In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
df2=pd.read_csv(r'C:\Users\user\Downloads\C2_test.gender_submission.csv')
df2
fin2=pd.read_csv(r'C:\Users\user\Downloads\C2_train.gender_submission.csv')
fin2
```

Out[2]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Far
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.250
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.283
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050
...	...	...	...	...	...	...	...	...	...	.
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.450
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750

891 rows × 12 columns

In [3]:

```
df0=df2.head(10)
df=df0.fillna("30.0")
df
fin0=fin2.head(10)
fin=fin0.fillna("35.0")
fin
```

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	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	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	35.0	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708

In [4]:

```
df.info()
fin.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  10 non-null     int64  
 1   Pclass       10 non-null     int64  
 2   Name         10 non-null     object 
 3   Sex          10 non-null     object 
 4   Age          10 non-null     float64
 5   SibSp        10 non-null     int64  
 6   Parch        10 non-null     int64  
 7   Ticket       10 non-null     object 
 8   Fare         10 non-null     float64
 9   Cabin        10 non-null     object 
 10  Embarked     10 non-null     object 
dtypes: float64(2), int64(4), object(5)
memory usage: 1008.0+ bytes
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  10 non-null     int64  
 1   Survived     10 non-null     int64  
 2   Pclass       10 non-null     int64  
 3   Name         10 non-null     object 
 4   Sex          10 non-null     object 
 5   Age          10 non-null     object 
 6   SibSp        10 non-null     int64  
 7   Parch        10 non-null     int64  
 8   Ticket       10 non-null     object 
 9   Fare         10 non-null     float64
 10  Cabin        10 non-null     object 
 11  Embarked     10 non-null     object 
dtypes: float64(1), int64(5), object(6)
memory usage: 1.1+ KB
```

In [5]:

```
df.describe()
fin.describe()
```

Out[5]:

	PassengerId	Survived	Pclass	SibSp	Parch	Fare
count	10.00000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	5.50000	0.500000	2.300000	0.700000	0.300000	27.020820
std	3.02765	0.527046	0.948683	0.948683	0.674949	23.601938
min	1.00000	0.000000	1.000000	0.000000	0.000000	7.250000
25%	3.25000	0.000000	1.250000	0.000000	0.000000	8.152075
50%	5.50000	0.500000	3.000000	0.500000	0.000000	16.104150
75%	7.75000	1.000000	3.000000	1.000000	0.000000	46.414575
max	10.00000	1.000000	3.000000	3.000000	2.000000	71.283300

In [6]:

```
df.columns
fin.columns
```

Out[6]:

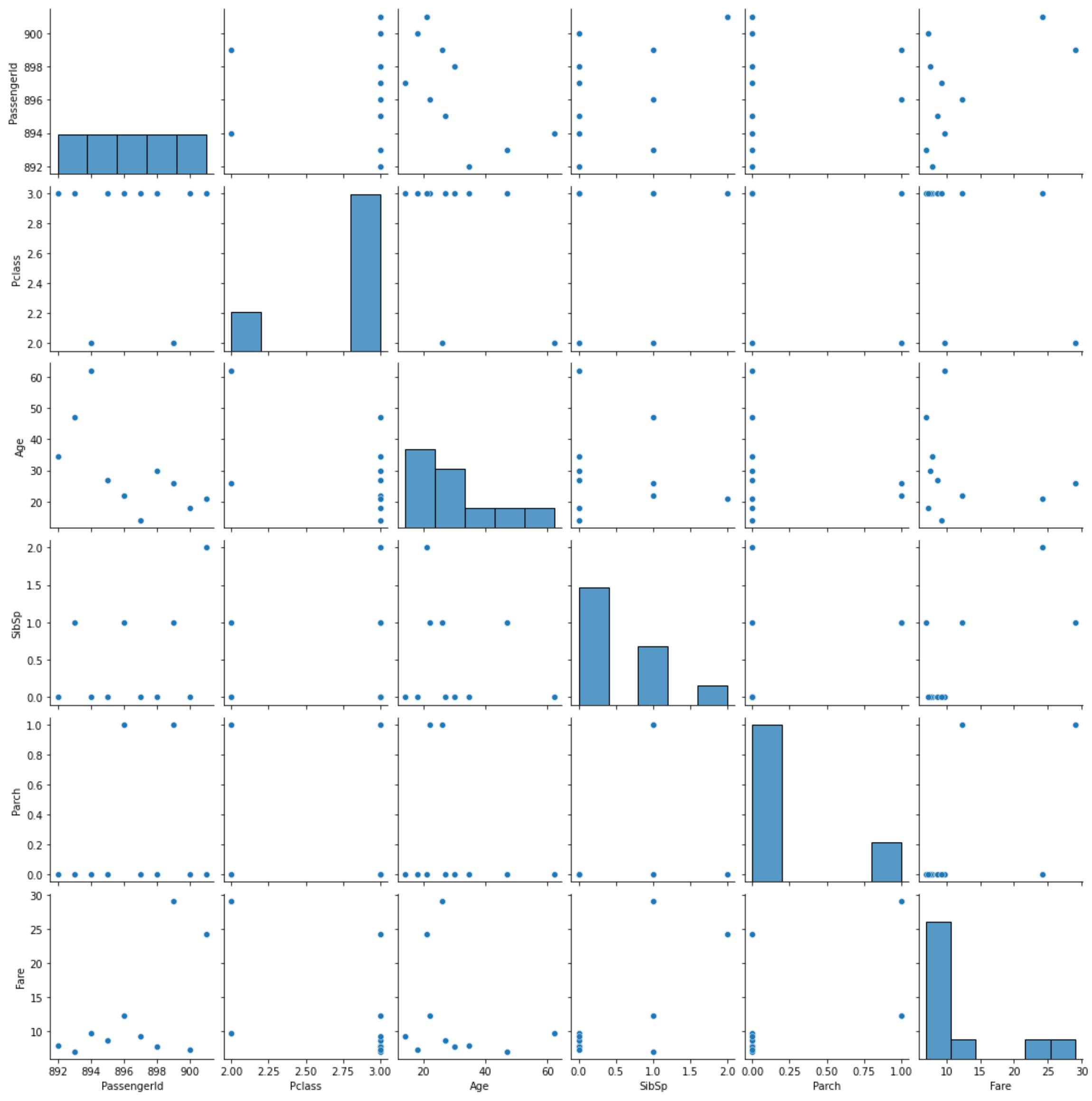
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

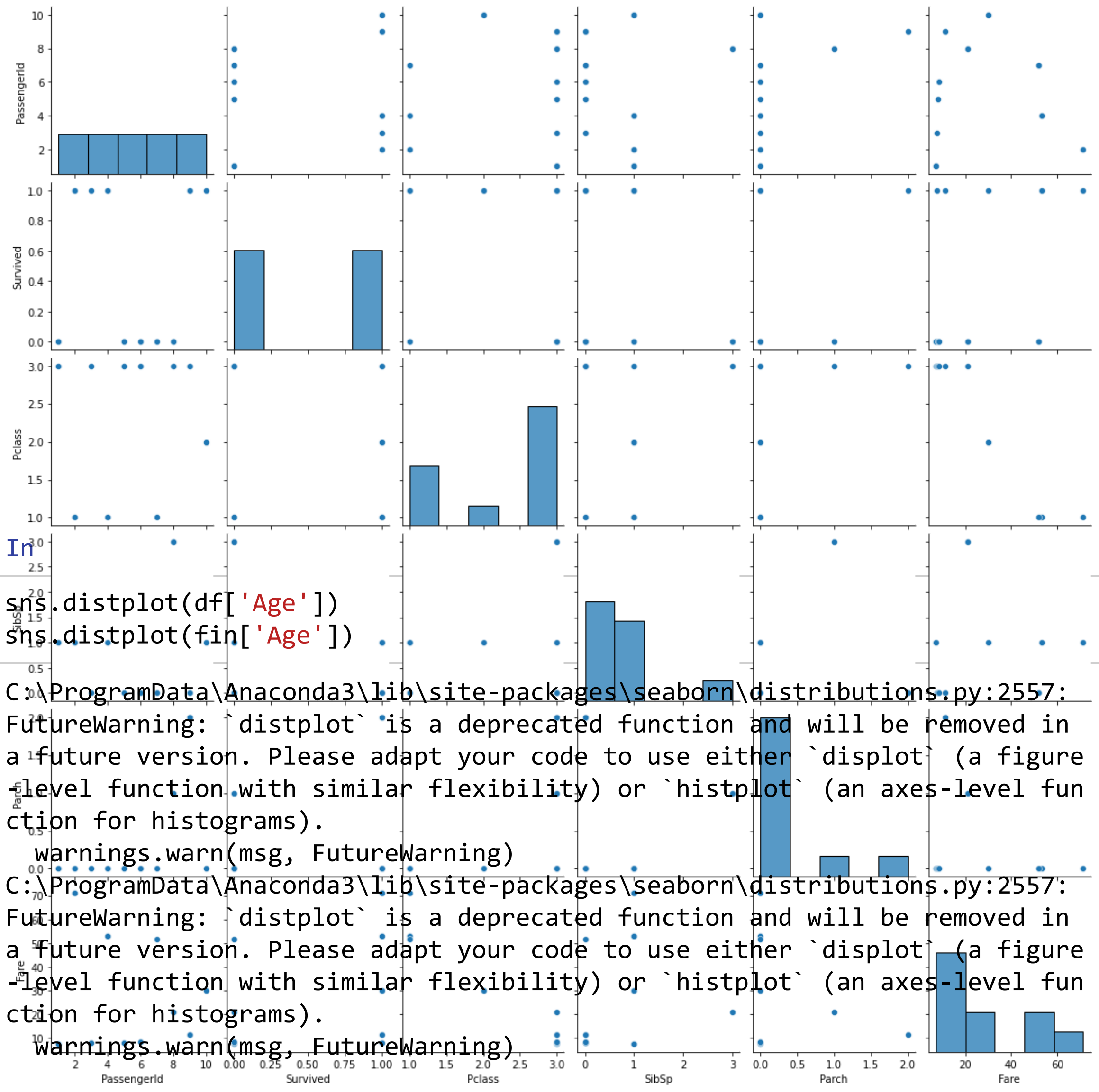
In [7]:

```
sns.pairplot(df)
sns.pairplot(fin)
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x2918b32d340>
```





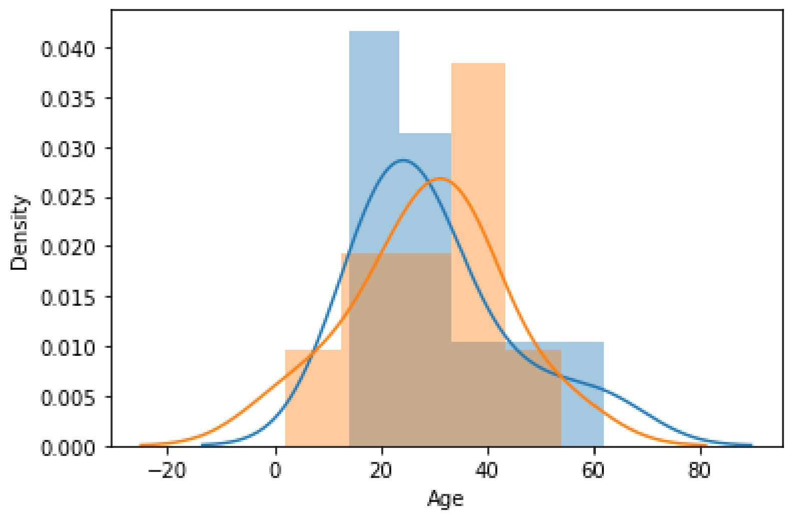
In

```
sns.distplot(df['Age'])
sns.distplot(fin['Age'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[8]:

```
<AxesSubplot:xlabel='Age', ylabel='Density'>
```

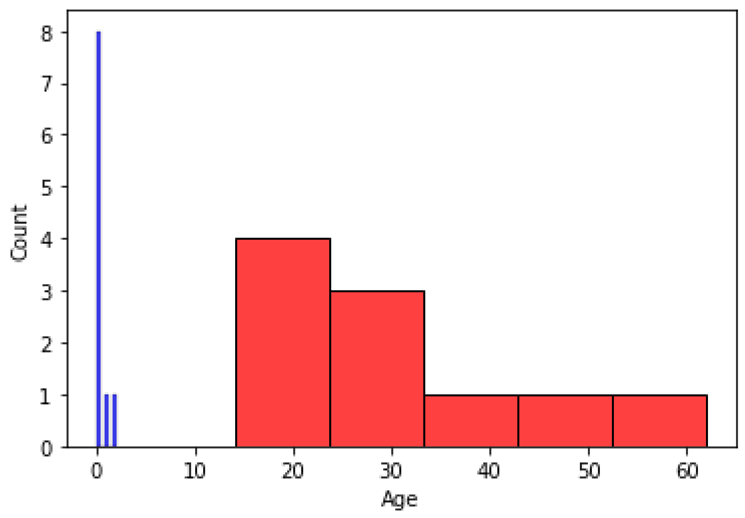


In [9]:

```
sns.histplot(df["Age"],color='red')
sns.histplot(fin["Parch"],color='blue')
```

Out[9]:

```
<AxesSubplot:xlabel='Age', ylabel='Count'>
```



In [10]:

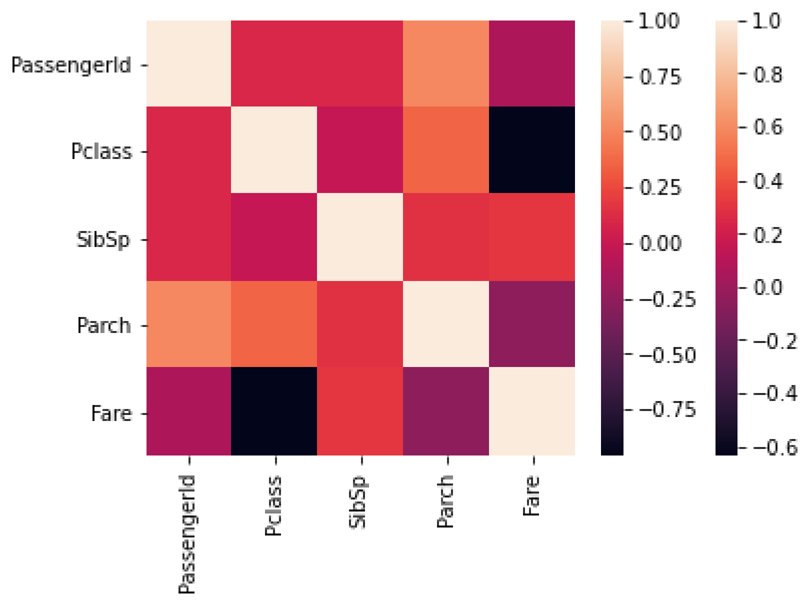
```
df1=df[['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked']]
fin1=fin[['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked']]
```

In [11]:

```
sns.heatmap(df1.corr())
sns.heatmap(fin1.corr())
```

Out[11]:

```
<AxesSubplot:>
```



In [12]:

```
x=df1[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch','Fare']]
y=fin1[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch','Fare']]
```

In [13]:

```
from sklearn.model_selection import train_test_split
```

In [14]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [15]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)#ValueError: Input contains NaN, infinity or a value too large for
```

Out[15]:

```
LinearRegression()
```

In [16]:

```
print(lr.intercept_)
```

```
[ -891.           332.87835864 -4954.83029001   138.61697769
     0.         -6934.40841297]
```

In [17]:

```
coef= pd.DataFrame(lr.coef_)
coef
```

Out[17]:

	0	1	2	3	4	5
0	1.000000	-4.866403e-16	6.924589e-16	5.515729e-16	0.0	3.279364e-15
1	-0.354184	-3.868372e+00	-9.726923e-02	2.681135e-01	0.0	5.464786e-02
2	5.532481	1.657792e+01	5.239007e-01	1.407403e-01	0.0	-3.184044e+00
3	-0.161799	1.715021e+00	2.249348e-02	-2.257961e-01	0.0	1.245994e-01
4	0.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.0	0.000000e+00
5	7.516158	7.225946e+01	1.859503e+00	1.470512e+01	0.0	-3.811048e+00

In [18]:

```
print(lr.score(x_test,y_test))
```

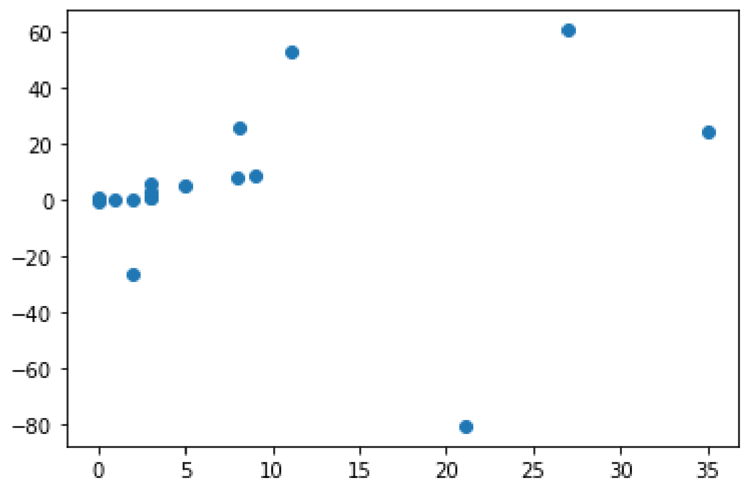
```
-22.52209907188785
```

In [19]:

```
prediction = lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[19]:

```
<matplotlib.collections.PathCollection at 0x2918bbb62b0>
```



In [20]:

```
lr.score(x_test,y_test)
```

Out[20]:

```
-22.52209907188785
```

In [21]:

```
lr.score(x_train,y_train)
```

Out[21]:

```
0.8916852718919287
```

In [22]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [23]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[23]:

```
Ridge(alpha=10)
```

In [24]:

```
rr.score(x_test,y_test)
```

Out[24]:

```
-3.383452576892554
```

In [25]:

```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 0.0, tolerance:
0.0
  model = cd_fast.enet_coordinate_descent(
```

Out[25]:

```
Lasso(alpha=10)
```

In [26]:

```
la.score(x_test,y_test)
```

Out[26]:

```
-2.207084556972902
```

Elastic Net

In [27]:

```
from sklearn.linear_model import ElasticNet
en = ElasticNet()
en.fit(x_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_coordinat
e_descent.py:530: ConvergenceWarning: Objective did not converge. You migh
t want to increase the number of iterations. Duality gap: 0.0, tolerance:
0.0
  model = cd_fast.enet_coordinate_descent(
```

Out[27]:

```
ElasticNet()
```

In [28]:

```
print(en.coef_)
```

```
[[ 7.38992951e-01 -0.00000000e+00 -1.95680928e-02  0.00000000e+00
   0.00000000e+00  6.77545964e-02]
 [-0.00000000e+00 -0.00000000e+00  2.08752798e-03 -0.00000000e+00
   0.00000000e+00  1.61910530e-03]
 [ 3.68143731e+00  7.97420625e-02  6.51140915e-02  5.10797984e-01
   0.00000000e+00 -2.78083821e+00]
 [-0.00000000e+00  0.00000000e+00 -1.77696945e-05  0.00000000e+00
   0.00000000e+00  7.83565915e-03]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00
   0.00000000e+00  0.00000000e+00]
 [ 3.42014506e+00  5.01721588e+00  2.76294345e-01  9.34754689e+00
   0.00000000e+00 -2.71379332e+00]]
```

In [29]:

```
print(en.intercept_)
```

```
[-6.57271969e+02  1.91261092e+00 -3.23868308e+03  4.88986989e-01
  0.00000000e+00 -3.02949935e+03]
```

In [30]:

```
prediction=en.predict(x_test)
print(prediction)
```

```
[[ 5.26775162  1.97843129 27.89774242  0.58487672  0.         32.08255313]
 [ 8.5388068   2.0138407  -7.35198985  0.71575909  0.         -6.92332105]
 [ 7.95927272  1.96189126 55.91855122  0.54531268  0.         49.0375898
3]]
```

In [31]:

```
print(en.score(x_test,y_test))
```

```
-5.080616467467732
```

Logistic Regression

In [32]:

```
from sklearn.linear_model import LogisticRegression
```

In [33]:

```
feature_matrix=df1[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch','Fare']]
target_vector = df1[['Embarked']]
```

In [34]:

```
feature_matrix.shape
```

Out[34]:

```
(10, 6)
```

In [35]:

```
target_vector.shape
```

Out[35]:

```
(10, 1)
```

In [36]:

```
from sklearn.preprocessing import StandardScaler
```

In [37]:

```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [38]:

```
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63:
DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using r
avel().
  return f(*args, **kwargs)
```

Out[38]:

```
LogisticRegression()
```

In [41]:

```
observation=df1[['PassengerId', 'Pclass', 'Age', 'SibSp', 'Parch','Fare']]
```

In [42]:

```
prediction = logr.predict(observation)
print(prediction)
```

```
['C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C' 'C']
```

In [43]:

```
logr.classes_
```

Out[43]:

```
array(['C', 'Q', 'S'], dtype=object)
```

In [44]:

```
logr.predict_proba(observation)[0][0]
```

Out[44]:

```
1.0
```

In [45]:

```
logr.predict_proba(observation)[0][1]
```

Out[45]:

```
0.0
```

Evaluation Metrics

In [46]:

```
from sklearn import metrics
```

In [47]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,prediction))
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-47-4b3ebc69c71d> in <module>
----> 1 print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,pr
ediction))

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64 
     65             # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py
in mean_absolute_error(y_true, y_pred, sample_weight, multioutput)
    180     0.85...
    181     """
--> 182     y_type, y_true, y_pred, multioutput = _check_reg_targets(
    183         y_true, y_pred, multioutput)
    184     check_consistent_length(y_true, y_pred, sample_weight)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py
in _check_reg_targets(y_true, y_pred, multioutput, dtype)
     86         the dtype argument passed to check_array.
     87     """
---> 88     check_consistent_length(y_true, y_pred)
     89     y_true = check_array(y_true, ensure_2d=False, dtype=dtype)
     90     y_pred = check_array(y_pred, ensure_2d=False, dtype=dtype)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_consistent_length(*arrays)
    260     uniques = np.unique(lengths)
    261     if len(uniques) > 1:
--> 262         raise ValueError("Found input variables with inconsistent
numbers of"
    263                          " samples: %r" % [int(l) for l in length
s])
    264 

ValueError: Found input variables with inconsistent numbers of samples:
[3, 10]
```

In [48]:

```
print("Mean Squared Error:",metrics.mean_squared_error(y_test,prediction))
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-48-a8cbb0b7f78a> in <module>
----> 1 print("Mean Squared Error:",metrics.mean_squared_error(y_test,pred
iction))

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64 
     65             # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py
in mean_squared_error(y_true, y_pred, sample_weight, multioutput, squared)
    333     0.825...
    334     """
--> 335     y_type, y_true, y_pred, multioutput = _check_reg_targets(
    336         y_true, y_pred, multioutput)
    337     check_consistent_length(y_true, y_pred, sample_weight)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py
in _check_reg_targets(y_true, y_pred, multioutput, dtype)
     86         the dtype argument passed to check_array.
     87     """
---> 88     check_consistent_length(y_true, y_pred)
     89     y_true = check_array(y_true, ensure_2d=False, dtype=dtype)
     90     y_pred = check_array(y_pred, ensure_2d=False, dtype=dtype)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_consistent_length(*arrays)
    260     uniques = np.unique(lengths)
    261     if len(uniques) > 1:
--> 262         raise ValueError("Found input variables with inconsistent
numbers of"
    263                          " samples: %r" % [int(l) for l in length
s])
    264 

ValueError: Found input variables with inconsistent numbers of samples:
[3, 10]
```

In [49]:

```
print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_error(y_test,prediction)))
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-49-c32fd676945a> in <module>
----> 1 print("Root Mean Squared Error:",np.sqrt(metrics.mean_squared_erro
r(y_test,prediction)))

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64 
     65             # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py
in mean_squared_error(y_true, y_pred, sample_weight, multioutput, squared)
    333     0.825...
    334     """
--> 335     y_type, y_true, y_pred, multioutput = _check_reg_targets(
    336         y_true, y_pred, multioutput)
    337     check_consistent_length(y_true, y_pred, sample_weight)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py
in _check_reg_targets(y_true, y_pred, multioutput, dtype)
     86         the dtype argument passed to check_array.
     87     """
---> 88     check_consistent_length(y_true, y_pred)
     89     y_true = check_array(y_true, ensure_2d=False, dtype=dtype)
     90     y_pred = check_array(y_pred, ensure_2d=False, dtype=dtype)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_consistent_length(*arrays)
    260     uniques = np.unique(lengths)
    261     if len(uniques) > 1:
--> 262         raise ValueError("Found input variables with inconsistent
numbers of"
    263                          " samples: %r" % [int(l) for l in length
s])
    264 

ValueError: Found input variables with inconsistent numbers of samples:
[3, 10]
```

Model Saving

In [50]:

```
import pickle
```

In [51]:

```
filename="prediction"
pickle.dump(lr,open(filename,'wb'))
```