

Assignment 11

Referral id: SIRSS1088

NAME: ONASVEE BANARSE

EMAIL : 2obanarse@gmail.com

COLLEGE: AISSMS IOIT

GitHub : https://github.com/ORION-22/RegexSoftware_ASSIGNMENT.git

Kaggle:<https://www.kaggle.com/c/leaf-classification/data>

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
```

```
In [ ]: train_data=pd.read_csv('../input/leafclassification/train.csv')
test_data=pd.read_csv('../input/leafclassification/test.csv')
```

```
In [ ]: train_data.head(5)
```

```
In [ ]: test_data.head(5)
```

```
In [ ]: train_data.info()
```

```
In [ ]: train_data.describe()
```

```
In [ ]: train_data.columns
```

```
In [ ]: test_data.columns
```

```
In [ ]: print('train shape ',train_data.shape)
print('test shape ',test_data.shape)
```

```
In [ ]: train_data.nunique()
```

```
In [ ]: print(train_data.isnull().sum())
print(test_data.isnull().sum())
```

```
In [ ]: from sklearn.preprocessing import LabelEncoder
enc=LabelEncoder()
train_data['species']=enc.fit_transform(train_data['species'])
```

```
In [ ]: sns.heatmap(data=train_data)
```

```
In [ ]: sns.heatmap(data=train_data.corr())
```

```
In [ ]: X=train_data.drop(['id','species'],axis=1).values
Y=train_data[['species']].values
print(X.shape,Y.shape)
```

```
In [ ]: scaler = StandardScaler().fit(X)
X = scaler.transform(X)
```

```
In [ ]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.3,random_state=5)
```

RandomForestClassifier

```
In [ ]: from sklearn.ensemble import RandomForestClassifier
```

```
rf_classifier = RandomForestClassifier(n_estimators = 20,criterion = 'entropy', max_depth = 20, random_state = 5)
rf_classifier.fit(x_train, y_train)
```

```
In [ ]: pred_train = rf_classifier.predict(x_train)
pred_test = rf_classifier.predict(x_test)
```

```
In [ ]: from sklearn.metrics import accuracy_score
print('Training Accuracy: ', accuracy_score(y_train, pred_train))
print('Testing Accuracy: ', accuracy_score(y_test, pred_test))
```

```
In [ ]: y_pred=rf_classifier.predict(x_test)
print(confusion_matrix(y_test,y_pred))
```

```
In [ ]: print(accuracy_score(y_test,y_pred))
print(classification_report(y_test,y_pred))
```

```
In [ ]: test_data.head()
```

```
In [ ]: test_ids = test_data.pop('id')
```

```
In [ ]: x_test = test_data.values
```

```
In [ ]: x_test = scaler.transform(x_test)
y_test = rf_classifier.predict_proba(x_test)
```

```
In [ ]: submission = pd.DataFrame(y_test, index=test_ids, columns=enc.classes_)
```

```
In [ ]: submission.head(5)
```

```
In [ ]: submission.to_csv('submission_leaf_classification.csv')
```

XGBoost

```
In [ ]: import xgboost as xgb
xg_classifier = xgb.XGBClassifier(n_estimators = 20)
xg_classifier.fit(x_train, y_train)
```

```
In [ ]: pred_train = xg_classifier.predict(x_train)
pred_test = xg_classifier.predict(x_test)

print('Training Accuracy: ', accuracy_score(y_train, pred_train))
print('Testing Accuracy: ', accuracy_score(y_test, pred_test))
```