

```
8] data.head()
```

	Age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	outcome
0	65	Female	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	Male	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	Male	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	Male	3.9	2.0	195	27	59	7.3	2.4	0.40	1

```
X_train = pd.DataFrame(X_train).dropna()
y_train = y_train[X_train.index]
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
print(y_train_smote.value_counts())
```

```
1    328
2    328
Name: outcome, dtype: int64
```

```
[78] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_train_smote, y_train_smote, test_size=0.2, random_state=42)
from sklearn.ensemble import RandomForestClassifier
modell = RandomForestClassifier()
modell.fit(X_train, y_train)
y_predict = modell.predict(X_test)
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
rfc1 = accuracy_score(y_test, y_predict)
print("Accuracy:", rfc1)
print("Classification Report:\n", classification_report(y_test, y_predict))
```

```

78] from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_train_smote, y_train_smote, test_size=0.2, random_state=42)
from sklearn.ensemble import RandomForestClassifier
modell = RandomForestClassifier()
modell.fit(X_train, y_train)
y_predict = modell.predict(X_test)
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
rfc1 = accuracy_score(y_test, y_predict)
print("Accuracy:", rfc1)
print("Classification Report:\n", classification_report(y_test, y_predict))

```

Accuracy: 0.8484848484848485

Classification Report:

	precision	recall	f1-score	support
1	0.91	0.77	0.84	66
2	0.80	0.92	0.86	66
accuracy			0.85	132
macro avg	0.86	0.85	0.85	132
weighted avg	0.86	0.85	0.85	132

```
82] from sklearn.neighbors import KNeighborsClassifier
model2=KNeighborsClassifier()
model2.fit(X_train_smote, y_train_smote)
y_predict=model2.predict(X_test)
knn1=(accuracy_score(y_test, y_predict))
knn1
pd.crosstab(y_test, y_predict)
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
1	0.96	0.68	0.80	66
2	0.75	0.97	0.85	66
accuracy			0.83	132
macro avg	0.86	0.83	0.82	132
weighted avg	0.86	0.83	0.82	132

```
from sklearn.linear_model import LogisticRegression
model5=LogisticRegression()
model5.fit(X_train_smote, y_train_smote)
y_predict=model5.predict(X_test)
logi1=(accuracy_score(y_test, y_predict))
```

```
def predict_exit(sample_value):
```

```
    sample_value= np.array(sample_value)
```

```
    sample_value=sample_value.reshape(1,-1)
```

```
    sample_value=scale(sample_value)
```

```
    return classifier.predict(sample_value)
```

```
sample_value=[[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]
```

```
if predict_exit(sample_value)>0.5:
```

```
    print('Prediction: Liver Patient')
```

```
else:
```

```
    print('Prediction: Healthy')
```

```
1/1 [=====] - 0s 94ms/step
```

```
Prediction: Healthy
```

```
[94] acc_smote= [['KNN Classifier', knn1], ['RandomForestClassifier', rfc1],  
               ['DecisionTreeClassifier', dtc1], ['LogisticRegression', logi1]]
```

```
Liverpatient_pred=pd.DataFrame(acc_smote, columns=['classification models', 'accuracy_score'])
```

```
Liverpatient_pred
```



```
if predict_exit(sample_value)>0.5:
```

```
    print('Prediction: Liver Patient')
```

```
else:
```

```
    print('Prediction: Healthy')
```

```
1/1 [=====] - 0s 94ms/step
```

```
Prediction: Healthy
```

```
[94] acc_smote= [['KNN Classifier', knn1], ['RandomForestClassifier', rfc1],
```

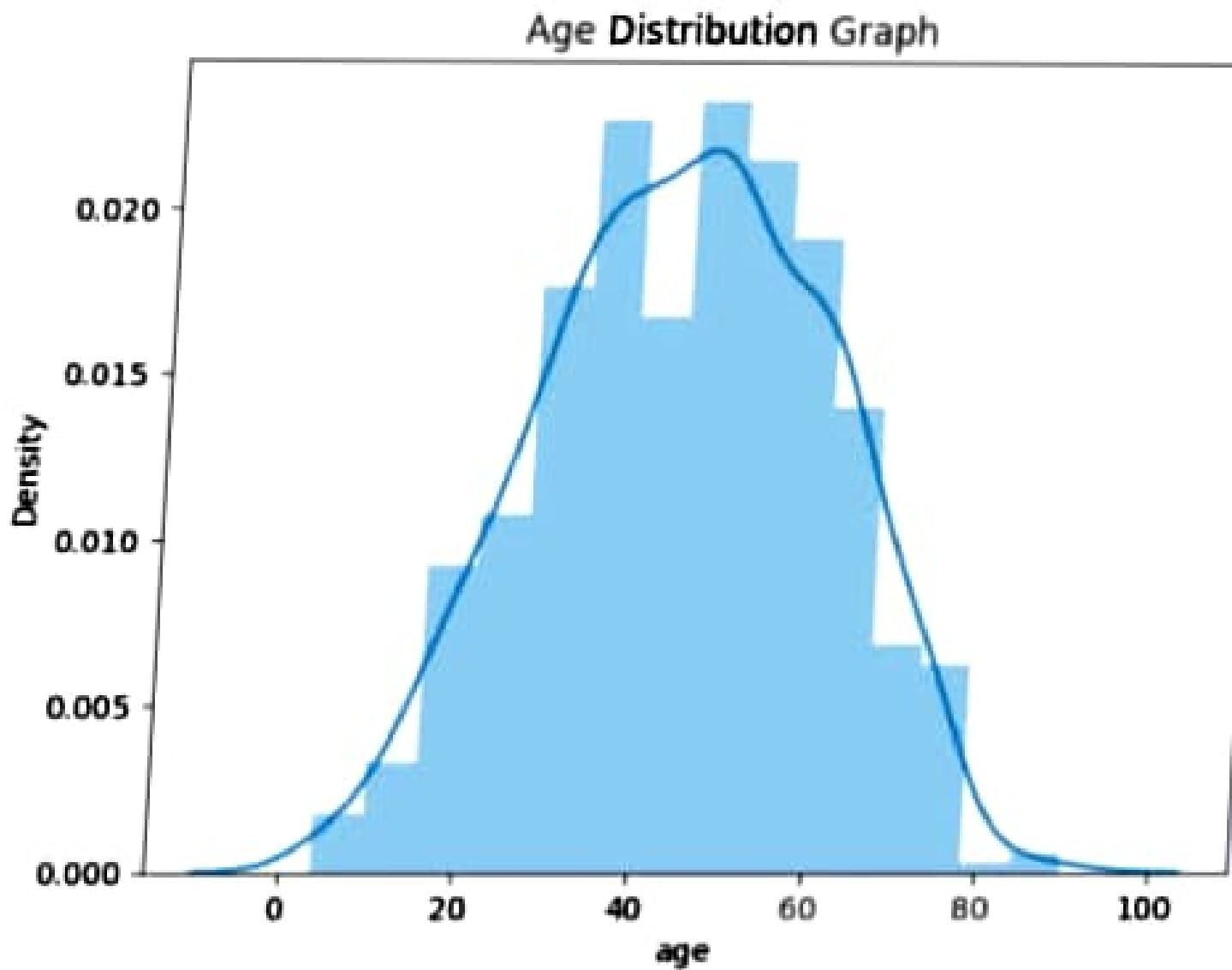
```
               ['DecisionTreeClassifier', dtc1], ['LogisticRegression', logi1]]
```

```
Liverpatient_pred=pd.DataFrame(acc_smote, columns=['classification models', 'accuracy_score'])
```

```
Liverpatient_pred
```



```
sns.distplot(data['age'])  
plt.title('Age Distribution Graph')  
plt.show()
```



```
plt.figure(figsize=(10,7))
sns.heatmap(data.corr(),annot=True)
```

<Axes: >





```
X=data.head()  
X
```



	age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	outcome
0	65	0	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	1	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	1	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	1	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	1	3.9	2.0	195	27	59	7.3	2.4	0.40	1



	age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	outcome
✓ [148] 0	65	0	0.7	0.1	187	16	18	6.8	3.3	0.90	1
1	62	1	10.9	5.5	699	64	100	7.5	3.2	0.74	1
2	62	1	7.3	4.1	490	60	68	7.0	3.3	0.89	1
3	58	1	1.0	0.4	182	14	20	6.8	3.4	1.00	1
4	72	1	3.9	2.0	195	27	59	7.3	2.4	0.40	1



```

from sklearn.preprocessing import scale
X_scaled=pd.DataFrame(scale(X),columns=X.columns)
X_scaled.head()

```

	age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	outcome
0	0.257248	-2.0	-1.044728	-1.107326	-0.779548	-0.936793	-1.131508	-1.005141	0.492458	0.542587	
1	-0.385872	0.5	1.579957	1.470071	1.660113	1.289250	1.519453	1.507712	0.218870	-0.218938	
2	-0.385872	0.5	0.653598	0.801857	0.664236	1.103746	0.484932	-0.287183	0.492458	0.494991	
3	-1.243365	0.5	-0.967531	-0.964137	-0.803372	-1.029545	-1.066850	-1.005141	0.766046	1.018540	
4	1.757860	0.5	-0.221297	-0.200464	-0.741428	-0.426658	0.193973	0.789754	-1.969832	-1.837179	



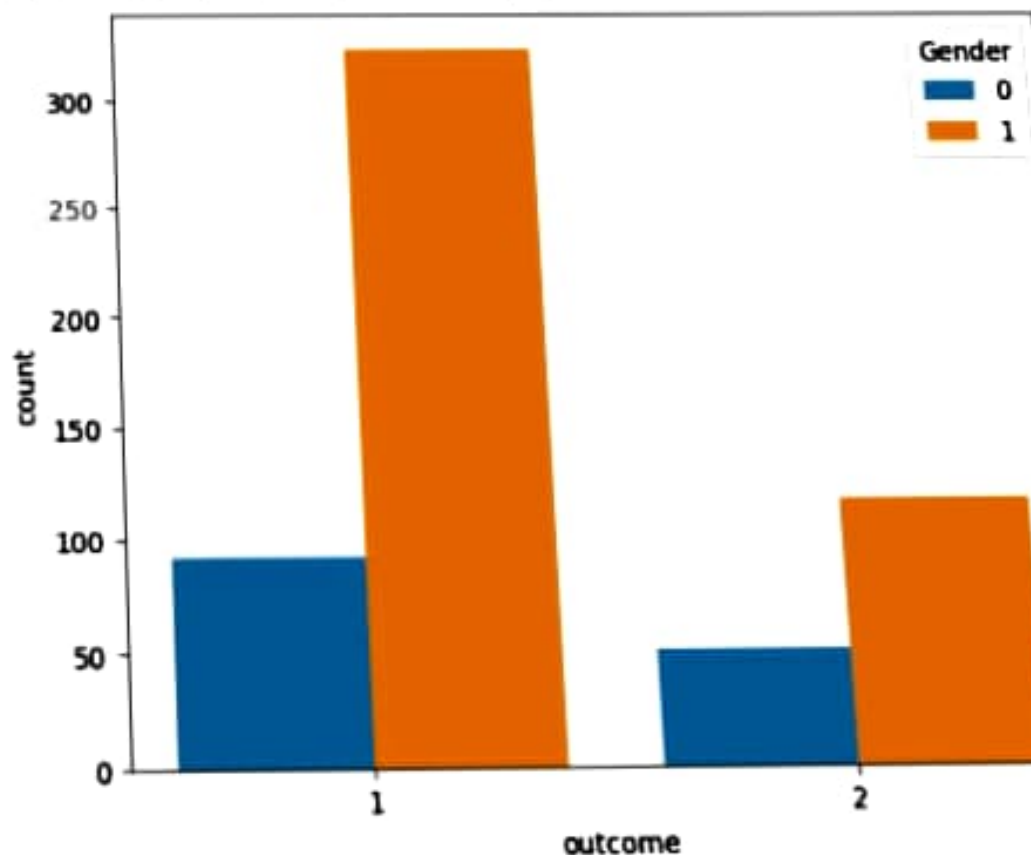
- Menaka.PE 3 (1).pdf
- Menaka.PE 3.pdf
- Noodles Manda 🍜
- PHONE DETAILS.gsheet
- Photo from ~Queen~
- Pikachu cuty 🐾
- Project.csv
- Snapchat-862608341.jpg
- Snapchat-927820151 (1)...
- Snapchat-927820151.jpg
- Unknown.wav
- Untitled spreadsheet (1)...
- Untitled spreadsheet (2)...
- Untitled spreadsheet (3)...
- Untitled spreadsheet.gsh...
- Video from ~🍜Queen🍜
- liver_patient.csv

mple_data



```
[44] sns.countplot(x='outcome', hue='Gender', data=data)
```

<Axes: xlabel='outcome', ylabel='count'>



```
Age                False
Gender             False
Total_Bilirubin    False
Direct_Bilirubin   False
Alkaline_Phosphatase False
Alamine_Aminotransferase False
Aspartate_Aminotransferase False
Total_Protiens     False
Albumin            False
Albumin_and_Globulin_Ratio True
outcome            False
dtype: bool
```

```
data.isnull().sum()
```

```
Age                0
Gender             0
Total_Bilirubin    0
Direct_Bilirubin   0
Alkaline_Phosphatase 0
Alamine_Aminotransferase 0
Aspartate_Aminotransferase 0
Total_Protiens     0
Albumin            0
Albumin_and_Globulin_Ratio 4
outcome            0
dtype: int64
```

```

from sklearn.preprocessing import LabelEncoder
Ic=LabelEncoder()
data['Gender']=Ic.fit_transform(data['Gender'])

```

data.describe()

	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	outcome
0	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	583.000000	579.000000	583.000000
1	0.756432	3.298799	1.486106	290.576329	80.713551	109.910806	6.483190	3.141852	0.947064	1.286449
3	0.429603	6.209522	2.808498	242.937989	182.620356	288.918529	1.085451	0.795519	0.319592	0.452490
0	0.000000	0.400000	0.100000	63.000000	10.000000	10.000000	2.700000	0.900000	0.300000	1.000000
0	1.000000	0.800000	0.200000	175.500000	23.000000	25.000000	5.800000	2.600000	0.700000	1.000000
0	1.000000	1.000000	0.300000	208.000000	35.000000	42.000000	6.600000	3.100000	0.930000	1.000000
0	1.000000	2.600000	1.300000	298.000000	60.500000	87.000000	7.200000	3.800000	1.100000	2.000000
0	1.000000	75.000000	19.700000	2110.000000	2000.000000	4929.000000	9.600000	5.500000	2.800000	2.000000

```
[31] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 583 entries, 0 to 582
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Age                   583 non-null   int64
1   Gender                583 non-null   object
2   Total_Bilirubin       583 non-null   float64
3   Direct_Bilirubin      583 non-null   float64
4   Alkaline_Phosphotase  583 non-null   int64
5   Alamine_Aminotransferase 583 non-null   int64
6   Aspartate_Aminotransferase 583 non-null   int64
7   Total_Protiens        583 non-null   float64
8   Albumin               583 non-null   float64
9   Albumin_and_Globulin_Ratio 579 non-null   float64
10  outcome               583 non-null   int64
dtypes: float64(5), int64(5), object(1)
memory usage: 50.2+ KB

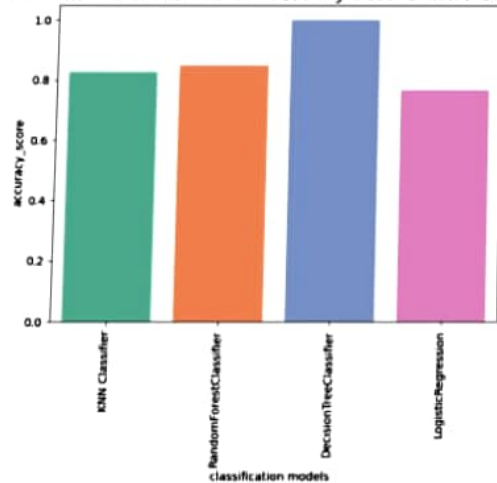
[32] data.isnull().any()

Age                False
Gender              False
Total_Bilirubin     False
Direct_Bilirubin    False
Alkaline_Phosphotase False
Alamine_Aminotransferase False
Aspartate_Aminotransferase False
Total_Protiens      False
```

```
plt.figure(figsize=(7,5))
plt.xticks(rotation=90)
plt.title('Classification models & accuracy scores after SMOTE', fontsize=18)
sns.barplot(x="classification models", y="accuracy_score", data=liverpatient_pred,palette="Set2")
```

<Axes: title={'center': 'Classification models & accuracy scores after SMOTE'}, xlabel='classification models', ylabel='accuracy_score'>

Classification models & accuracy scores after SMOTE











11

10


```
✓ [64] Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.9/dist-packages (from imblearn) (0.10.1)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.1.1)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.10.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.22.4)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (3.1.0)
Installing collected packages: imblearn
Successfully installed imblearn-0.0
```

```
✓ [65] from imblearn.over_sampling import SMOTE
smote=SMOTE()
```

```
✓ [66] y_train.value_counts()
```

```
from sklearn.tree import DecisionTreeClassifier
model4=DecisionTreeClassifier()
model4.fit(X_train_smote, y_train_smote)
y_predict=model4.predict(X_test)
dtc1=accuracy_score(y_test, y_predict)
dtc1
pd.crosstab(y_test, y_predict)
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
1	1.00	1.00	1.00	66
2	1.00	1.00	1.00	66
accuracy			1.00	132
macro avg	1.00	1.00	1.00	132
weighted avg	1.00	1.00	1.00	132

```
from sklearn.neighbors import KNeighborsClassifier
model2=KNeighborsClassifier()
```

```
from sklearn.linear_model import LogisticRegression
model5=LogisticRegression()
model5.fit(X_train_smote, y_train_smote)
y_predict=model5.predict(X_test)
logit=(accuracy_score(y_test, y_predict))
logit
pd.crosstab(y_test, y_predict)
print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
1	0.87	0.62	0.73	66
2	0.71	0.91	0.79	66
accuracy			0.77	132
macro avg	0.79	0.77	0.76	132
weighted avg	0.79	0.77	0.76	132

File

	age	Gender	Total_Bilirubin	Direct_Bilirubin	Alkaline_Phosphotase	Alamine_Aminotransferase	Aspartate_Aminotransferase	Total_Protiens	Albumin	Albumin_and_Globulin_Ratio	outc
+ Cod	0	0.257248	-2.0	-1.044728	-1.107326	-0.779548	-0.936793	-1.131508	-1.005141	0.492458	0.542587
✓ [149]	1	-0.385872	0.5	1.579957	1.470071	1.660113	1.289250	1.519453	1.507712	0.218870	-0.218938
	2	-0.385872	0.5	0.653598	0.801857	0.664236	1.103746	0.484932	-0.287183	0.492458	0.494991
	3	-1.243365	0.5	-0.967531	-0.964137	-0.803372	-1.029545	-1.066850	-1.005141	0.766046	1.018540
	4	1.757860	0.5	-0.221297	-0.200464	-0.741428	-0.426658	0.193973	0.789754	-1.969832	-1.837179



```

✓ [63] X=data.iloc[:, :-1]
      y=data.outcome
      from sklearn.model_selection import train_test_split
      X_train,x_test,y_train,y_test=train_test_split(X_scaled,y,test_size=0.2,random_state=42)

```



pip install imblearn

```

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting imblearn
  Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.9/dist-packages (from imblearn) (0.10.1)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.2.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.1.1)
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.10.1)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (1.22.4)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.9/dist-packages (from imbalanced-learn->imblearn) (3.1.0)
Installing collected packages: imblearn
Successfully installed imblearn-0.0

```