In [2]:
```
import numpy as np
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import classification_report
import warnings
warnings.filterwarnings('ignore')
```

In [14]:
```
x,y=make_classification(n_samples=1000,n_features=10,n_informative=2, n_redundan
                        weights=[0.9,0.1],flip_y=0,random_state=42)
np.unique(y, return_counts=True)
```

Out[14]:
```
(array([0, 1]), array([900, 100]))
```

In [15]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y, test_size=0.3,stratify=y,ran
```

Handle class imbalance

In [18]:
```
from imblearn.combine import SMOTETomek

smt=SMOTETomek(random_state=42)
x_train_res,y_train_res=smt.fit_resample(x_train,y_train)
np.unique(y_train_res,return_counts=True)
```

Out[18]:
```
(array([0, 1]), array([619, 619]))
```

Track Experiments

In [20]:
```
models=[
    (
        "Logistic Regression",
        {"C":1,"solver":'lbfgs'},
        LogisticRegression(),
        (x_train,y_train),
        (x_test,y_test)
    ),
    (
        "Random Forest",
        {"n_estimators":30,"max_depth":3},
        RandomForestClassifier(),
        (x_train,y_train),
        (x_test,y_test)
    ),
    (
        "XGBClassifier",
        {"use_label_encoder": False,"eval_metric":'logloss'},
        XGBClassifier(),
        (x_train,y_train),
        (x_test,y_test)
    ),
    (
```

```
        "XGBClassifier With SMOTE",
        {"use_label_encoder":False,"eval_metric":'logloss'},
        XGBClassifier(),
        (x_train_res,y_train_res),
        (x_test,y_test)
    )
]
```

In [22]:
```
reports=[]

for model_name,params,model,train_set,test_set in models:
    x_train=train_set[0]
    y_train=train_set[1]
    x_test=test_set[0]
    y_test=test_set[1]

    model.set_params(**params)
    model.fit(x_train,y_train)
    y_pred=model.predict(x_test)
    report=classification_report(y_test,y_pred,output_dict=True)
    reports.append(report)
```

In [23]:
```
import mlflow
import mlflow.sklearn
import mlflow.xgboost
```

In [25]:
```
mlflow.set_experiment("deployment model")
mlflow.set_tracking_uri("http://localhost:5000")

for i, element in enumerate(models):
    model_name=element[0]
    params=element[1]
    model=element[2]
    report=reports[i]

    with mlflow.start_run(run_name=model_name):
        mlflow.log_params(params)
        mlflow.log_metrics({
            'accuracy': report['accuracy'],
            'recall_class_1': report['1']['recall'],
            'recall_class_0': report['0']['recall'],
            'f1_score_macro': report['macro avg']['f1-score']
        })

        if "XGB" in model_name:
            mlflow.xgboost.log_model(model,"model")
        else:
            mlflow.sklearn.log_model(model,"model")
```

```
2025/09/28 14:43:37 INFO mlflow.tracking.fluent: Experiment with name 'deployment
model' does not exist. Creating a new experiment.
2025/09/28 14:43:38 WARNING mlflow.models.model: `artifact_path` is deprecated. P
lease use `name` instead.
2025/09/28 14:43:54 WARNING mlflow.models.model: Model logged without a signature
and input example. Please set `input_example` parameter when logging the model to
auto infer the model signature.
2025/09/28 14:43:54 WARNING mlflow.models.model: `artifact_path` is deprecated. P
lease use `name` instead.
🏃 View run Logistic Regression at: http://localhost:5000/#/experiments/845712284
558116041/runs/5d6d5e891c91446c94b54ff14b42710f
🧪 View experiment at: http://localhost:5000/#/experiments/845712284558116041
2025/09/28 14:44:02 WARNING mlflow.models.model: Model logged without a signature
and input example. Please set `input_example` parameter when logging the model to
auto infer the model signature.
2025/09/28 14:44:02 WARNING mlflow.models.model: `artifact_path` is deprecated. P
lease use `name` instead.
🏃 View run Random Forest at: http://localhost:5000/#/experiments/845712284558116
041/runs/bc15d05f42bd4be38fc246d98321dca7
🧪 View experiment at: http://localhost:5000/#/experiments/845712284558116041
2025/09/28 14:44:11 WARNING mlflow.models.model: Model logged without a signature
and input example. Please set `input_example` parameter when logging the model to
auto infer the model signature.
2025/09/28 14:44:11 WARNING mlflow.models.model: `artifact_path` is deprecated. P
lease use `name` instead.
🏃 View run XGBClassifier at: http://localhost:5000/#/experiments/845712284558116
041/runs/46e55c81b7c242c0b18748e7c995095d
🧪 View experiment at: http://localhost:5000/#/experiments/845712284558116041
2025/09/28 14:44:19 WARNING mlflow.models.model: Model logged without a signature
and input example. Please set `input_example` parameter when logging the model to
auto infer the model signature.
🏃 View run XGBClassifier With SMOTE at: http://localhost:5000/#/experiments/8457
12284558116041/runs/6fb7320425c84f249cf4a35b3feda281
🧪 View experiment at: http://localhost:5000/#/experiments/845712284558116041
```

Register the Model

In [29]:
```
model_name='XGB-Smote'
run_id=input('please type RunID')
model_uri=f'runs:/{run_id}/model'

with mlflow.start_run(run_id=run_id):
    mlflow.register_model(model_uri,name=model_name)
```

```
Registered model 'XGB-Smote' already exists. Creating a new version of this mode
l...
2025/09/28 16:19:39 WARNING mlflow.tracking._model_registry.fluent: Run with id 6
fb7320425c84f249cf4a35b3feda281 has no artifacts at artifact path 'model', regist
ering model based on models:/m-992e777e4a1040e3bf9d037b90c4d9e8 instead
2025/09/28 16:19:39 INFO mlflow.store.model_registry.abstract_store: Waiting up t
o 300 seconds for model version to finish creation. Model name: XGB-Smote, versio
n 2
🏃 View run XGBClassifier With SMOTE at: http://localhost:5000/#/experiments/8457
12284558116041/runs/6fb7320425c84f249cf4a35b3feda281
🧪 View experiment at: http://localhost:5000/#/experiments/845712284558116041
Created version '2' of model 'XGB-Smote'.
```

Load the Model

In [30]:
```
model_name='XGB-Smote'
model_version=1
model_uri=f"models:/{model_name}/{model_version}"

loaded_model=mlflow.xgboost.load_model(model_uri)
y_pred=loaded_model.predict(x_test)
y_pred[:4]
```

```
Downloading artifacts:   0%|          | 0/5 [00:00<?, ?it/s]
```

Out[30]:
```
array([0, 0, 0, 0])
```

Transition the Model to Production server

In [31]:
```
current_model_uri=f"models:/{model_name}@appserver"
production_model_name="finalproduction"

client=mlflow.MlflowClient()
client.copy_model_version(src_model_uri=current_model_uri,dst_name=production_mo
```

```
Registered model 'finalproduction' already exists. Creating a new version of this
model...
Copied version '1' of model 'XGB-Smote' to version '2' of model 'finalproductio
n'.
```

Out[31]:
```
<ModelVersion: aliases=[], creation_timestamp=1759056624301, current_stage='Non
e', deployment_job_state=<ModelVersionDeploymentJobState: current_task_name='',
job_id='', job_state='DEPLOYMENT_JOB_CONNECTION_STATE_UNSPECIFIED', run_id='',
run_state='DEPLOYMENT_JOB_RUN_STATE_UNSPECIFIED'>, description='', last_updated
_timestamp=1759056624301, metrics=None, model_id=None, name='finalproduction',
params=None, run_id='6fb7320425c84f249cf4a35b3feda281', run_link='', source='mo
dels:/XGB-Smote/1', status='READY', status_message=None, tags={}, user_id='', v
ersion='2'>
```

In [ ]: