Exercise 4.2: Hyperparameter Tuning

□□□ Hyperparameter Tuning

Tuning the hyperparameters

Random Forests perform very well out-of-the-box, with the pre-set hyperparameters in `sklearn`. Some of the tunable parameters are:

- The number of trees in the forest: `n_estimators`, int, default=100
- The complexity of each tree: stop when a leaf has <= `min_samples_leaf` samples
- The sampling scheme: number of features to consider at any given split: `max_features` {“auto”, “sqrt”, “log2”}, int or float, default=”auto”.

Instructions:

- Read the datafile `diabetes.csv` as a Pandas data frame.
- Assign the predictor and response variable as mentioned in the scaffold.
- Split the data into train and validation sets.
- Define a vanilla Random Forest and fit the model on the entire data.
- For various hyper parameters of the model, define different Random Forest models and train on the data.
- Compare the results with each model.

Hints:

```
RandomForestClassifier()
```

Defines the RandomForestClassifier and includes more details on the definition and range of values for its tunable parameters.

```
model.predict_proba(X)
```

Predict class probabilities for X

```
roc_auc(y_test, y_proba)
```

Calculates the area under the receiver operating curve (AUC).

```
GridSearchCV()
```

Performs exhaustive search over specified parameter values for an estimator.