Gradient Boosting Classification

```
from warnings import filterwarnings
filterwarnings('ignore')
```

Reading the training dataset

```
import pandas as pd
df = pd.read_csv('train_titanic.csv')
df.head()
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age
SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0
1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0
1
2                             Heikkinen, Miss. Laina  female  26.0
0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0
1
4                           Allen, Mr. William Henry    male  35.0
0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
```

```
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

```
df.nunique()
```

```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

```
s = df.isna().sum()
s
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

```
s[s>0]
```

```
Age         177
Cabin       687
Embarked      2
dtype: int64
```

Seprating X and Y

```
X = df.drop(labels=['PassengerId','Name','Ticket','Survived'],axis=1)
Y = df[['Survived']]

X.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	3	male	22.0	1	0	7.2500	NaN	S
1	1	female	38.0	1	0	71.2833	C85	C
2	3	female	26.0	0	0	7.9250	NaN	S
3	1	female	35.0	1	0	53.1000	C123	S
4	3	male	35.0	0	0	8.0500	NaN	S

```
Y.head()
```

	Survived
0	0
1	1
2	1
3	1
4	0

Creating a pipline for X preprocessing

```
cat = list(X.columns[X.dtypes=='object'])
cat
```

```
['Sex', 'Cabin', 'Embarked']
```

```
con = list(X.columns[X.dtypes!='object'])
con
```

```
['Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
```

```
cat1 = ['Cabin']
cat2 = ['Sex','Embarked']
```

Preprocessing Pipeline

```
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer

# Num Pipeline
num_pipe =
Pipeline(steps=[('impute',SimpleImputer(strategy='median')),
                            ('scaler',StandardScaler())])

# Pipeline for cabin column
cat_pipe1 =
```

```
Pipeline(steps=[('imputer',SimpleImputer(strategy='constant',fill_valu
e='Unknown')),

('Ohe',OneHotEncoder(handle_unknown='ignore'))])

# Pipeline for remaining features
cat_pipe2 =
Pipeline(steps=[('imuter',SimpleImputer(strategy='most_frequent')),

('Ohe',OneHotEncoder(handle_unknown='ignore'))])

# Compose
from sklearn.compose import ColumnTransformer
pre = ColumnTransformer([('num',num_pipe,con),
                         ('cat1',cat_pipe1,cat1),
                         ('cat2',cat_pipe2,cat2)])

X_pre = pre.fit_transform(X).toarray()
X_pre

array([[ 0.82737724, -0.56573646,  0.43279337, ...,  0.        ,
         0.        ,  1.        ],
       [-1.56610693,  0.66386103,  0.43279337, ...,  1.        ,
         0.        ,  0.        ],
       [ 0.82737724, -0.25833709, -0.4745452 , ...,  0.        ,
         0.        ,  1.        ],
       ...,
       [ 0.82737724, -0.1046374 ,  0.43279337, ...,  0.        ,
         0.        ,  1.        ],
       [-1.56610693, -0.25833709, -0.4745452 , ...,  1.        ,
         0.        ,  0.        ],
       [ 0.82737724,  0.20276197, -0.4745452 , ...,  0.        ,
         1.        ,  0.        ]])

cols = pre.get_feature_names_out()
cols

array(['num__Pclass', 'num__Age', 'num__SibSp', 'num__Parch',
'num__Fare',
       'cat1__Cabin_A10', 'cat1__Cabin_A14', 'cat1__Cabin_A16',
       'cat1__Cabin_A19', 'cat1__Cabin_A20', 'cat1__Cabin_A23',
       'cat1__Cabin_A24', 'cat1__Cabin_A26', 'cat1__Cabin_A31',
       'cat1__Cabin_A32', 'cat1__Cabin_A34', 'cat1__Cabin_A36',
       'cat1__Cabin_A5', 'cat1__Cabin_A6', 'cat1__Cabin_A7',
       'cat1__Cabin_B101', 'cat1__Cabin_B102', 'cat1__Cabin_B18',
       'cat1__Cabin_B19', 'cat1__Cabin_B20', 'cat1__Cabin_B22',
       'cat1__Cabin_B28', 'cat1__Cabin_B3', 'cat1__Cabin_B30',
       'cat1__Cabin_B35', 'cat1__Cabin_B37', 'cat1__Cabin_B38',
       'cat1__Cabin_B39', 'cat1__Cabin_B4', 'cat1__Cabin_B41',
       'cat1__Cabin_B42', 'cat1__Cabin_B49', 'cat1__Cabin_B5',
       'cat1__Cabin_B50', 'cat1__Cabin_B51 B53 B55',
```

```
       'cat1__Cabin_B57 B59 B63 B66', 'cat1__Cabin_B58 B60',
       'cat1__Cabin_B69', 'cat1__Cabin_B71', 'cat1__Cabin_B73',
       'cat1__Cabin_B77', 'cat1__Cabin_B78', 'cat1__Cabin_B79',
       'cat1__Cabin_B80', 'cat1__Cabin_B82 B84', 'cat1__Cabin_B86',
       'cat1__Cabin_B94', 'cat1__Cabin_B96 B98', 'cat1__Cabin_C101',
       'cat1__Cabin_C103', 'cat1__Cabin_C104', 'cat1__Cabin_C106',
       'cat1__Cabin_C110', 'cat1__Cabin_C111', 'cat1__Cabin_C118',
       'cat1__Cabin_C123', 'cat1__Cabin_C124', 'cat1__Cabin_C125',
       'cat1__Cabin_C126', 'cat1__Cabin_C128', 'cat1__Cabin_C148',
       'cat1__Cabin_C2', 'cat1__Cabin_C22 C26', 'cat1__Cabin_C23 C25
C27',
       'cat1__Cabin_C30', 'cat1__Cabin_C32', 'cat1__Cabin_C45',
       'cat1__Cabin_C46', 'cat1__Cabin_C47', 'cat1__Cabin_C49',
       'cat1__Cabin_C50', 'cat1__Cabin_C52', 'cat1__Cabin_C54',
       'cat1__Cabin_C62 C64', 'cat1__Cabin_C65', 'cat1__Cabin_C68',
       'cat1__Cabin_C7', 'cat1__Cabin_C70', 'cat1__Cabin_C78',
       'cat1__Cabin_C82', 'cat1__Cabin_C83', 'cat1__Cabin_C85',
       'cat1__Cabin_C86', 'cat1__Cabin_C87', 'cat1__Cabin_C90',
       'cat1__Cabin_C91', 'cat1__Cabin_C92', 'cat1__Cabin_C93',
       'cat1__Cabin_C95', 'cat1__Cabin_C99', 'cat1__Cabin_D',
       'cat1__Cabin_D10 D12', 'cat1__Cabin_D11', 'cat1__Cabin_D15',
       'cat1__Cabin_D17', 'cat1__Cabin_D19', 'cat1__Cabin_D20',
       'cat1__Cabin_D21', 'cat1__Cabin_D26', 'cat1__Cabin_D28',
       'cat1__Cabin_D30', 'cat1__Cabin_D33', 'cat1__Cabin_D35',
       'cat1__Cabin_D36', 'cat1__Cabin_D37', 'cat1__Cabin_D45',
       'cat1__Cabin_D46', 'cat1__Cabin_D47', 'cat1__Cabin_D48',
       'cat1__Cabin_D49', 'cat1__Cabin_D50', 'cat1__Cabin_D56',
       'cat1__Cabin_D6', 'cat1__Cabin_D7', 'cat1__Cabin_D9',
       'cat1__Cabin_E10', 'cat1__Cabin_E101', 'cat1__Cabin_E12',
       'cat1__Cabin_E121', 'cat1__Cabin_E17', 'cat1__Cabin_E24',
       'cat1__Cabin_E25', 'cat1__Cabin_E31', 'cat1__Cabin_E33',
       'cat1__Cabin_E34', 'cat1__Cabin_E36', 'cat1__Cabin_E38',
       'cat1__Cabin_E40', 'cat1__Cabin_E44', 'cat1__Cabin_E46',
       'cat1__Cabin_E49', 'cat1__Cabin_E50', 'cat1__Cabin_E58',
       'cat1__Cabin_E63', 'cat1__Cabin_E67', 'cat1__Cabin_E68',
       'cat1__Cabin_E77', 'cat1__Cabin_E8', 'cat1__Cabin_F E69',
       'cat1__Cabin_F G63', 'cat1__Cabin_F G73', 'cat1__Cabin_F2',
       'cat1__Cabin_F33', 'cat1__Cabin_F38', 'cat1__Cabin_F4',
       'cat1__Cabin_G6', 'cat1__Cabin_T', 'cat1__Cabin_Unknown',
       'cat2__Sex_female', 'cat2__Sex_male', 'cat2__Embarked_C',
       'cat2__Embarked_Q', 'cat2__Embarked_S'], dtype=object)
```

```
len(cols)
```

```
158
```

```
X_pre = pd.DataFrame(X_pre, columns=cols)
X_pre.head()
```

```
   num__Pclass  num__Age  num__SibSp  num__Parch  num__Fare
cat1__Cabin_A10  \
0     0.827377 -0.565736    0.432793   -0.473674  -0.502445
0.0
1    -1.566107  0.663861    0.432793   -0.473674   0.786845
0.0
2     0.827377 -0.258337   -0.474545   -0.473674  -0.488854
0.0
3    -1.566107  0.433312    0.432793   -0.473674   0.420730
0.0
4     0.827377  0.433312   -0.474545   -0.473674  -0.486337
0.0

   cat1__Cabin_A14  cat1__Cabin_A16  cat1__Cabin_A19  cat1__Cabin_A20
...  \
0              0.0              0.0              0.0              0.0
...
1              0.0              0.0              0.0              0.0
...
2              0.0              0.0              0.0              0.0
...
3              0.0              0.0              0.0              0.0
...
4              0.0              0.0              0.0              0.0
...

   cat1__Cabin_F38  cat1__Cabin_F4  cat1__Cabin_G6  cat1__Cabin_T  \
0              0.0             0.0             0.0            0.0
1              0.0             0.0             0.0            0.0
2              0.0             0.0             0.0            0.0
3              0.0             0.0             0.0            0.0
4              0.0             0.0             0.0            0.0

   cat1__Cabin_Unknown  cat2__Sex_female  cat2__Sex_male
cat2__Embarked_C  \
0                  1.0               0.0             1.0
0.0
1                  0.0               1.0             0.0
1.0
2                  1.0               1.0             0.0
0.0
3                  0.0               1.0             0.0
0.0
4                  1.0               0.0             1.0
0.0

   cat2__Embarked_Q  cat2__Embarked_S
0               0.0               1.0
1               0.0               0.0
2               0.0               1.0
```

```
3                0.0                1.0
4                0.0                1.0

[5 rows x 158 columns]
```

```
X_pre.isna().sum()
```

```
num__Pclass          0
num__Age             0
num__SibSp           0
num__Parch           0
num__Fare            0
                    ..
cat2__Sex_female     0
cat2__Sex_male       0
cat2__Embarked_C     0
cat2__Embarked_Q     0
cat2__Embarked_S     0
Length: 158, dtype: int64
```

```
Y.value_counts()
```

```
Survived
0           549
1           342
Name: count, dtype: int64
```

Train test split

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(X_pre, Y,
test_size=0.2, random_state=63)
```

```
xtrain.shape
```

```
(712, 158)
```

```
xtest.shape
```

```
(179, 158)
```

Model creation

```
from sklearn.ensemble import GradientBoostingClassifier
model = GradientBoostingClassifier(random_state=42)
model.fit(xtrain, ytrain)
```

```
GradientBoostingClassifier(random_state=42)
```

```
# Training Accuracy
model.score(xtrain,ytrain)
```

```
0.8960674157303371

# Testing
model.score(xtest, ytest)

0.8268156424581006
```

Hyperparameter tuning

```
params = {'learning_rate':[0.001, 0.01, 0.05, 0.1],
          'n_estimators':[10,50,100,200],
          'max_depth':[3,4,5,6,7,8,9,10],
          'min_samples_split':[6,7,8,9,10]}

from sklearn.model_selection import RandomizedSearchCV
gbc = GradientBoostingClassifier(random_state=42)
rscv = RandomizedSearchCV(gbc, param_distributions=params, cv=5,
scoring='f1')
rscv.fit(xtrain,ytrain)

RandomizedSearchCV(cv=5,
estimator=GradientBoostingClassifier(random_state=42),
                   param_distributions={'learning_rate': [0.001, 0.01,
0.05,
                                                          0.1],
                                        'max_depth': [3, 4, 5, 6, 7,
8, 9, 10],
                                        'min_samples_split': [6, 7, 8,
9, 10],
                                        'n_estimators': [10, 50, 100,
200]},
                   scoring='f1')

rscv.best_params_

{'n_estimators': 200,
 'min_samples_split': 9,
 'max_depth': 5,
 'learning_rate': 0.1}

rscv.best_score_

0.7394691774362689

best_gbc = rscv.best_estimator_
best_gbc

GradientBoostingClassifier(max_depth=5, min_samples_split=9,
n_estimators=200,
                           random_state=42)
```

Evaluate best gbc

```
best_gbc.score(xtrain,ytrain)
```

```
0.9325842696629213
```

```
best_gbc.score(xtest, ytest)
```

```
0.8324022346368715
```

Predicting training and testing data

```
ypred_tr = best_gbc.predict(xtrain)
ypred_ts = best_gbc.predict(xtest)
```

```
ypred_tr[0:5]
```

```
array([0, 0, 0, 0, 0], dtype=int64)
```

```
ytrain.head()
```

```
     Survived
610         0
728         0
90          0
509         1
834         0
```

```
ypred_ts[0:5]
```

```
array([0, 0, 0, 1, 0], dtype=int64)
```

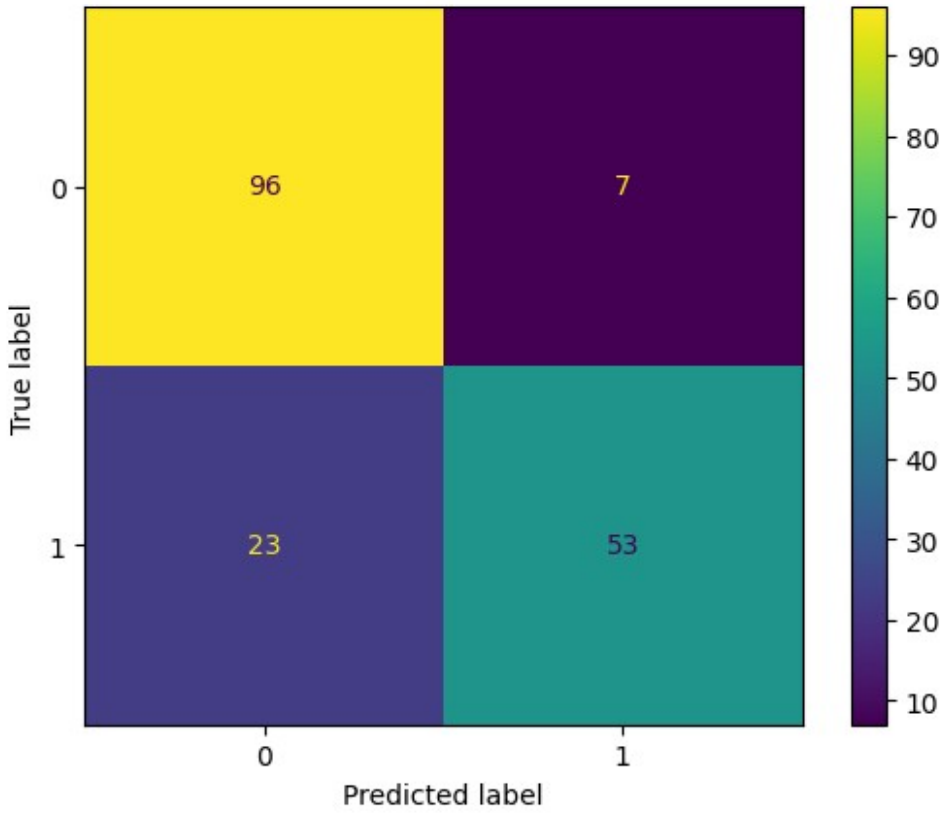
```
ytest.head()
```

```
     Survived
789         0
823         1
4           0
3           1
721         0
```

Confusion Matrix

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
cf = confusion_matrix(ytest, ypred_ts)
cfd = ConfusionMatrixDisplay(cf,display_labels=best_gbc.classes_)
cfd.plot()
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x20735f3c410>
```



Perform classification report

```
from sklearn.metrics import classification_report
print(classification_report(ytest, ypred_ts))
```

```
              precision    recall  f1-score   support

           0       0.81      0.93      0.86       103
           1       0.88      0.70      0.78        76

    accuracy                           0.83       179
   macro avg       0.85      0.81      0.82       179
weighted avg       0.84      0.83      0.83       179
```

Use the model for predictions

```
xnew = pd.read_csv('test_titanic.csv')
xnew.head()
```

```
   PassengerId  Pclass                                          Name
Sex  \
0          892       3                              Kelly, Mr. James
male
1          893       3              Wilkes, Mrs. James (Ellen Needs)
```

```
female
2          894       2                      Myles, Mr. Thomas Francis
male
3          895       3                               Wirz, Mr. Albert
male
4          896       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)
female

    Age  SibSp  Parch   Ticket     Fare Cabin Embarked
0  34.5      0      0   330911   7.8292   NaN        Q
1  47.0      1      0   363272   7.0000   NaN        S
2  62.0      0      0   240276   9.6875   NaN        Q
3  27.0      0      0   315154   8.6625   NaN        S
4  22.0      1      1  3101298  12.2875   NaN        S
```

```
xnew.isna().sum()
```

```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

```
xnew_pre = pre.transform(xnew).toarray()
xnew_pre
```

```
array([[ 0.82737724,  0.39488658, -0.4745452 , ...,  0.        ,
         1.        ,  0.        ],
       [ 0.82737724,  1.35550962,  0.43279337, ...,  0.        ,
         0.        ,  1.        ],
       [-0.36936484,  2.50825727, -0.4745452 , ...,  0.        ,
         1.        ,  0.        ],
       ...,
       [ 0.82737724,  0.70228595, -0.4745452 , ...,  0.        ,
         0.        ,  1.        ],
       [ 0.82737724, -0.1046374 , -0.4745452 , ...,  0.        ,
         0.        ,  1.        ],
       [ 0.82737724, -0.1046374 ,  0.43279337, ...,  1.        ,
         0.        ,  0.        ]])
```

```
xnew_pre = pd.DataFrame(xnew_pre,columns=cols)
xnew_pre
```

```
     num__Pclass  num__Age  num__SibSp  num__Parch  num__Fare  \
0       0.827377  0.394887   -0.474545   -0.473674  -0.490783
1       0.827377  1.355510    0.432793   -0.473674  -0.507479
2      -0.369365  2.508257   -0.474545   -0.473674  -0.453367
3       0.827377 -0.181487   -0.474545   -0.473674  -0.474005
4       0.827377 -0.565736    0.432793    0.767630  -0.401017
..           ...       ...         ...         ...        ...
413     0.827377 -0.104637   -0.474545   -0.473674  -0.486337
414    -1.566107  0.740711   -0.474545   -0.473674   1.544246
415     0.827377  0.702286   -0.474545   -0.473674  -0.502445
416     0.827377 -0.104637   -0.474545   -0.473674  -0.486337
417     0.827377 -0.104637    0.432793    0.767630  -0.198244

     cat1__Cabin_A10  cat1__Cabin_A14  cat1__Cabin_A16
cat1__Cabin_A19  \
0                0.0              0.0              0.0
0.0
1                0.0              0.0              0.0
0.0
2                0.0              0.0              0.0
0.0
3                0.0              0.0              0.0
0.0
4                0.0              0.0              0.0
0.0
..               ...              ...              ...                ..
.
413              0.0              0.0              0.0
0.0
414              0.0              0.0              0.0
0.0
415              0.0              0.0              0.0
0.0
416              0.0              0.0              0.0
0.0
417              0.0              0.0              0.0
0.0

     cat1__Cabin_A20  ...  cat1__Cabin_F38  cat1__Cabin_F4
cat1__Cabin_G6  \
0                0.0  ...              0.0             0.0
0.0
1                0.0  ...              0.0             0.0
0.0
2                0.0  ...              0.0             0.0
0.0
3                0.0  ...              0.0             0.0
0.0
4                0.0  ...              0.0             0.0
0.0
```

```
..                ...  ...                ...                ...
...
413               0.0  ...                0.0                0.0
0.0
414               0.0  ...                0.0                0.0
0.0
415               0.0  ...                0.0                0.0
0.0
416               0.0  ...                0.0                0.0
0.0
417               0.0  ...                0.0                0.0
0.0

     cat1__Cabin_T  cat1__Cabin_Unknown  cat2__Sex_female
cat2__Sex_male  \
0              0.0                  1.0               0.0
1.0
1              0.0                  1.0               1.0
0.0
2              0.0                  1.0               0.0
1.0
3              0.0                  1.0               0.0
1.0
4              0.0                  1.0               1.0
0.0
..             ...                  ...               ...
...
413            0.0                  1.0               0.0
1.0
414            0.0                  0.0               1.0
0.0
415            0.0                  1.0               0.0
1.0
416            0.0                  1.0               0.0
1.0
417            0.0                  1.0               0.0
1.0

     cat2__Embarked_C  cat2__Embarked_Q  cat2__Embarked_S
0                 0.0               1.0               0.0
1                 0.0               0.0               1.0
2                 0.0               1.0               0.0
3                 0.0               0.0               1.0
4                 0.0               0.0               1.0
..                ...               ...               ...
413               0.0               0.0               1.0
414               1.0               0.0               0.0
415               0.0               0.0               1.0
416               0.0               0.0               1.0
417               1.0               0.0               0.0
```

```
[418 rows x 158 columns]
```

Use the model for predictions

```
pred = best_gbc.predict(xnew_pre)
pred
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
0,
       1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1,
1,
       1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1,
0,
       1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
1,
       0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
1,
       1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1,
0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1,
1,
       0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1,
1,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
1,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0,
       1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1,
0,
       0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0,
0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
1,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0,
0],
      dtype=int64)
```

```
len(pred)
```

```
418
```

Saving Predictions in Dataframe

```
df_pred = xnew[['PassengerId']]
df_pred
```

```
     PassengerId
0            892
1            893
2            894
3            895
4            896
..           ...
413         1305
414         1306
415         1307
416         1308
417         1309

[418 rows x 1 columns]
```

```
df_pred['Survived_Pred'] = pred
df_pred
```

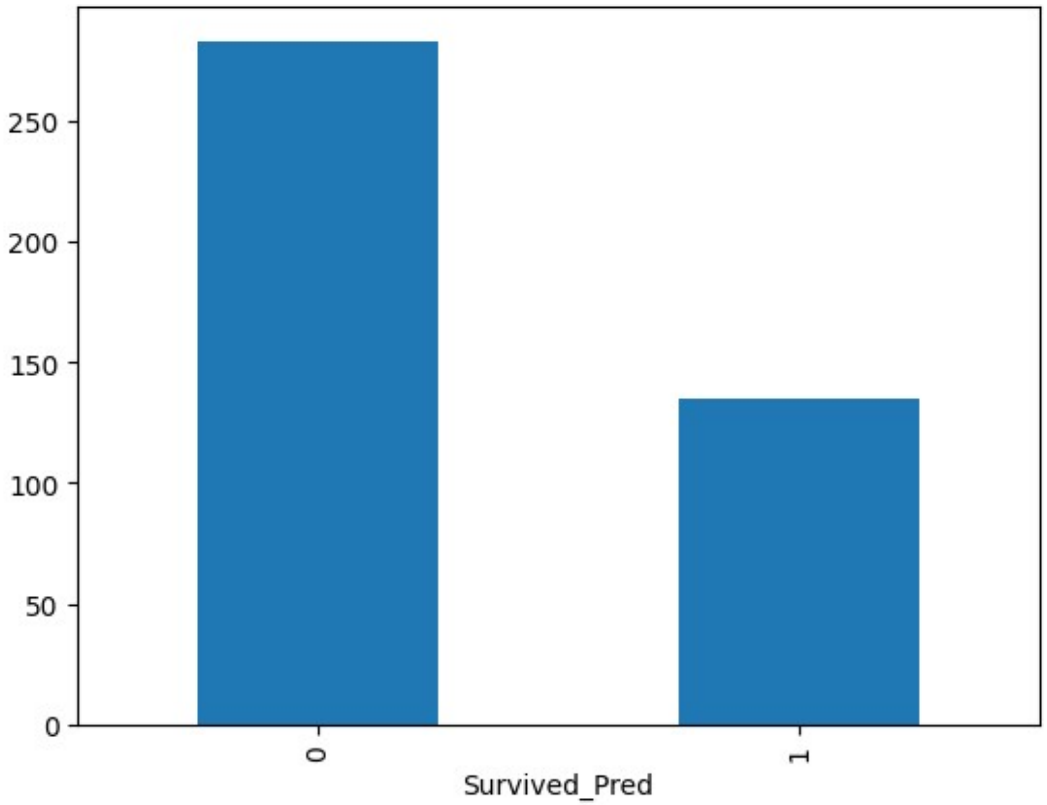
```
     PassengerId  Survived_Pred
0            892              0
1            893              0
2            894              0
3            895              0
4            896              0
..           ...            ...
413         1305              0
414         1306              1
415         1307              0
416         1308              0
417         1309              0

[418 rows x 2 columns]
```

```
df_pred['Survived_Pred'].value_counts()
```

```
Survived_Pred
0    283
1    135
Name: count, dtype: int64
```

```
df_pred['Survived_Pred'].value_counts().plot(kind='bar')
```

```
<Axes: xlabel='Survived_Pred'>
```



Save Predictions to CSV

```
df_pred.to_csv('Survived_Predictions.csv', index = False)
```