

## Random Forest Algorithm

```
> ~
rf=RandomForestClassifier()
rf.fit(X_train_res,y_train_res)

[71]

... RandomForestClassifier()

y_pred1 = rf.predict(X_test)
print(confusion_matrix(y_test,y_pred1))
print(accuracy_score(y_test,y_pred1))
print(classification_report(y_test,y_pred1))

[72]

... [[20633 20884]
 [ 2470 3905]]
0.8434621201704936
precision    recall  f1-score   support

0         0.89     0.91     0.90    22717
1         0.65     0.61     0.63     6375
```

## Logistic Regression

```
> ~
logreg = LogisticRegression()
logreg.fit(X_train_res, y_train_res)

[74] Python

c:\users\admin\anaconda3\envs\rainprediction\lib\site-packages\sklearn\linear_model\_logistic.py:765: ConvergenceWarning: lbfgs failed to converge
(status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
https://scikit-learn.org/stable/modules/linear\_model.html#logistic-regression
extra_warning_msg= LOGISTIC_SOLVER_CONVERGENCE_MSG)

LogisticRegression()
```

## Gaussian NB

```
> ~
gnb = GaussianNB()
gnb.fit(X_train_res, y_train_res)

[77]

... GaussianNB()

+ Code + Markdown

y_pred3 = gnb.predict(X_test)
print(confusion_matrix(y_test,y_pred3))
print(accuracy_score(y_test,y_pred3))
print(classification_report(y_test,y_pred3))

[78]

... [[17078 5639]
 [ 1661 4714]]
0.7490719098033823
precision    recall  f1-score   support

0         0.91     0.75     0.82    22717
1         0.46     0.74     0.56     6375
```

## XGB Classifier

```
xgb = XGBClassifier()
xgb.fit(X_train_res, y_train_res)
```

[87] Python

c:\users\admin\anaconda3\envs\rainprediction\lib\site-packages\xgboost\sklearn.py:888: UserWarning: The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use\_label\_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num\_class - 1].

warnings.warn(label\_encoder\_deprecation\_msg, UserWarning)

[09:09:27] WARNING: C:/Users/Administrator/workspace/xgboost-win64\_release.1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval\_metric if you'd like to restore the old behavior.

XGBClassifier(base\_score=0.5, booster='gbtree', colsample\_bylevel=1, colsample\_bynode=1, colsample\_bytree=1, gamma=0, gpu\_id=-1, importance\_type='gain', interaction\_constraints='', learning\_rate=0.300000012, max\_delta\_step=0, max\_depth=6, min\_child\_weight=1, missing=nan, monotone\_constraints=()), n\_estimators=100, n\_jobs=8, num\_parallel\_tree=1, random\_state=0, reg\_alpha=0, reg\_lambda=1, scale\_pos\_weight=1, subsample=1, tree\_method='exact', validate\_parameters=1, verbosity=None)

## SVC Algorithm

```
svc = SVC()
svc.fit(X_train_res, y_train_res)
```

[88]

... SVC()

```
y_pred5 = svc.predict(X_test)
print(confusion_matrix(y_test, y_pred5))
print(accuracy_score(y_test, y_pred5))
print(classification_report(y_test, y_pred5))
```

[89]

```
[[17824  4893]
 [ 1594  4781]]
0.7770177368348687
```

	precision	recall	f1-score	support
0	0.92	0.78	0.85	22717
1	0.49	0.75	0.60	6375