

lab11_random forest

November 15, 2022

```
[ ]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
from sklearn.ensemble import AdaBoostClassifier
from sklearn.svm import SVC
```

```
[ ]: dataset = pd.read_csv('C:
    ↪\Coding\ML_python\machine-learning-lab-main\datasets\bill_authentication (1).
    ↪csv')
```

```
[ ]: dataset.head()
```

```
[ ]:      Variance  Skewness  Curtosis  Entropy  Class
0    3.62160    8.6661   -2.8073 -0.44699    0
1    4.54590    8.1674   -2.4586 -1.46210    0
2    3.86600   -2.6383    1.9242  0.10645    0
3    3.45660    9.5228   -4.0112 -3.59440    0
4    0.32924   -4.4552    4.5718 -0.98880    0
```

```
[ ]: X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, 4].values
```

```
[ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    ↪random_state=0)
```

```
[ ]: sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
[ ]: regressor = RandomForestRegressor(n_estimators=20, random_state=0)
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
```

```
[ ]: print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
```

```
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

Mean Absolute Error: 0.018000000000000006

Mean Squared Error: 0.009536363636363636

Root Mean Squared Error: 0.09765430679884854

```
[ ]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
[ ]: abc = AdaBoostClassifier(n_estimators=50, learning_rate=1)
      model = abc.fit(X_train, y_train)
      y_pred = model.predict(X_test)
```

```
[ ]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.9927184466019418

```
[ ]: svc=SVC(probability=True, kernel='linear')
      abc = AdaBoostClassifier(n_estimators=50, base_estimator=svc, learning_rate=1)
      model = abc.fit(X_train, y_train)
```

```
[ ]: y_pred = model.predict(X_test)
```

```
[ ]: print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.720873786407767