Model 5 Metrics

**Final model** =

X = scaled\_data[['ETH', 'ETH T-1', 'SP500', 'SP500 T-1', 'SP500 T-2']]

y = scaled\_data['ETH Yes/No']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20, random\_state = 42)

X\_train = X\_train.dropna()

X\_test = X\_test.dropna()

prac\_model = RandomForestClassifier()

prac\_model = RandomizedSearchCV(estimator = prac\_model, param\_distributions = random\_grid, n\_iter = 100, cv = 3, verbose=2, random\_state=42, n\_jobs = -1)

prac\_model.fit(X\_train, y\_train)

prac\_model.best\_params\_

model5 = RandomForestClassifier(n\_estimators = 90, min\_samples\_split = 2, min\_samples\_leaf = 2, max\_features = 'auto', max\_depth = 80, bootstrap = True)

model5.fit(X\_train, y\_train)

y\_pred = model5.predict(X\_test)

f1\_score(y\_test, y\_pred, average= 'micro')

**Features =**

X = scaled\_data[['ETH', 'ETH T-1', 'SP500', 'SP500 T-1', 'SP500 T-2']]

y = scaled\_data['ETH Yes/No']

X\_train, y\_train, X\_test, y\_test

**Parameters =**

'n\_estimators': 90,

'min\_samples\_split': 2,

'min\_samples\_leaf': 2,

'max\_features': 'auto',

'max\_depth': 80,

'bootstrap': True

**Performance Metrics =**

F1 score = 0.62.5

Precision = 0.62

Recall = 0.63