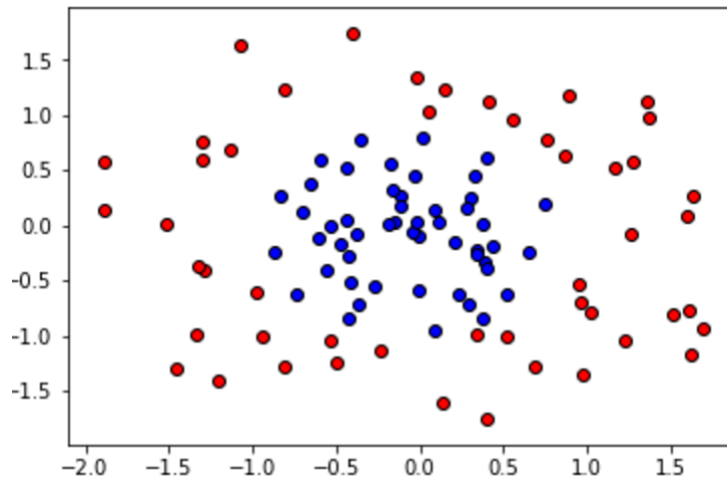


Detect Overfitting and Underfitting with Learning Curves

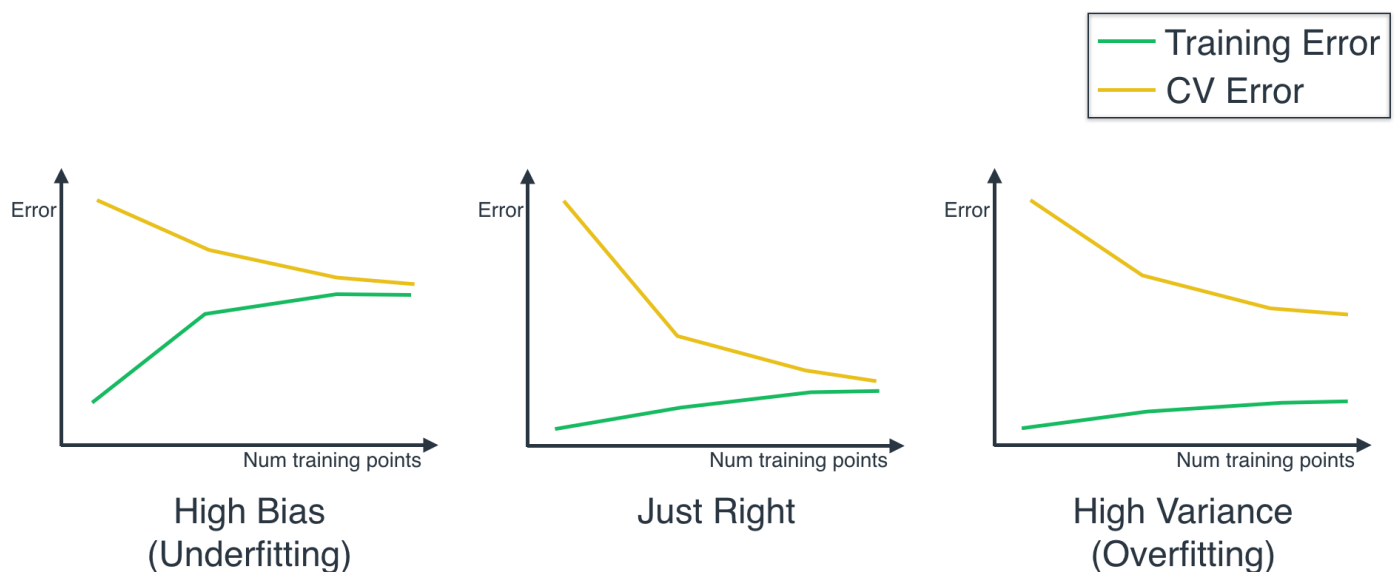
For this quiz, we'll be using three models to train the circular dataset below.

- A Decision Tree model,
- a Logistic Regression model, and
- a Support Vector Machine model.



One of the models overfits, one underfits, and the other one is just right. First, we'll write some code to draw the learning curves for each model, and finally we'll look at the learning curves to decide which model is which.

First, let's remember that the way the curves look for the three models, is as follows:



For the first part of the quiz, all you need is to uncomment one of the classifiers, and hit 'Test Run' to see the graph of the Learning Curve. But if you like coding, here are some details. We'll be using the function called `learning_curve`:

```
train_sizes, train_scores, test_scores = learning_curve(  
    estimator, X, y, cv=None, n_jobs=1, train_sizes=np.linspace(.1, 1.0, num_tr
```

No need to worry about all the parameters of this function (you can read some more in [here](#), but here we'll explain the main ones:

- `estimator`, is the actual classifier we're using for the data, e.g., `LogisticRegression()` or `GradientBoostingClassifier()`.
- `X` and `y` is our data, split into features and labels.
- `train_sizes` are the sizes of the chunks of data used to draw each point in the curve.
- `train_scores` are the training scores for the algorithm trained on each chunk of data.
- `test_scores` are the testing scores for the algorithm trained on each chunk of data.

Two very important observations:

- The training and testing scores come in as a list of 3 values, and this is because the function uses 3-Fold Cross-Validation.
- **Very important:** As you can see, we defined our curves with Training and Testing **Error**, and this function defines them with Training and Testing **Score**. These are opposite, so the higher the error, the lower the score. Thus, when you see the curve, you need to flip it upside down in your mind, in order to compare it with the curves above.

Part 1: Drawing the learning curves

In here, we'll be comparing three models:

- A **Logistic Regression** model.
- A **Decision Tree** model.
- A **Support Vector Machine** model with an rbf kernel, and a gamma parameter of 1000 (this is another type of model, don't worry about how it works for now).

Uncomment the code for each one, and examine the learning curve that gets drawn. If you're curious about the code used to draw the learning curves, it's on the **utils.py** tab.

quiz.py

utils.py

data.csv

```
1 # Import, read, and split data
2 import pandas as pd
3 data = pd.read_csv('data.csv')
4 import numpy as np
5 X = np.array(data[['x1', 'x2']])
6 y = np.array(data['y'])
7
8 # Fix random seed
9 np.random.seed(55)
10
11 ### Imports
12 from sklearn.linear_model import LogisticRegression
13 from sklearn.ensemble import GradientBoostingClassifier
14 from sklearn.svm import SVC
15
16 # TODO: Uncomment one of the three classifiers, and hit "Test Run"
17 # to see the learning curve. Use these to answer the quiz below.
18
19 ### Logistic Regression
20 estimator = LogisticRegression()
21
22 ### Decision Tree
23 # estimator = GradientBoostingClassifier()
24
25 ### Support Vector Machine
26 # estimator = SVC(kernel='rbf', gamma=1000)
```

RESET QUIZ

TEST RUN

SUBMIT ANSWER

Part 2: Analyzing the learning curves

For this second part of the quiz, you can look at the curves you've drawn before, to decide which one of the three models underfits, which one overfits, and which one is just right.

QUESTION 2 OF 2

From the models above, which model underfits, which one overfits, and which one is just right?

Submit to check your answer choices!

MODEL

OVERFITS, UNDEFITS, JUST RIGHT?

Logistic Regression

Underfits

Decision Tree

Just Right

Support Vector Machine

Overfit

SUBMIT