assign

April 22, 2024

[1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
↪f1_score
from sklearn.impute import SimpleImputer

data = pd.read_csv("/content/breast_cancer_survival[1].csv")
print(data.head())
```

```
   Age  Gender  Protein1  Protein2  Protein3  Protein4 Tumour_Stage  \
0   42  FEMALE   0.95256   2.15000  0.007972 -0.048340           II
1   54  FEMALE   0.00000   1.38020 -0.498030 -0.507320           II
2   63  FEMALE  -0.52303   1.76400 -0.370190  0.010815           II
3   78  FEMALE  -0.87618   0.12943 -0.370380  0.132190            I
4   42  FEMALE   0.22611   1.74910 -0.543970 -0.390210           II

                      Histology ER status PR status HER2 status Surgery_type  \
0  Infiltrating Ductal Carcinoma  Positive  Positive    Negative        Other
1  Infiltrating Ductal Carcinoma  Positive  Positive    Negative        Other
2  Infiltrating Ductal Carcinoma  Positive  Positive    Negative   Lumpectomy
3  Infiltrating Ductal Carcinoma  Positive  Positive    Negative        Other
4  Infiltrating Ductal Carcinoma  Positive  Positive    Positive   Lumpectomy

  Date_of_Surgery Date_of_Last_Visit Patient_Status
0       20-May-18          26-Aug-18          Alive
1       26-Apr-18          25-Jan-19           Dead
2       24-Aug-18          08-Apr-20          Alive
3       16-Nov-18          28-Jul-20          Alive
4       12-Dec-18          05-Jan-19          Alive
```

[2]:
```
data = data.drop(['Histology', 'ER status', 'PR status', 'HER2 status',␣
↪'Surgery_type', 'Date_of_Surgery', 'Date_of_Last_Visit'], axis=1)

data = pd.get_dummies(data, columns=['Gender', 'Tumour_Stage'])

print("Missing Values:")
```

```
print(data.isna().sum())

imputer = SimpleImputer(strategy='most_frequent')
data_imputed = imputer.fit_transform(data)

data_imputed = pd.DataFrame(data_imputed, columns=data.columns)

X = data_imputed.drop("Patient_Status", axis=1)
y = data_imputed["Patient_Status"].map({'Alive': 1, 'Dead': 0})
```

```
Missing Values:
Age                  0
Protein1             0
Protein2             0
Protein3             0
Protein4             0
Patient_Status      13
Gender_FEMALE        0
Gender_MALE          0
Tumour_Stage_I       0
Tumour_Stage_II      0
Tumour_Stage_III     0
dtype: int64
```

[3]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
rf_model = RandomForestClassifier()
rf_model.fit(X_train, y_train)
ada_model = AdaBoostClassifier()
ada_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
ada_pred = ada_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_pred)
ada_accuracy = accuracy_score(y_test, ada_pred)
rf_precision = precision_score(y_test, rf_pred)
ada_precision = precision_score(y_test, ada_pred)
rf_recall = recall_score(y_test, rf_pred)
ada_recall = recall_score(y_test, ada_pred)
rf_f1_score = f1_score(y_test, rf_pred)
ada_f1_score = f1_score(y_test, ada_pred)
print("Random Forest Metrics:")
print("Accuracy:", rf_accuracy)
print("Precision:", rf_precision)
print("Recall:", rf_recall)
print("F1 Score:", rf_f1_score)
print("\nAdaBoost Metrics:")
print("Accuracy:", ada_accuracy)
```

```
print("Precision:", ada_precision)
print("Recall:", ada_recall)
print("F1 Score:", ada_f1_score)
```

```
Random Forest Metrics:
Accuracy: 0.8059701492537313
Precision: 0.8181818181818182
Recall: 0.9818181818181818
F1 Score: 0.8925619834710744

AdaBoost Metrics:
Accuracy: 0.7313432835820896
Precision: 0.8135593220338984
Recall: 0.8727272727272727
F1 Score: 0.8421052631578948
```