Question 5

Max accuracy was achieved when I used the following hyperparameters:

- `n_estimators=700`
- `min_samples_leaf=2`
- `max_depth=4`
- `max_features=None`
- `learning_rate=0.1`

For **best** implementation:

```
Accuracy: 0.9974782852339591
Precision: 0.997039230199852
Recall: 1.0
```

As no of trees increases, all three (accuracy, precision, recall) seem to improve. Gradient boosting is robust to overfitting, and hence, it performs better as we increase n_estimators.

```
For n_estimators=100:

Accuracy: 0.9948164752031381
Precision: 0.9940949725252194
Recall: 0.9998350243339107

For n_estimators=300:

Accuracy: 0.9965676660128888
Precision: 0.996137409598948
Recall: 0.9998350243339107

For n_estimators=500:

Accuracy: 0.997128047072009
Precision: 0.9967927631578948
Recall: 0.9998350243339107
```

For `n_estimators=700`:

```
Accuracy: 0.997338189969179
Precision: 0.9969569865942923
Recall: 0.9999175121669553
```

For a **simple decision tree** implementation:

```
Accuracy: 0.9912440459512468
Precision: 0.9952121512299819
Recall: 0.99447331518601
```

Clearly, gradient boosting is much better than a simple decision tree in terms of accuracy. Precision, recall too, are better for boosting method.