

```
In [54]: # Importing Pandas and NumPy
import pandas as pd
import numpy as np

In [55]: #importing dataset
titanic = pd.read_csv(r"C:\Users\Ananth Kumar\Desktop\Stats\nrippner-titanic-disaster

In [56]:
```

Out[56]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	bc
0	1.0	1.0	Allen, Miss. Elisabeth Walton	female	29.0000	0.0	0.0	24160	211.3375	B5	S	2	N
1	1.0	1.0	Allison, Master. Hudson Trevor	male	0.9167	1.0	2.0	113781	151.5500	C22 C26	S	11	N
2	1.0	0.0	Allison, Miss. Helen Loraine	female	2.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	N
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	13
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	N
...
1305	3.0	0.0	Zabour, Miss. Thamine	female	NaN	1.0	0.0	2665	14.4542	NaN	C	NaN	N
1306	3.0	0.0	Zakarian, Mr. Mapriededer	male	26.5000	0.0	0.0	2656	7.2250	NaN	C	NaN	30
1307	3.0	0.0	Zakarian, Mr. Ortin	male	27.0000	0.0	0.0	2670	7.2250	NaN	C	NaN	N
1308	3.0	0.0	Zimmerman, Mr. Leo	male	29.0000	0.0	0.0	315082	7.8750	NaN	S	NaN	N
1309	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

1310 rows × 14 columns

In [57]:

Out[57]:

	pclass	survived	age	sibsp	parch	fare	body
count	1309.000000	1309.000000	1046.000000	1309.000000	1309.000000	1308.000000	121.000000
mean	2.294882	0.381971	29.881135	0.498854	0.385027	33.295479	160.809917
std	0.837836	0.486055	14.413500	1.041658	0.865560	51.758668	97.696922
min	1.000000	0.000000	0.166700	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	21.000000	0.000000	0.000000	7.895800	72.000000
50%	3.000000	0.000000	28.000000	0.000000	0.000000	14.454200	155.000000
75%	3.000000	1.000000	39.000000	1.000000	0.000000	31.275000	256.000000
max	3.000000	1.000000	80.000000	8.000000	9.000000	512.329200	328.000000

In [58]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1310 entries, 0 to 1309
Data columns (total 14 columns):
pclass      1309 non-null float64
survived     1309 non-null float64
name        1309 non-null object
sex         1309 non-null object
age         1046 non-null float64
sibsp       1309 non-null float64
parch       1309 non-null float64
ticket      1309 non-null object
fare        1308 non-null float64
cabin       295 non-null object
embarked    1307 non-null object
boat        486 non-null object
body        121 non-null float64
home.dest   745 non-null object
dtypes: float64(7), object(7)
memory usage: 143.4+ KB
```

In []:

```
In [59]: # Creating a dummy variable for the variable 'sex' and dropping the first one.
gen = pd.get_dummies(titanic['sex'],prefix='sex')
#Adding the results to the master dataframe
```

```
In [60]: #dropping sex column
```

In [68]:

Out[68]:

	pclass	survived	name	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body
0	1.0	1.0	Allen, Miss. Elisabeth Walton	29.000000	0.0	0.0	24160	211.3375	B5	S	2	NaN
1	1.0	1.0	Allison, Master. Hudson Trevor	0.916700	1.0	2.0	113781	151.5500	C22 C26	S	11	NaN
2	1.0	0.0	Allison, Miss. Helen Loraine	2.000000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	30.000000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	135.0
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	25.000000	1.0	2.0	113781	151.5500	C22 C26	S	NaN	NaN
...
1304	3.0	0.0	Zabour, Miss. Hileni	14.500000	1.0	0.0	2665	14.4542	NaN	C	NaN	328.0
1305	3.0	0.0	Zabour, Miss. Thamine	29.881135	1.0	0.0	2665	14.4542	NaN	C	NaN	NaN
1306	3.0	0.0	Zakarian, Mr. Mapriededer	26.500000	0.0	0.0	2656	7.2250	NaN	C	NaN	304.0
1307	3.0	0.0	Zakarian, Mr. Ortin	27.000000	0.0	0.0	2670	7.2250	NaN	C	NaN	NaN
1308	3.0	0.0	Zimmerman, Mr. Leo	29.000000	0.0	0.0	315082	7.8750	NaN	S	NaN	NaN

1308 rows × 15 columns

In [62]:

```

-----
KeyError                                Traceback (most recent call last)
C:\Anaconda\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2896         try:
-> 2897             return self._engine.get_loc(key)
    2898         except KeyError:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'sex'

During handling of the above exception, another exception occurred:

KeyError                                Traceback (most recent call last)
<ipython-input-62-a500312ca371> in <module>
----> 1 titanic['sex'].unique()

C:\Anaconda\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
    2978         if self.columns.nlevels > 1:
    2979             return self._getitem_multilevel(key)
-> 2980         indexer = self.columns.get_loc(key)
    2981         if is_integer(indexer):
    2982             indexer = [indexer]

C:\Anaconda\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
    2897             return self._engine.get_loc(key)
    2898         except KeyError:
-> 2899             return self._engine.get_loc(self._maybe_cast_indexer(key))
    2900         indexer = self.get_indexer([key], method=method, tolerance=tolerance)
    2901         if indexer.ndim > 1 or indexer.size > 1:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'sex'

```

```
In [63]:
```

```
Out[63]: pclass      1
survived      1
name          1
age          264
sibsp         1
parch         1
ticket        1
fare          2
cabin        1015
embarked      3
boat          824
body         1189
home.dest     565
sex_female    0
sex_male      0
dtype: int64
```

```
In [64]:
```

```
Out[64]: pclass      0.08
survived      0.08
name          0.08
age          20.15
sibsp         0.08
parch         0.08
ticket        0.08
fare          0.15
cabin        77.48
embarked      0.23
boat          62.90
body          90.76
home.dest     43.13
sex_female    0.00
sex_male      0.00
dtype: float64
```

```
In [65]: #20% of age got null values - so filling missing values with mean value
```

```
In [66]: #0.08% null values there for pclass, so removing the rows.(which will remove all other rows)
titanic = titanic[~np.isnan(titanic['pclass'])]
```

```
In [67]: #0.08% null values there for fare, so removing the rows
```

```
In [69]: #0.08% null values there for embarked, dummification of embarked (coz it has uniquely
emb = pd.get_dummies (titanic['embarked'],prefix='embarked',drop_first=True)
#Adding the results to the master dataframe
```

```
In [70]: #dropping embarked column
```

```
In [71]: #body column has 90% null values so dropping the entire column
```

```
In [72]: #dropping columns which have more than 50% null values i.e, cabin(77.45%) and boat (62.90%)
```

```
In [73]: #home.dest has 43.04% null values filling it up with mode since it is object dtype
#renaming
titanic.rename(columns={'home.dest':'home_destination'},
```

In [74]:

In [75]:

Out[75]:

	pclass	survived	name	age	sibsp	parch	ticket	fare	home_destination	sex_female	s
0	1.0	1.0	Allen, Miss. Elisabeth Walton	29.000000	0.0	0.0	24160	211.3375	St Louis, MO	1	
1	1.0	1.0	Allison, Master. Hudson Trevor	0.916700	1.0	2.0	113781	151.5500	Montreal, PQ / Chesterville, ON	0	
2	1.0	0.0	Allison, Miss. Helen Loraine	2.000000	1.0	2.0	113781	151.5500	Montreal, PQ / Chesterville, ON	1	
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	30.000000	1.0	2.0	113781	151.5500	Montreal, PQ / Chesterville, ON	0	
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	25.000000	1.0	2.0	113781	151.5500	Montreal, PQ / Chesterville, ON	1	
...
1304	3.0	0.0	Zabour, Miss. Hileni	14.500000	1.0	0.0	2665	14.4542	New York, NY	1	
1305	3.0	0.0	Zabour, Miss. Thamine	29.881135	1.0	0.0	2665	14.4542	New York, NY	1	
1306	3.0	0.0	Zakarian, Mr. Mapriededer	26.500000	0.0	0.0	2656	7.2250	New York, NY	0	
1307	3.0	0.0	Zakarian, Mr. Ortin	27.000000	0.0	0.0	2670	7.2250	New York, NY	0	
1308	3.0	0.0	Zimmerman, Mr. Leo	29.000000	0.0	0.0	315082	7.8750	New York, NY	0	

1308 rows × 13 columns

In [76]: *#now there is no null values*

Out[76]: pclass 0.0
survived 0.0
name 0.0
age 0.0
sibsp 0.0
parch 0.0
ticket 0.0
fare 0.0
home_destination 0.0
sex_female 0.0
sex_male 0.0
embarked_Q 0.0
embarked_S 0.0
dtype: float64

In [77]:

```
Out[77]: pclass      0
survived      0
name          0
age           0
sibsp         0
parch         0
ticket        0
fare          0
home_destination  0
sex_female    0
sex_male      0
embarked_Q    0
embarked_S    0
dtype: int64
```

In [78]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1308 entries, 0 to 1308
Data columns (total 13 columns):
pclass      1308 non-null float64
survived     1308 non-null float64
name        1308 non-null object
age         1308 non-null float64
sibsp       1308 non-null float64
parch       1308 non-null float64
ticket      1308 non-null object
fare        1308 non-null float64
home_destination  1308 non-null object
sex_female  1308 non-null uint8
sex_male    1308 non-null uint8
embarked_Q  1308 non-null uint8
embarked_S  1308 non-null uint8
dtypes: float64(6), object(3), uint8(4)
memory usage: 107.3+ KB
```

In [79]:

Out[79]:

	pclass	survived	age	sibsp	parch	fare	sex_female	sex_male
count	1308.000000	1308.000000	1308.000000	1308.000000	1308.000000	1308.000000	1308.000000	1308.000000
mean	2.294343	0.382263	29.857726	0.499235	0.385321	33.295479	0.356269	0.643731
std	0.837929	0.486126	12.860247	1.041965	0.865826	51.758668	0.479079	0.479079
min	1.000000	0.000000	0.166700	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	22.000000	0.000000	0.000000	7.895800	0.000000	0.000000
50%	3.000000	0.000000	29.881135	0.000000	0.000000	14.454200	0.000000	1.000000
75%	3.000000	1.000000	35.000000	1.000000	0.000000	31.275000	1.000000	1.000000
max	3.000000	1.000000	80.000000	8.000000	9.000000	512.329200	1.000000	1.000000

In []:

In [80]:

```
In [82]: num.describe()
```

Out[82]:

	age	fare
count	1308.000000	1308.000000
mean	29.857726	33.295479
std	12.860247	51.758668
min	0.166700	0.000000
25%	22.000000	7.895800
50%	29.881135	14.454200
75%	35.000000	31.275000
90%	48.000000	78.050820
95%	54.650000	133.650000
99%	64.000000	262.375000
max	80.000000	512.329200

```
In [90]: num[num['age'] > 40].describe()
```

```
In [92]: num[num['age'] > 40].describe().round(2)
```

```
In [97]: num
```

Out[97]:

	fare1
0	211.3375
1	151.5500
2	151.5500
3	151.5500
4	151.5500
...	...
1304	14.4542
1305	14.4542
1306	7.2250
1307	7.2250
1308	7.8750

1304 rows × 1 columns

```
In [96]: num.rename(columns={'fare':'fare1'},
```

C:\Anaconda\lib\site-packages\pandas\core\frame.py:4223: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
return super().rename(**kwargs)

```
In [98]: num[num['age'] > 40].describe()
```


In [100]:

In [101]:

Out[101]:

	pclass	survived	name	age	sibsp	parch	ticket	home_destination	sex_female	sex_male	embarked
0	1.0	1.0	Allen, Miss. Elisabeth Walton	29.0000	0.0	0.0	24160	St Louis, MO	1	0	
1	1.0	1.0	Allison, Master. Hudson Trevor	0.9167	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	0	1	
2	1.0	0.0	Allison, Miss. Helen Loraine	2.0000	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	1	0	
3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	30.0000	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	0	1	
4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	25.0000	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	1	0	

In [102]:

```
Out[102]: pclass      0
survived      0
name          0
age           0
sibsp         0
parch         0
ticket        0
home_destination  0
sex_female    0
sex_male      0
embarked_Q    0
embarked_S    0
fael          4
dtype: int64
```

In [103]:

In [104]:

```
Out[104]: pclass          0
survived          0
name              0
age              0
sibsp            0
parch            0
ticket           0
home_destination 0
sex_female        0
sex_male          0
embarked_Q        0
embarked_S        0
fare1            0
dtype: int64
```

In [106]:

Out[106]:

	pclass	survived	name	age	sibsp	parch	ticket	home_destination	sex_female	sex_male	
	0	1.0	1.0	Allen, Miss. Elisabeth Walton	29.000000	0.0	0.0	24160	St Louis, MO	1	0
	1	1.0	1.0	Allison, Master. Hudson Trevor	0.916700	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	0	1
	2	1.0	0.0	Allison, Miss. Helen Loraine	2.000000	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	1	0
	3	1.0	0.0	Allison, Mr. Hudson Joshua Creighton	30.000000	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	0	1
	4	1.0	0.0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	25.000000	1.0	2.0	113781	Montreal, PQ / Chesterville, ON	1	0
...
1304	3.0	0.0	Zabour, Miss. Hileni	14.500000	1.0	0.0	2665	New York, NY	1	0	
1305	3.0	0.0	Zabour, Miss. Thamine	29.881135	1.0	0.0	2665	New York, NY	1	0	
1306	3.0	0.0	Zakarian, Mr. Mapriededer	26.500000	0.0	0.0	2656	New York, NY	0	1	
1307	3.0	0.0	Zakarian, Mr. Ortin	27.000000	0.0	0.0	2670	New York, NY	0	1	
1308	3.0	0.0	Zimmerman, Mr. Leo	29.000000	0.0	0.0	315082	New York, NY	0	1	

1304 rows × 13 columns

Model Building

In [108]: `### Splitting Data into Training and Test Sets`

In [109]: `# Putting feature variable to X
#axis means column
x= titanic.drop(['survived','home_destination','name','ticket'],axis=1)

Putting response variable to y`

In [110]:

Out[110]:

	pclass	age	sibsp	parch	sex_female	sex_male	embarked_Q	embarked_S	fare1
0	1.0	29.0000	0.0	0.0	1	0	0	1	211.3375
1	1.0	0.9167	1.0	2.0	0	1	0	1	151.5500
2	1.0	2.0000	1.0	2.0	1	0	0	1	151.5500
3	1.0	30.0000	1.0	2.0	0	1	0	1	151.5500
4	1.0	25.0000	1.0	2.0	1	0	0	1	151.5500

In [111]:

Out[111]:

0	1.0
1	1.0
2	0.0
3	0.0
4	0.0

Name: survived, dtype: float64

In [112]: `# Splitting the data into train and test`

In [113]: `from sklearn import metrics
logsk = LogisticRegression()`

C:\Anaconda\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence thi
s warning.
FutureWarning)

Out[113]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='warn', n_jobs=None, penalty='l2',
random_state=None, solver='warn', tol=0.0001, verbose=0,
warm_start=False)

In [141]:

In [142]:

C:\Anaconda\lib\site-packages\sklearn\linear_model\sag.py:337: ConvergenceWarning:
The max_iter was reached which means the coef_ did not converge
"the coef_ did not converge", ConvergenceWarning)

Out[142]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=1, max_iter=100,
multi_class='warn', n_jobs=None, penalty='elasticnet',
random_state=None, solver='saga', tol=0.0001, verbose=0,
warm_start=False)

In [143]:

In [144]:

Out[144]: 68.11224489795919

In [118]:

```
# Prediction
```

In [119]:

```
Out[119]: array([[1., 0., 0., 1., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0.,
0., 0., 0., 0., 0., 1., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0.,
0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 0., 1., 1., 0.,
0., 1., 0., 1., 0., 0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 1.,
0., 0., 1., 0., 0., 1., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 1., 0., 1., 0., 0., 1., 0., 1., 0., 0., 0., 1., 0., 1., 1., 0.,
0., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 1.,
0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0.,
0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 0.,
1., 0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0.,
1., 1., 1., 0., 0., 1., 0., 0., 1., 1., 1., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 1., 0., 1., 0., 0., 0., 0., 0., 0., 1., 0., 1., 0., 0.,
0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 1., 1., 0.,
1., 1., 1., 1., 0., 0., 1., 1., 0., 1., 1., 1., 1., 0., 1., 0., 0.,
1., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 1., 1., 0.,
1., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1., 1., 0., 0.,
0., 0., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 0., 1., 1., 0., 0.,
0.]])
```

In [125]:

```
from sklearn import metrics
```

Out[125]: 77.8061224489796

using l2 penalty and c=1

In [132]:

In [133]:

```
C:\Anaconda\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning:
Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence thi
s warning.
    FutureWarning)
```

Out[133]:

```
LogisticRegression(C=1, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='warn', n_jobs=None, penalty='l1',
random_state=None, solver='warn', tol=0.0001, verbose=0,
warm_start=False)
```

In [134]:

In [149]:

Out[149]: 78.06122448979592

#Evaluation metrics

```
In [157]: # Predicted probabilities
```

```
In [158]: # Converting y_pred to a dataframe which is an array
```

```
In [159]: # Converting to column dataframe
```

```
In [160]: # Let's see the head
```

```
Out[160]:
```

	1
0	0.930689
1	0.201310
2	0.485978
3	0.620718
4	0.076645

```
In [161]: # Converting y_test to dataframe
```

```
In [162]: # Putting name to index
```

```
In [163]: # Removing index for both dataframes to append them side by side  
y_pred_1.reset_index(drop=True, inplace=True)
```

```
In [164]: # Appending y_test_df and y_pred_1
```

```
In [165]: # Renaming the column
```

```
In [169]: # Rearranging the columns
```

```
Out[169]:
```

	survived	name	Survived_Prob
0	1.0	144	0.930689
1	0.0	154	0.201310
2	0.0	244	0.485978
3	1.0	529	0.620718
4	0.0	691	0.076645

```
In [171]: # Creating new column 'predicted' with 1 if Churn_Prob>0.5 else 0  
y_pred_final['predicted'] = y_pred_final.Survived_Prob.map( lambda x: 1 if x > 0.5 else 0)
```

In [172]: `# Let's see the head`

Out[172]:

	survived	name	Survived_Prob	predicted
0	1.0	144	0.930689	1
1	0.0	154	0.201310	0
2	0.0	244	0.485978	0
3	1.0	529	0.620718	1
4	0.0	691	0.076645	0

`#confusion matrix`

In [174]:

In [175]: `# Confusion matrix`

`confusion = metrics.confusion_matrix(y_pred_final.survived, y_pred_final.predicted)`

Out[175]: `array([[210, 27],
[59, 96]], dtype=int64)`

In [176]: `#Let's check the overall accuracy.`

Out[176]: 0.7806122448979592

In [177]: `TP = confusion[0,0] # true positive
TN = confusion[1,1] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives`

In [180]: `#let's check precision`

In [182]:

Out[182]: 0.8860759493670886

In [181]: `#let's check recall - sensitivity`

In [183]:

Out[183]: 0.7806691449814126

In [185]: `#let's check recall - f1 score`

In [186]:

Out[186]: 0.8300395256916996

In [187]: `TPR = Recall`

In [188]:

Out[188]: 0.7806691449814126

In [189]:

Out[189]: 0.21951219512195122

In [190]:

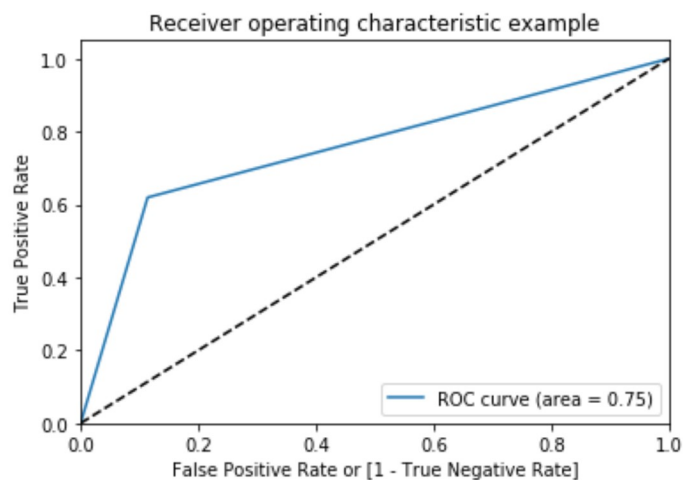
In [191]:

Out[191]: 1.2195121951219512

ROC Curve

```
In [193]: def draw_roc( actual, probs ):
            fpr, tpr, thresholds = metrics.roc_curve( actual, probs,
                                                    drop_intermediate = False )
            auc_score = metrics.roc_auc_score( actual, probs )
            plt.figure(figsize=(6, 4))
            plt.plot( fpr, tpr, label='ROC curve (area = %0.2f)' % auc_score )
            plt.plot([0, 1], [0, 1], 'k--')
            plt.xlim([0.0, 1.0])
            plt.ylim([0.0, 1.05])
            plt.xlabel('False Positive Rate or [1 - True Negative Rate]')
            plt.ylabel('True Positive Rate')
            plt.title('Receiver operating characteristic example')
            plt.legend(loc="lower right")
            plt.show()
```

In [198]: import matplotlib.pyplot as plt



Out[198]: (array([0. , 0.11392405, 1.]),
array([0. , 0.61935484, 1.]),
array([2, 1, 0], dtype=int64))

In []: