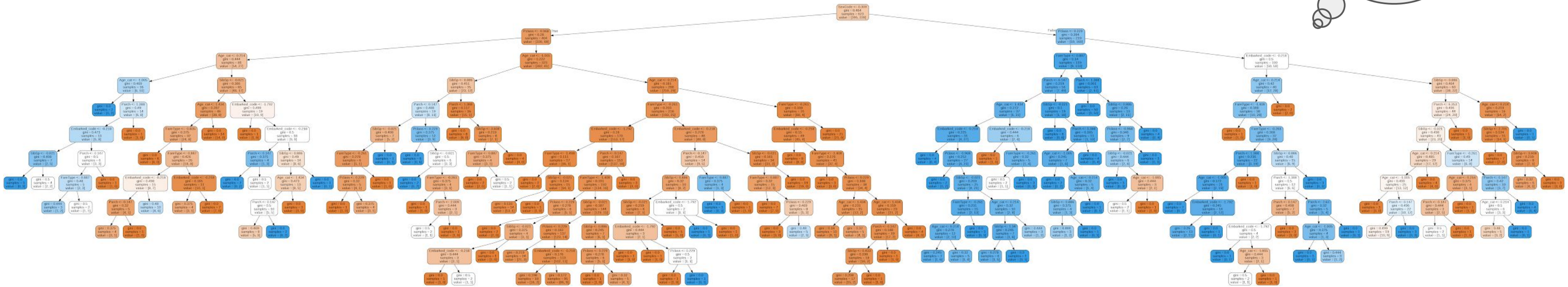
```
Accuracy for all folds: [0.77653631 0.76404494 0.78089888 0.80337079 0.78089888]
Mean accuracy: 0.78
Standard deviation: 0.01
```

- Remember from our first test ->
- 5-fold results are similar

```
Training accuracy: 0.897
Testin accuracy: 0.76
```

1.3 Decision Tree - visualisation

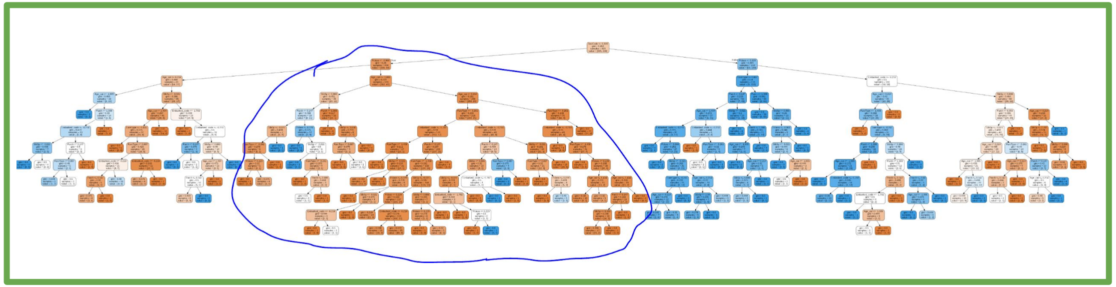
Great picture,
thank you !



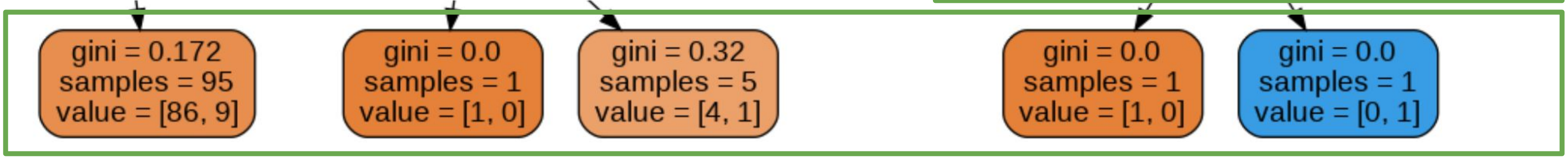
- Exploration (qualitative)
  - Have a look at the depth
  - Balanced ?
  - What do we find in the leaves
- Purpose: look for obvious opportunities to improve

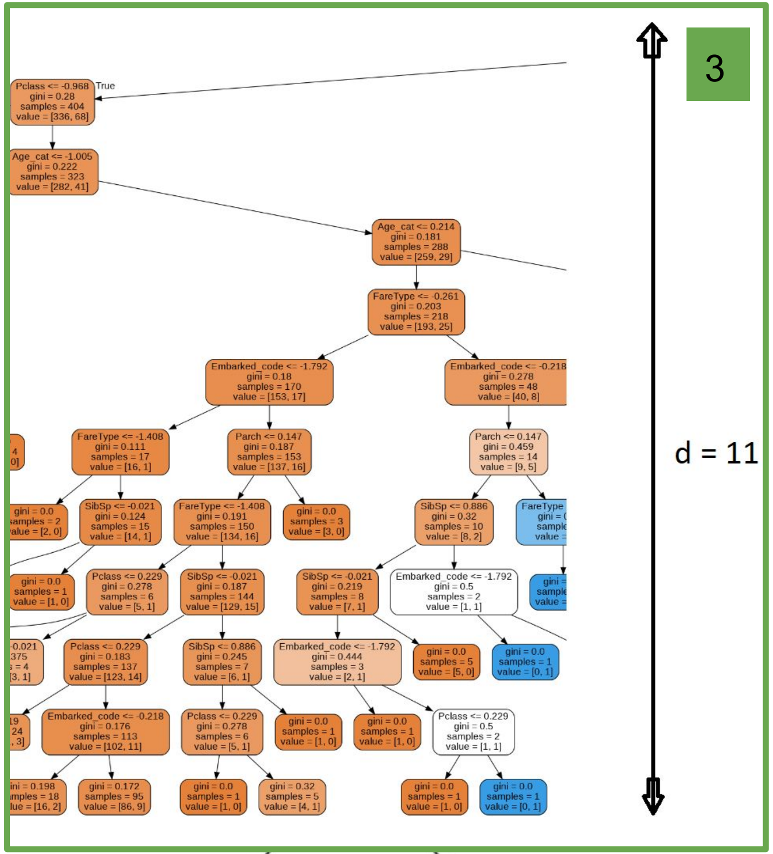
1.3 Decision Tree - visualisation

1



2





1.4 Overfitting

The disadvantages of decision trees include:

- Decision-tree learners can create over-complex trees that do not generalise the data well. This is called overfitting. Mechanisms such as pruning, setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree are necessary to avoid this problem.

https://scikit-learn.org/stable/modules/tree.html#tree

- Proceed with
  - Maximum depth
  - Minimum numbers of samples @ leaf level

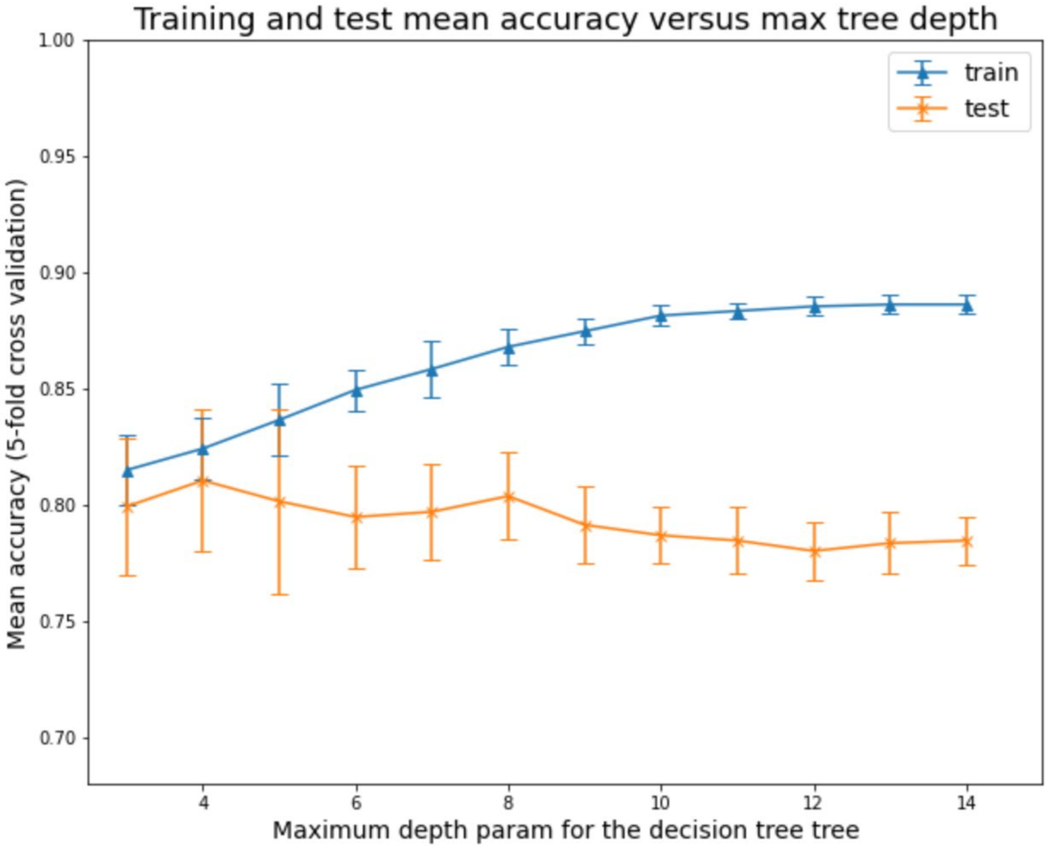
1.5 Decision Trees depth effect on train-test accuracy

Max depth: 3* -> 14

- 5-fold cross-validation
- Train decision tree
- Get accuracy for train and test
- Plot mean and standard error

Observations:

- Depth versus generalization



Recommended as start point in the official documentation: https://scikit-learn.org/stable/modules/tree.html#tree

1.5 Decision Trees depth effect on train-test accuracy

Max depth: 3* -> 14

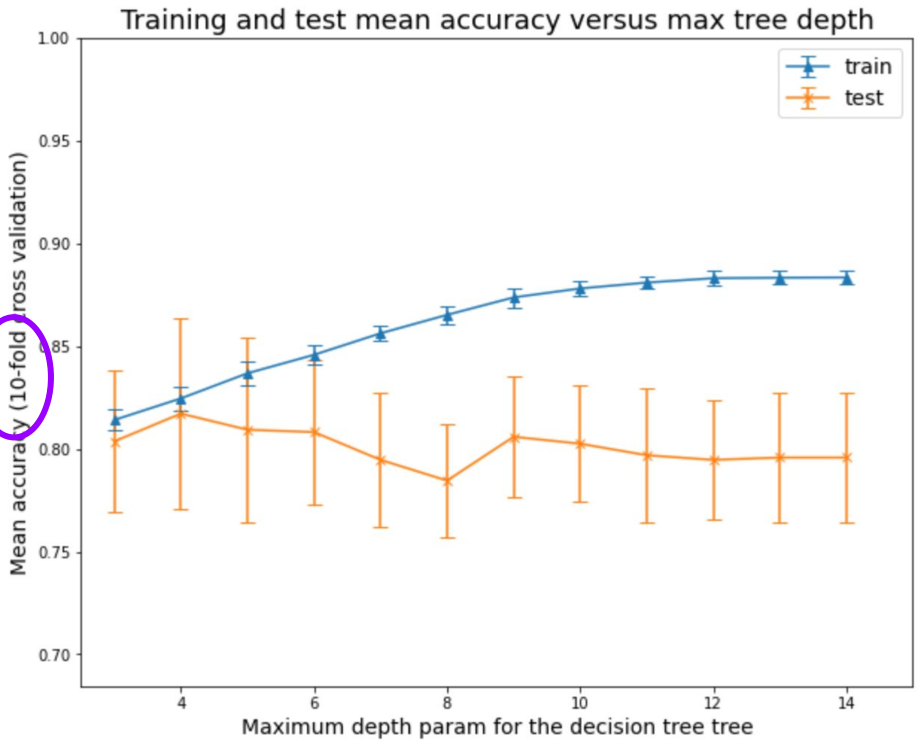
- 5-fold cross-validation
- Train decision tree
- Get accuracy for train and test
- Plot mean and standard error

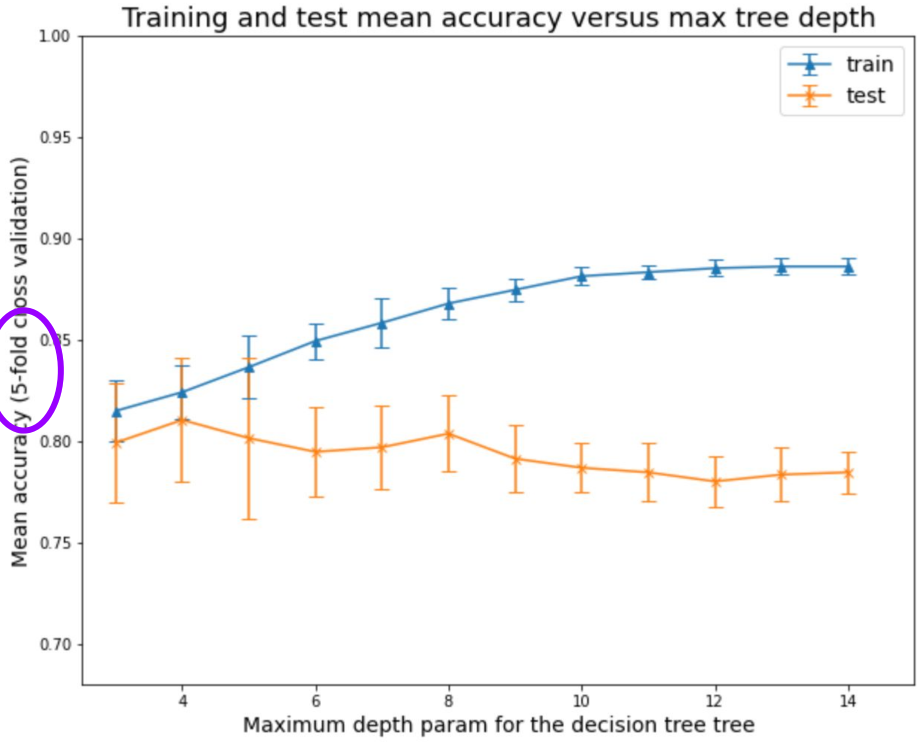
Observations:

- Depth versus generalization

Recommended as start point in the official documentation: https://scikit-learn.org/stable/modules/tree.html#tree

1.6 Decision Trees - cross-validation # of folds





1.7 Decision Trees - min samples

- Use
    - min_samples_split
    - min_samples_leaf
        - start: min_samples_leaf=5
        - for classification with few classes, min_samples_leaf=1 is often the best choice

to ensure that multiple samples inform every decision in the tree,

- very small number => overfit
- large number => ! learning the data.

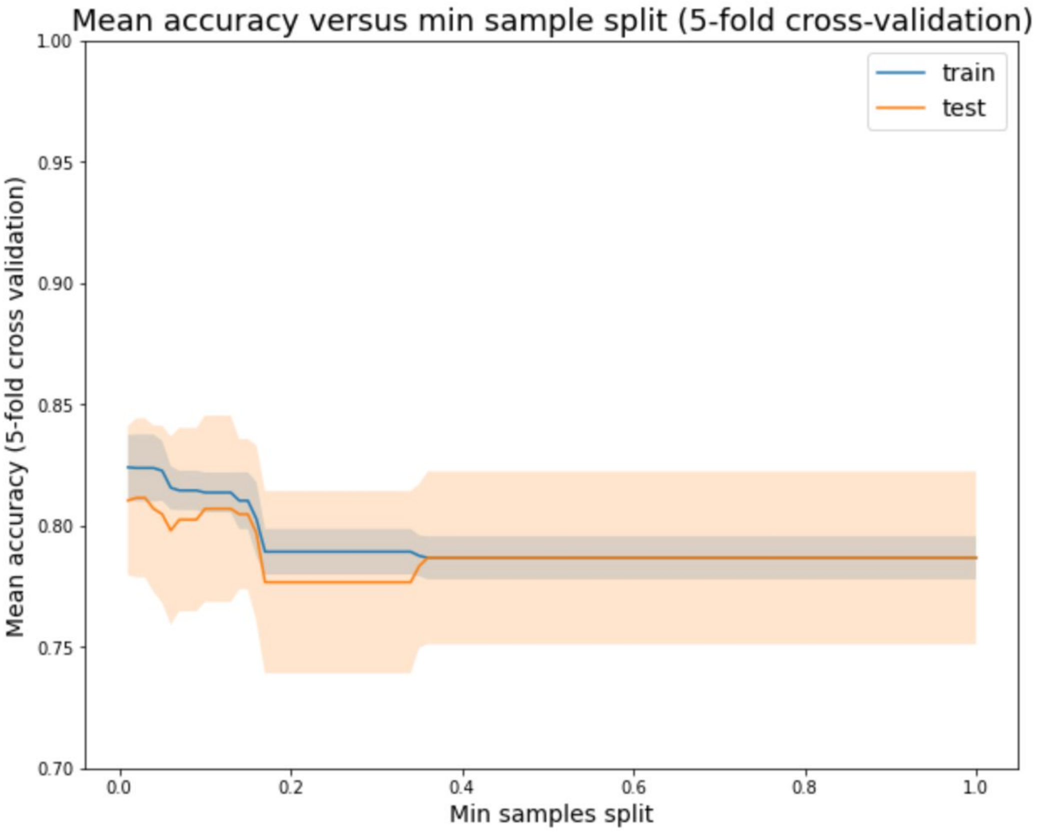
1.7.1 Decision Trees - min samples split

Conclusion:

better set to 1, as mentioned in the official documentation*

//default is 2

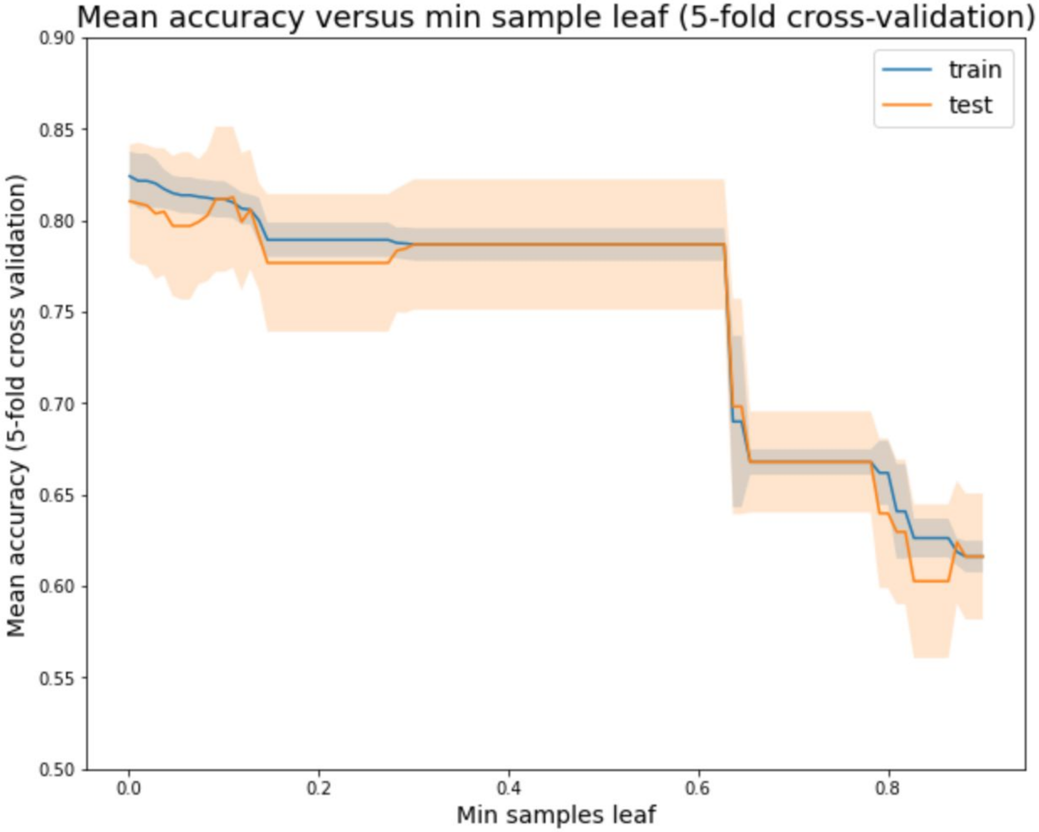
*https://scikit-learn.org/stable/modules/tree.html#tree



1.7.2 Decision Trees - min samples leaf

- For classification with few classes, min_samples_leaf=1 is often the best choice*

- Note that min sample leaf on X axis is shown as %, not absolute values



*https://scikit-learn.org/stable/modules/tree.html#tree

1.8 Post pruning with cost complexity

- Cost complexity pruning -> control the size of a tree (to prevent overfitting)
- Parameter: cost complexity parameter (ccp_alpha)
- Higher ccp_alpha => more nodes are pruned
- We choose the right ccp_alpha based on validation scores

$$R_\alpha(T_t) = R(T_t) + \alpha|T_t| \qquad R(T_t) = \sum_{t' \in L} R(t')$$

$$\boxed{\alpha_{eff}(t) = \frac{R(t) - R(T_t)}{|T| - 1}}$$

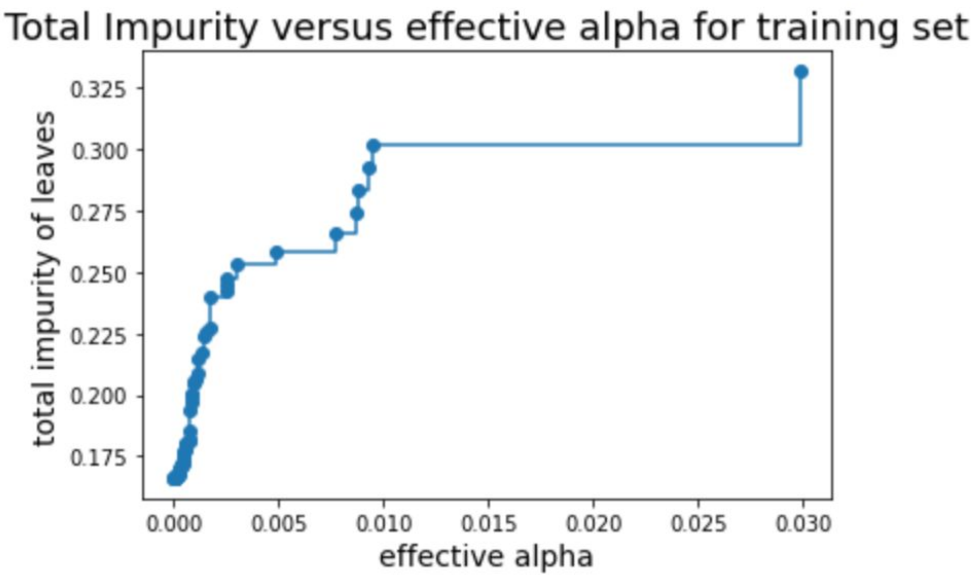
Sources:

https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py

https://scikit-learn.org/stable/modules/tree.html#minimal-cost-complexity-pruning

1.8 Post pruning with cost complexity - alpha values

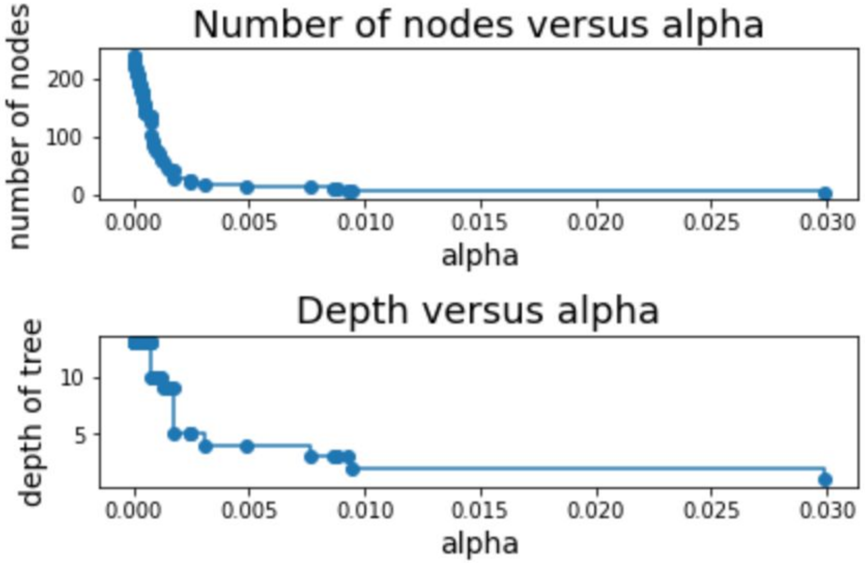
- Cost complexity pruning -> control the size of a tree (to prevent overfitting)
- Parameter: cost complexity parameter (ccp_alpha)
- Higher ccp_alpha => more nodes are pruned
- We choose the right ccp_alpha based on validation scores



source:https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py

1.8 Cost complexity pruning - further exploration

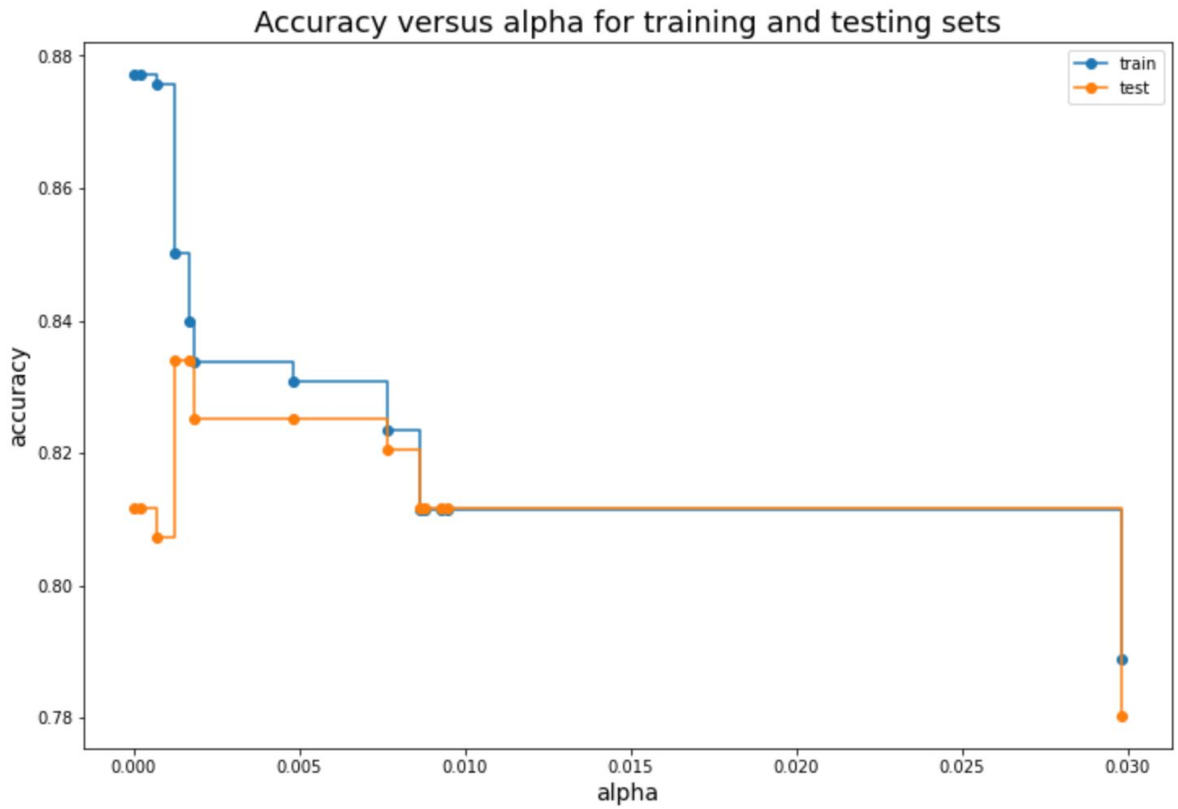
- The higher the alpha =>
  - the lower the # of nodes
  - The lower the depth of the tree



source:https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#sphx-glr-auto-examples-tree-plot-cost-complexity-pruning-py

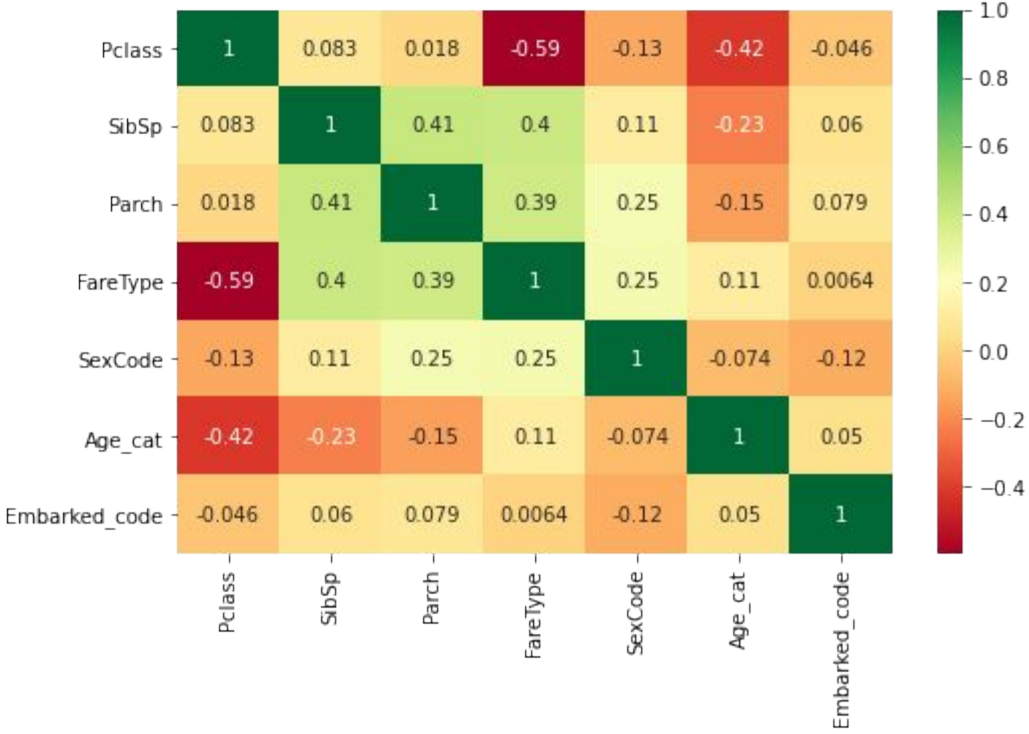
1.8 Cost complexity pruning - choosing the right val.

- Effect of choice of alpha on the accuracy for the train and test set
- Train-test split (0.25)
- Observation: should run cross-validation to check the noise in the results



Part 2: Naive Bayes

Naive Bayes: Pearson Correlation



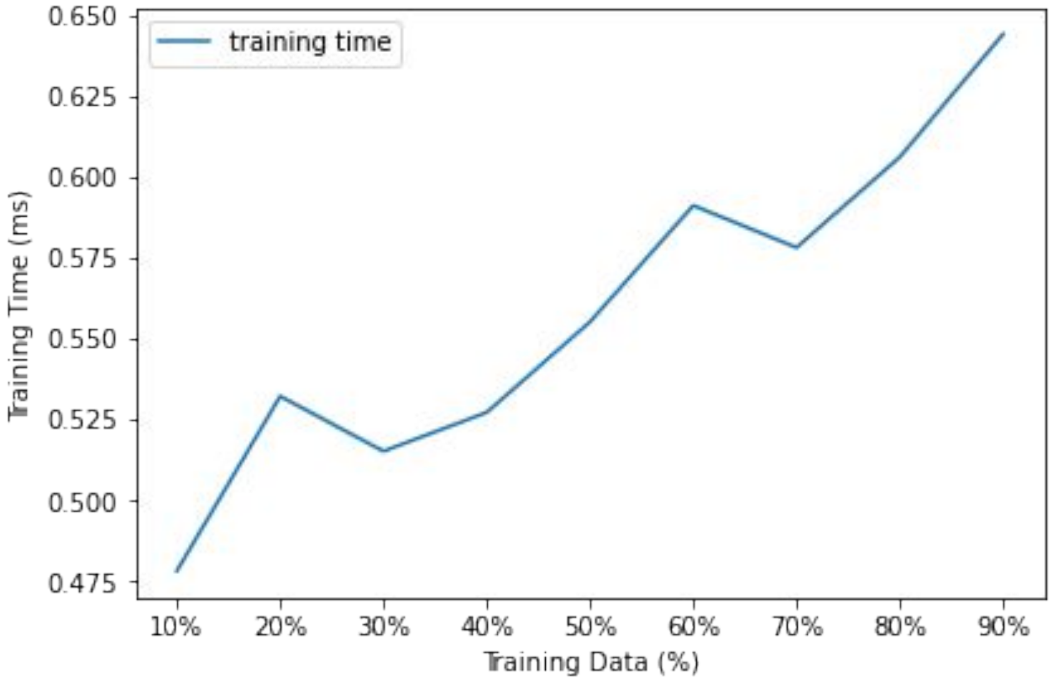
The variables “FareType” and “Pclass” are a bit correlated.

Naive Bayes: Score (after 100 experiments)

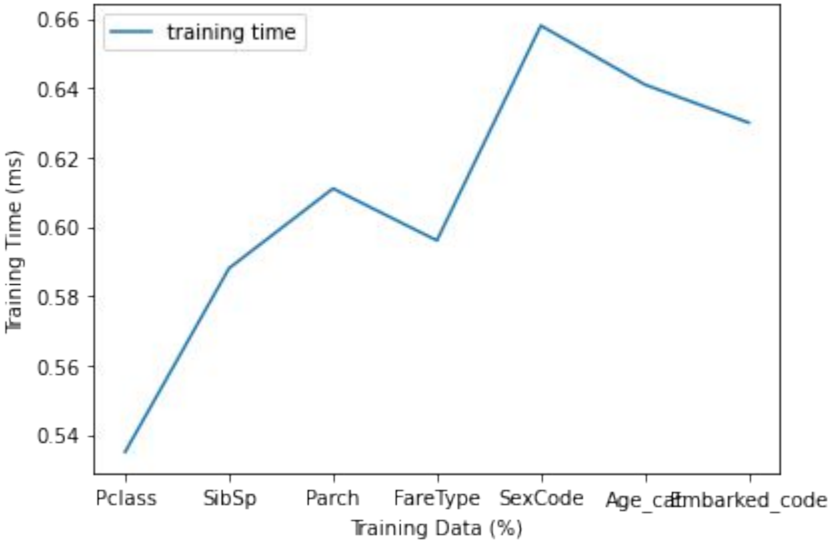
% of Training Data	acc	mcc
10%	0.6872	0.3587
20%	0.7486	0.4943
30%	0.7318	0.4528
40%	0.7598	0.5111
50%	0.7765	0.5468
60%	0.7486	0.4964
70%	0.7486	0.4964
80%	0.7486	0.4987
90%	0.7597	0.5212

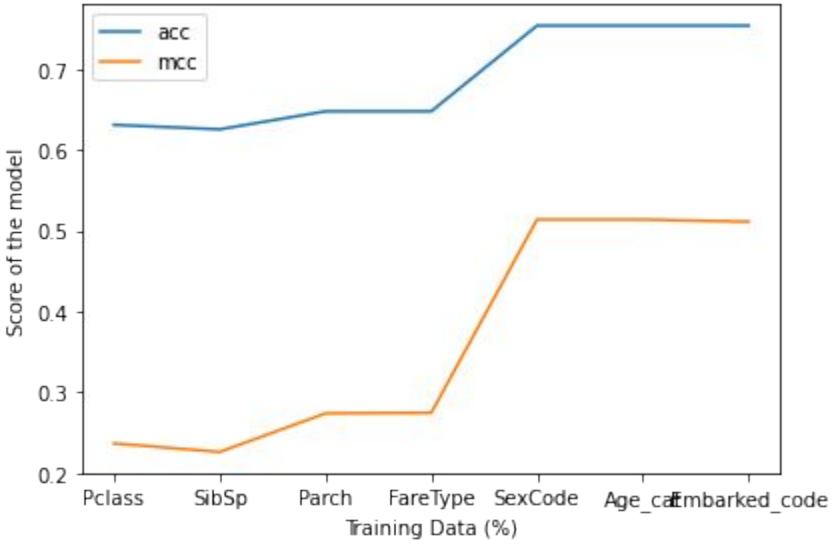
Naive Bayes: Training Time

% of Training Data	Training Time (ms)
10%	0.478
20%	0.532
30%	0.515
40%	0.527
50%	0.555
60%	0.591
70%	0.578
80%	0.606
90%	0.644



Naive Bayes: Features

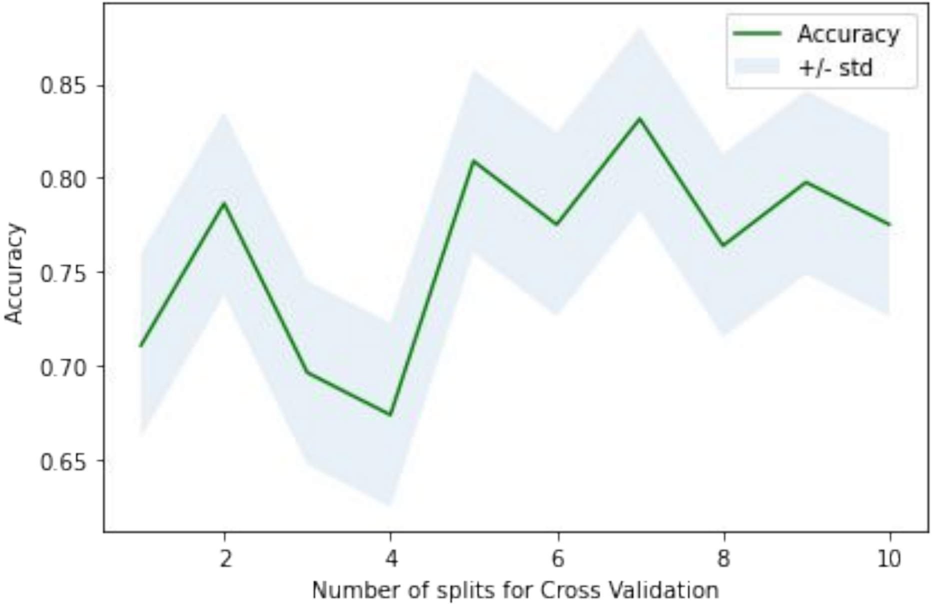




Naive Bayes: per Feature

Not survived/Survived	Pclass	SibSp	Parch	FareType	SexCode	Age_cat	Embarked_code
Mean	0.26 / -0.47	-0.01 / -0.06	-0.11 / 0.11	-0.26 / 0.38	-0.45 / 0.69	0.03 /-0.04	0.08/-0.14
Standard deviations	0.77 /1.06	1.36 /0.41	1.04 /0.91	0.9 /0.94	0.55 /0.95	0.95/1.07	0.95 / 1.05

Naive Bayes: Cross Validation



Mean accuracy of k-fold Cross Validation 0.7621

Naive Bayes: Performance