In [1]:

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

Out[1]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	100	0	A/5 21171	7.2500	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	100	0	113803	53.1000	C
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	
...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	

891 rows × 12 columns

In [2]:

```
df.isna().sum()
```

Out[2]:

```
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
```

In [3]:

```
mean_age = round(df['Age'].mean())
mean_age
```

Out[3]:

```
30
```

In [4]:

```
df['Age'].fillna(mean_age, inplace=True)
```

In [7]:

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['Cabin'] = le.fit_transform(df['Cabin'])
df['Embarked'] = le.fit_transform(df['Embarked'])
df['Ticket'] = le.fit_transform(df['Ticket'])
df['Sex'] = le.fit_transform(df['Sex'])
df['Name'] = le.fit_transform(df['Name'])
df
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emb
0	1	0	3	108	1	22.0	100	0	523	7.2500	147	
1	2	1	1	190	0	38.0	1	0	596	71.2833	81	
2	3	1	3	353	0	26.0	0	0	669	7.9250	147	
3	4	1	1	272	0	35.0	100	0	49	53.1000	55	
4	5	0	3	15	1	35.0	0	0	472	8.0500	147	
...	...	...	...	...	...	...	...	...	...	...	...	
886	887	0	2	548	1	27.0	0	0	101	13.0000	147	
887	888	1	1	303	0	19.0	0	0	14	30.0000	30	
888	889	0	3	413	0	30.0	1	2	675	23.4500	147	
889	890	1	1	81	1	26.0	0	0	8	30.0000	60	
890	891	0	3	220	1	32.0	0	0	466	7.7500	147	

891 rows × 12 columns

In [8]:

```
mode_cabin = df['Cabin'].mode()[0]
mode_cabin
```

Out[8]:

```
147
```

In [9]:

```
df['Cabin'].fillna(mode_cabin, inplace=True)
```

In [10]:

```
mode_embarked = df['Embarked'].mode()[0]
mode_embarked
```

Out[10]:

```
2
```

In [12]:

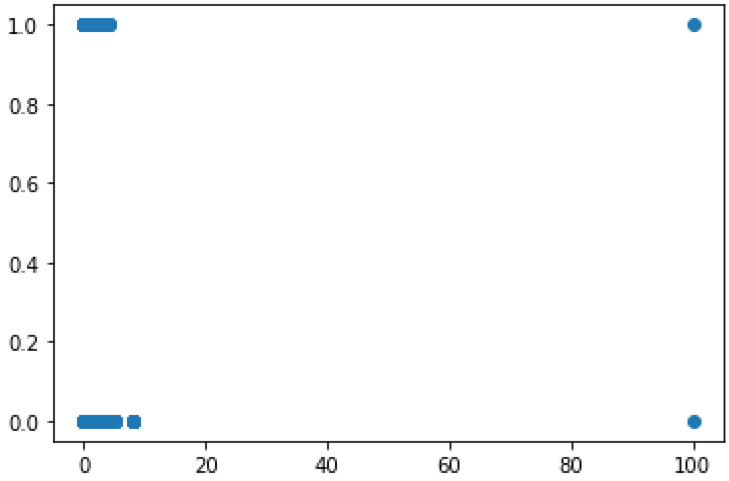
```
df['Embarked'].fillna(mode_embarked, inplace=True)
```

In [13]:
```
df.isna().sum()
```

Out[13]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [17]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(df['SibSp'], df['Survived'])
```

Out[17]:
```
<matplotlib.collections.PathCollection at 0x21309780e80>
```



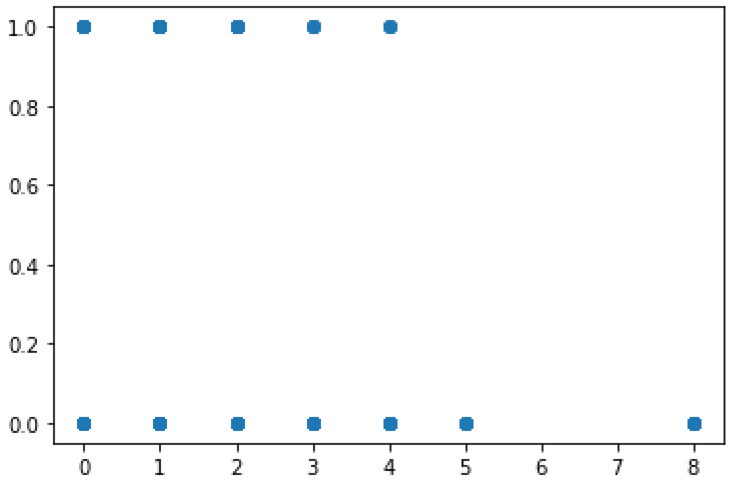
In [19]:
```
mode_sibsp = df['SibSp'].mode()[0]
for x in range(0,891):
    if df['SibSp'][x] >= 20:
        df['SibSp'][x] = mode_sibsp
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_10164\4235250280.py:4: SettingWithCo
pyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['SibSp'][x] = mode_sibsp
```

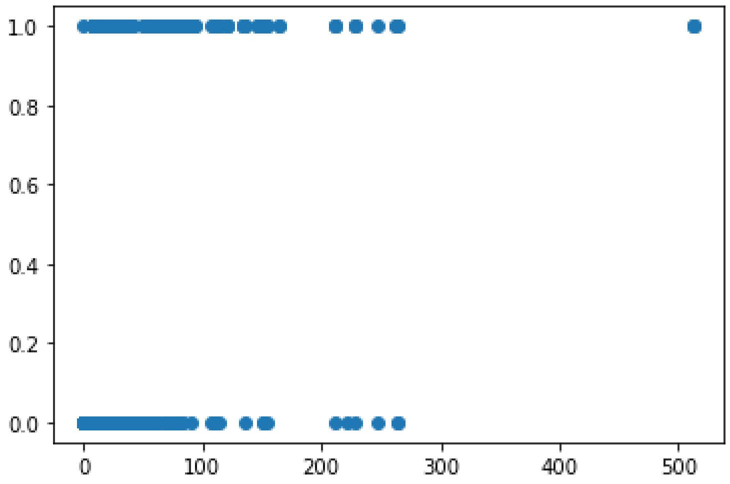
In [20]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(df['SibSp'], df['Survived'])
```

Out[20]:
```
<matplotlib.collections.PathCollection at 0x21309acd490>
```



In [22]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(df['Fare'], df['Survived'])
```

Out[22]:
```
<matplotlib.collections.PathCollection at 0x21309b8ae20>
```



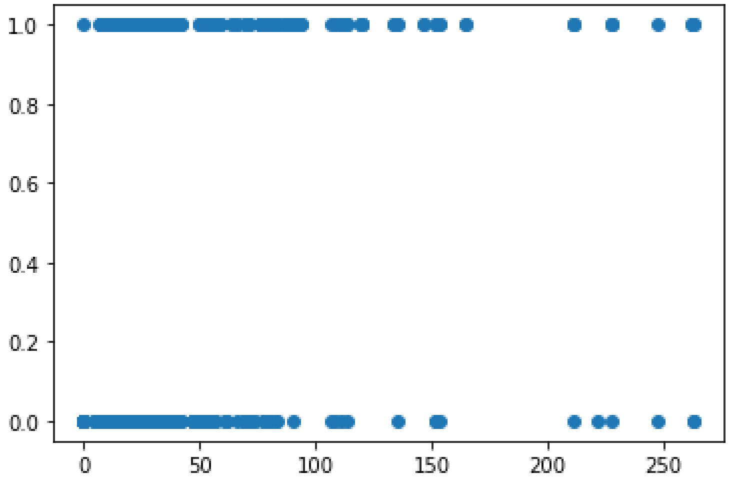
In [23]:
```
mode_fare = df['Fare'].mode()[0]
for x in range(0,891):
    if df['Fare'][x] >= 400:
        df['Fare'][x] = mode_fare
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_10164\690323287.py:4: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df['Fare'][x] = mode_fare
```

In [24]:
```
import matplotlib.pyplot as plt
%matplotlib inline
plt.scatter(df['Fare'], df['Survived'])
```

Out[24]:
```
<matplotlib.collections.PathCollection at 0x21309c0a220>
```



In [25]:
```
x = df.drop(['Survived'], axis=1)
y = df['Survived']
```

In [26]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2,random_stat
```

In [27]:
```
from sklearn.preprocessing import MinMaxScaler
min_max_scaler = MinMaxScaler()
```

In [28]:
```
x_train = min_max_scaler.fit_transform(x_train)
x_test = min_max_scaler.fit_transform(x_test)
```

In [29]:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

In [30]:

```
model.fit(x_train,y_train)
```

Out[30]:

```
LogisticRegression()
```

In [31]:

```
y_pred = model.predict(x_test)
y_pred
```

Out[31]:

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

In [33]:

```
from sklearn.metrics import classification_report
classification_report(y_test, y_pred)
```

Out[33]:

```
'              precision    recall  f1-score   support\n\n           0       0.
81      0.90      0.85       105\n           1       0.83      0.70      0.76
74\n\n    accuracy                           0.82       179\n   macro avg
0.82      0.80      0.80       179\nweighted avg       0.82      0.82      0.81
179\n'
```