1. Change the number of neurons in the hidden layer and plot the performance for varying number of hidden neurons wrt. the training set for various iterations.

```
import pandas as pd

df = pd.read_csv('../weather.csv').iloc[0:200]

data = df.drop(['RainTomorrow'], axis = 1)
target = df['RainTomorrow']

target = target.replace('Yes', '1')
target = target.replace('No', '0')

from sklearn.model_selection import train_test_split
datasets = train_test_split(data, target,
                            test_size=0.2)

train_data, test_data, train_labels, test_labels = datasets
```

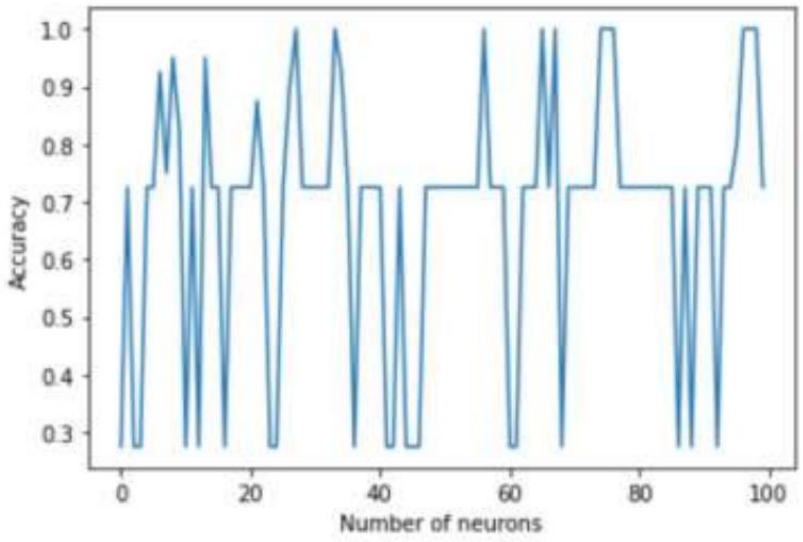
```
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score

acc = np.zeros(100)

for i in range(100):
    mlp = MLPClassifier(hidden_layer_sizes=(i+1,), max_iter=1000)
    mlp.fit(train_data, train_labels)
    predictions_test = mlp.predict(test_data)
    acc[i] = accuracy_score(predictions_test, test_labels)

plt.xlabel("Number of neurons")
plt.ylabel("Accuracy")
plt.plot(acc)
```

```
Out[386]: [<matplotlib.lines.Line2D at 0x7fd4abc10e20>]
```



2. Change the number of hidden layers for the number of neurons with best performance from 1 and plot the performance for varying number of hidden neurons wrt. the training set for various iterations.

```
n = np.argmax(acc)
print("Number of neurons = ", n)

arr = np.zeros(5)

mlp = MLPClassifier(hidden_layer_sizes=(n, ), max_iter=1000)
mlp.fit(train_data, train_labels)
predictions_test = mlp.predict(test_data)
arr[0] = accuracy_score(predictions_test, test_labels)

mlp = MLPClassifier(hidden_layer_sizes=(n, n), max_iter=1000)
mlp.fit(train_data, train_labels)
predictions_test = mlp.predict(test_data)
arr[1] = accuracy_score(predictions_test, test_labels)

mlp = MLPClassifier(hidden_layer_sizes=(n, n, n), max_iter=1000)
mlp.fit(train_data, train_labels)
predictions_test = mlp.predict(test_data)
arr[2] = accuracy_score(predictions_test, test_labels)

mlp = MLPClassifier(hidden_layer_sizes=(n, n, n, n), max_iter=1000)
mlp.fit(train_data, train_labels)
predictions_test = mlp.predict(test_data)
arr[3] = accuracy_score(predictions_test, test_labels)

mlp = MLPClassifier(hidden_layer_sizes=(n, n, n, n, n), max_iter=1000)
mlp.fit(train_data, train_labels)
predictions_test = mlp.predict(test_data)
arr[4] = accuracy_score(predictions_test, test_labels)

plt.xlabel("Number of hidden layers")
plt.ylabel("Accuracy")
plt.plot(arr)
```

```
Number of neurons =  27
```

Out[387]: `[<matplotlib.lines.Line2D at 0x7fd4abd5a1f0>]`