

# 7130hcxip

January 23, 2025

```
[67]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

```
[51]: # Load data
data = pd.read_csv("C:/Users/91703/OneDrive/Desktop/TITANIC.csv")
```

```
[52]: # to get top rows
data.head()
```

```
[52]:
```

	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

```
[4]: data.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

```
[5]: data.describe()
```

```
[5]:
```

	PassengerId	Survived	Pclass	Age	SibSp \
count	891.000000	891.000000	891.000000	714.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008
std	257.353842	0.486592	0.836071	14.526497	1.102743
min	1.000000	0.000000	1.000000	0.420000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000
50%	446.000000	0.000000	3.000000	28.000000	0.000000
75%	668.500000	1.000000	3.000000	38.000000	1.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000

  

	Parch	Fare
count	891.000000	891.000000
mean	0.381594	32.204208
std	0.806057	49.693429
min	0.000000	0.000000
25%	0.000000	7.910400
50%	0.000000	14.454200
75%	0.000000	31.000000
max	6.000000	512.329200

```
[6]: data.isnull().sum()
```

```
[6]: 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

```

```
[54]: data.dropna(subset=['Age', 'Cabin', 'Embarked'], inplace=True)
data['Sex'] = data['Sex'].map({'male': 0, 'female': 1})
```

```
[48]: data.isna().sum()
```

```
[48]: 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
```

```
[55]: data.head()
```

```
[55]:   PassengerId  Survived  Pclass  \
1             2         1       1
3             4         1       1
6             7         0       1
10            11         1       3
11            12         1       1
```

```

                                Name  Sex  Age  SibSp  \
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  1  38.0    1
3      Futrelle, Mrs. Jacques Heath (Lily May Peel)  1  35.0    1
6                        McCarthy, Mr. Timothy J    0  54.0    0
10                     Sandstrom, Miss. Marguerite Rut  1   4.0    1
11                     Bonnell, Miss. Elizabeth    1  58.0    0

```

```

    Parch  Ticket      Fare Cabin Embarked
1       0  PC 17599  71.2833   C85        C

```

3	0	113803	53.1000	C123	S
6	0	17463	51.8625	E46	S
10	1	PP 9549	16.7000	G6	S
11	0	113783	26.5500	C103	S

```
[56]: #define feautrs(X) and(Y)
X = data[['Pclass', 'Sex', 'SibSp', 'Parch', 'Fare']]
y = data['Age']
```

```
[57]: # Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↳random_state=42)
```

```
[58]: # Create and train Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
[58]: LinearRegression()
```

```
[60]: # Make predictions on test data
y_pred = model.predict(X_test)
print(y_pred)
```

```
[36.26568185 14.04105756 25.18876351 39.63473538 40.36820553 42.48201036
41.88167937 37.36735174 36.54658099 42.53799533 42.5487767 36.2493076
42.47061176 32.83164241 36.64798812 38.74758715 36.07248098 42.53125697
23.07026221 35.83231516 32.98424847 37.11179956 42.59476598 25.43063341
38.76617288 21.66156806 30.26293587 42.24824598 29.15332158 36.76625143
38.24603348 32.33142612 42.40828196 38.51281581 37.53327863 38.7930196
35.30140095]
```

```
[62]: #print the model coefficients and Intercepts
print("Coefficients:", model.coef_)
print("Intercept:", model.intercept_)
```

```
Coefficients: [-9.88514267 -3.76906535 -1.88004045 -3.34758627 -0.01347671]
Intercept: 52.834177769504436
```

```
[61]: # Evaluate model performance
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 199.49587487464927
```

```
[63]: error = y_test - y_pred
print(error)
```

```
118    -12.265682
251     14.958942
742     -4.188764
544     10.365265
712      7.631794
96      28.517990
139    -17.881679
337      3.632648
577      2.453419
492     12.462005
487     15.451223
765     14.750692
339      2.529388
835      6.168358
262     15.352012
97     -15.747587
291    -17.072481
209     -2.531257
429      8.929738
307    -18.832315
779     10.015752
609      2.888200
707     -0.594766
183    -24.430633
54      26.233827
618    -17.661568
318      0.737064
110      4.751754
717     -2.153322
556     11.233749
369    -14.246033
305    -31.411426
583     -6.408282
710    -14.512816
332      0.466721
177     11.206980
853    -19.301401
Name: Age, dtype: float64
```

```
[66]: sns.regplot(x=y_pred, y=error, data=data)
```

```
[66]: <Axes: ylabel='Age'>
```

