ML MODELS PROBE FOR TITANIC DISASTER PROBLEM

Made By Santiago Cadena A.

```
[notice] A new release of pip is available: 23.2.1 -> 24.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: pandoc in c:\users\santi\appdata\local\programs\python\py
thon312\lib\site-packages (2.3)
Requirement already satisfied: plumbum in c:\users\santi\appdata\local\programs\python\p
ython312\lib\site-packages (from pandoc) (1.8.3)
Requirement already satisfied: ply in c:\users\santi\appdata\local\programs\python\pytho
n312\lib\site-packages (from pandoc) (3.11)
Requirement already satisfied: pywin32 in c:\users\santi\appdata\local\programs\python\p
ython312\lib\site-packages (from plumbum->pandoc) (306)
```

Work stages find the solution:

1. Prepare, clean the data.
2. Identify patterns: correlation beween variables, analyze the data
3. Model and predict the problem
4. Visualize, report and present the problem solving ## Workflow goals
5. Classifying : Understand the implication between the classes.
6. Correlating: Find the features that contribute better than others.
7. Converting: Text to data
8. Correcting: Detect outliers, discard features
9. Creating: Create new features
10. Charging: Choose correct visualization charts for analyze them

Out[3]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarke
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Out[4]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

```
trained dataframe: Index(['Age', 'Embarked'], dtype='object')
test dataframe: Index(['Age', 'Fare'], dtype='object')
```

Analysis by single features (Pclass, Embarkation type, Sex)

Out[6]:

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

Out[7]:

	Embarked	Survived
0	C	0.553571
1	Q	0.389610
2	S	0.336957

Out[8]:

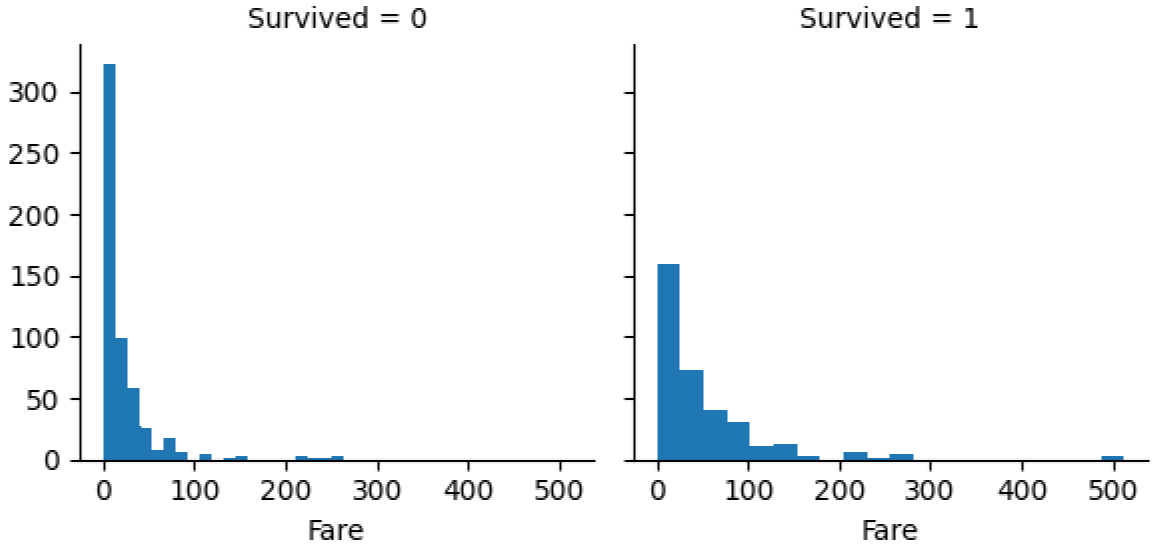
	Sex	Survived
0	female	0.742038
1	male	0.188908

- Sex ~ Survival
- PClass ~ Survival
- Embarkation ~ Survival

Analysis by visualizating features (age,fare)

Out[9]:

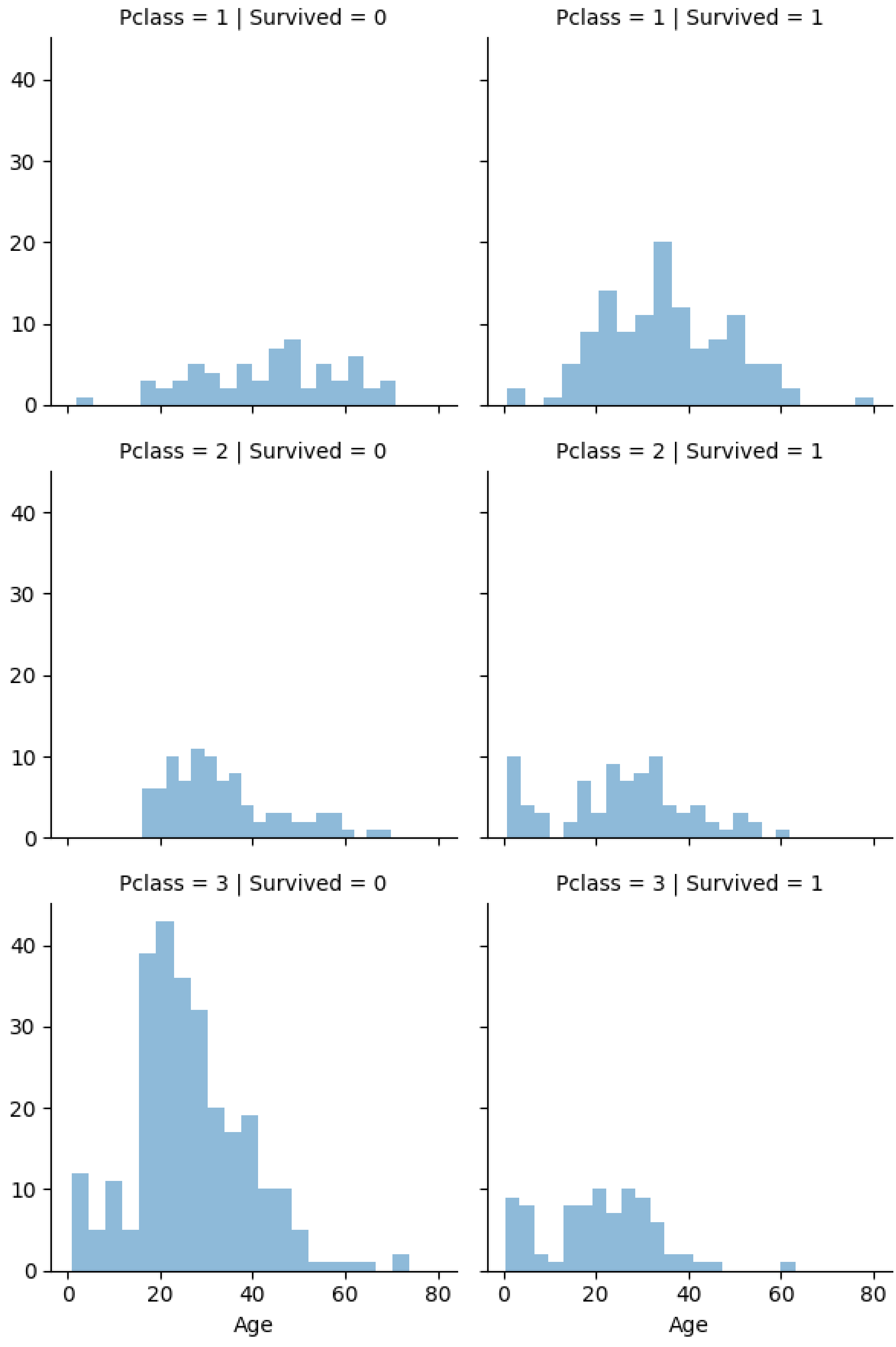
```
<seaborn.axisgrid.FacetGrid at 0x1dc27e17da0>
```



Similar behaviour for survived and unsurvived, no correlation.

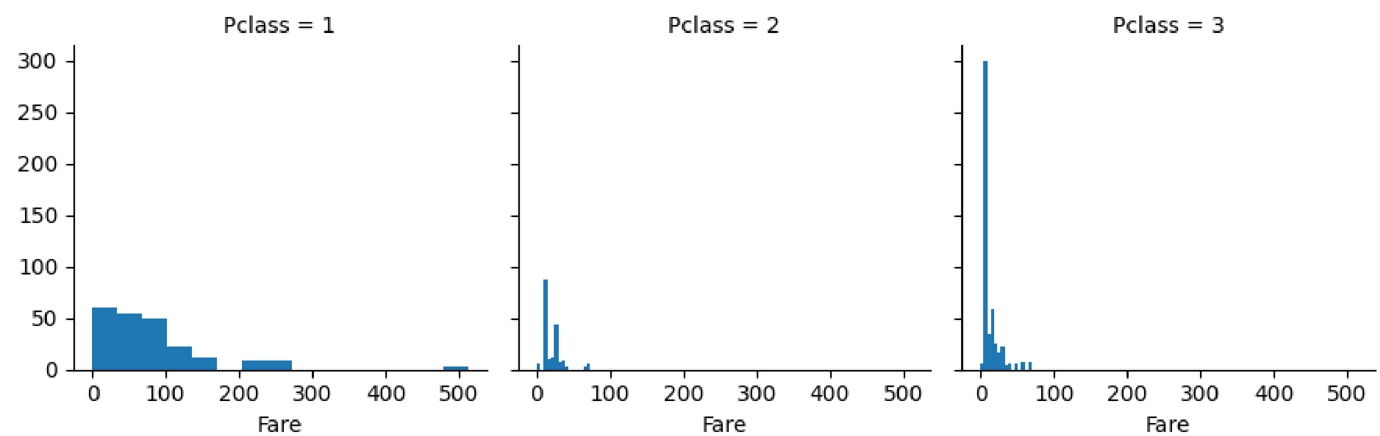
Out[10]:

```
<seaborn.axisgrid.FacetGrid at 0x1dc29f66de0>
```



- The lower the class, most probability of survival
- People of approximate 25 are the most probable to survive
- In class 2 and 1 most of the people who didn´t surive were from 20 to 30

Out[11]: `<seaborn.axisgrid.FacetGrid at 0x1dc27e16b40>`



We see a left-shifted distribution, hence it's appropiate use a metric like the median and not the mean for filling the Nan values in Fare

Filling up Data

The Name feature do contributte, due to the title 'Mr','Miss','Mrs' could be of utility in the survival classification. Also can be of interest in filling Fare, Age NaN values.

```
<>:2: SyntaxWarning: invalid escape sequence '\.'
<>:2: SyntaxWarning: invalid escape sequence '\.'
C:\Users\santi\AppData\Local\Temp\ipykernel_10036\2013435651.py:2: SyntaxWarning: invali
d escape sequence '\.'
  dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.', expand=False)
```

Out[13]:

Sex	female	male
Title		
Capt	0	1
Col	0	2
Countess	1	0
Don	0	1
Dr	1	6
Jonkheer	0	1
Lady	1	0
Major	0	2
Master	0	40
Miss	182	0
Mlle	2	0
Mme	1	0
Mr	0	517
Mrs	125	0
Ms	1	0
Rev	0	6
Sir	0	1

```
   Survived  Pclass                                                   Name  \
```

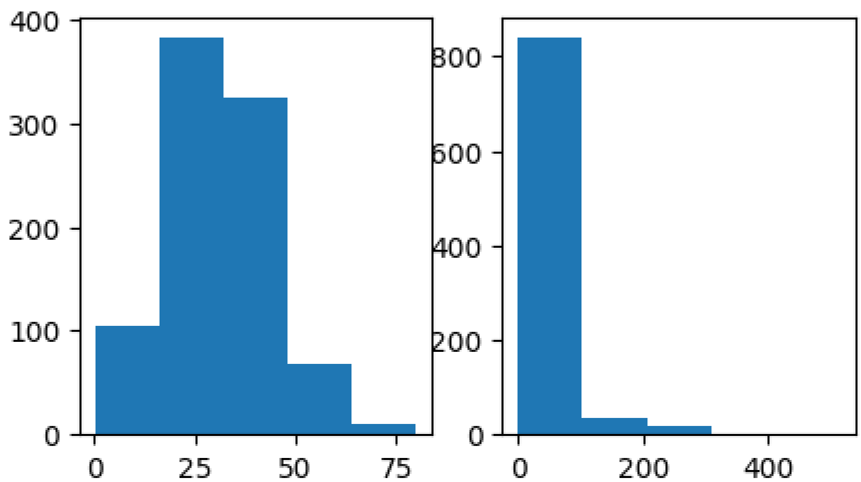
```
0         0       3                          Braund, Mr. Owen Harris
1         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th...
2         1       3                           Heikkinen, Miss. Laina
3         1       1     Futrelle, Mrs. Jacques Heath (Lily May Peel)
4         0       3                         Allen, Mr. William Henry

      Sex   Age  SibSp  Parch     Fare Embarked Title
0    male  22.0      1      0   7.2500        S    Mr
1  female  38.0      1      0  71.2833        C   Mrs
2  female  26.0      0      0   7.9250        S  Miss
3  female  35.0      1      0  53.1000        S   Mrs
4    male  35.0      0      0   8.0500        S    Mr
```

Out[14]:

	Title	Survived
4	Mrs	0.795276
2	Miss	0.702703
1	Master	0.575000
3	Mr	0.163842
0	Anyone	0.125000

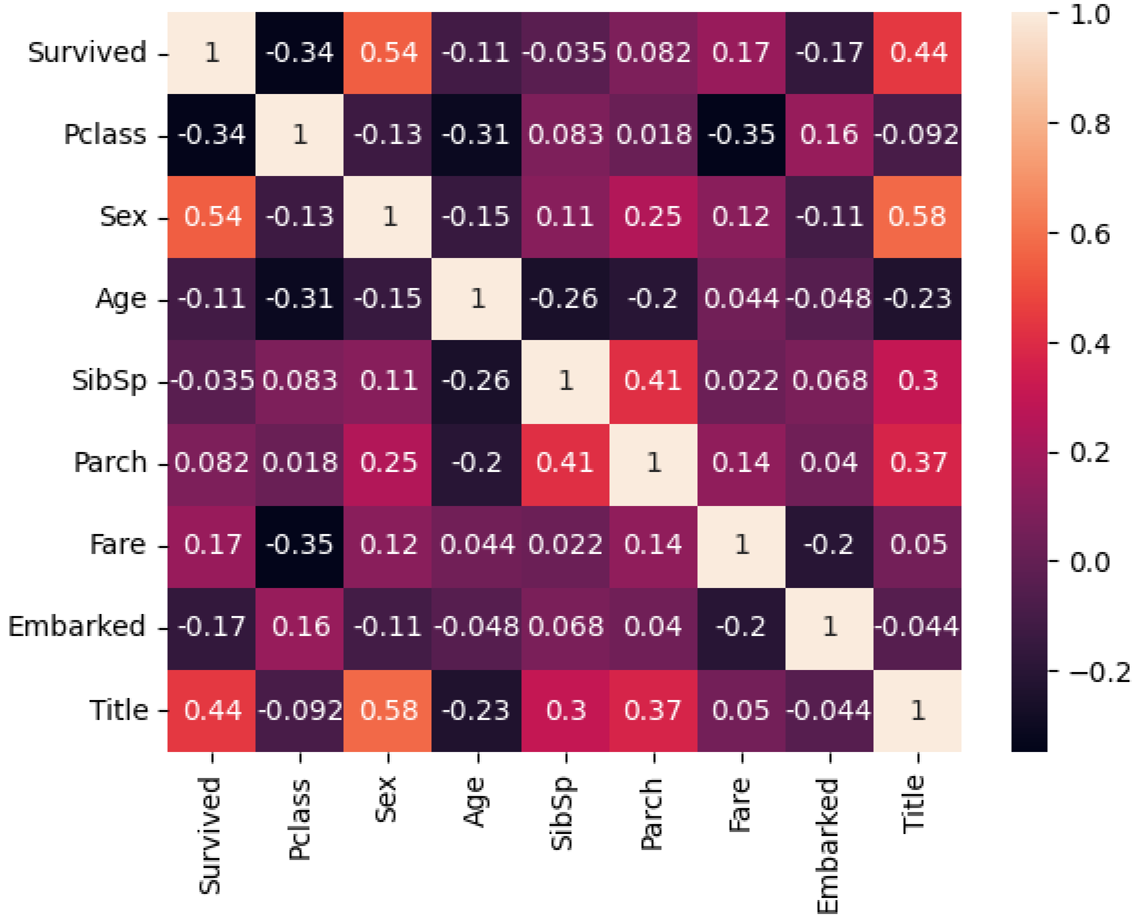
Fill the Nan ages based on the 'Title' aggrupation

convert cathegorical to numerical features, then see the heatmap for more correlation if exist



Out[17]:

```
<Axes: >
```

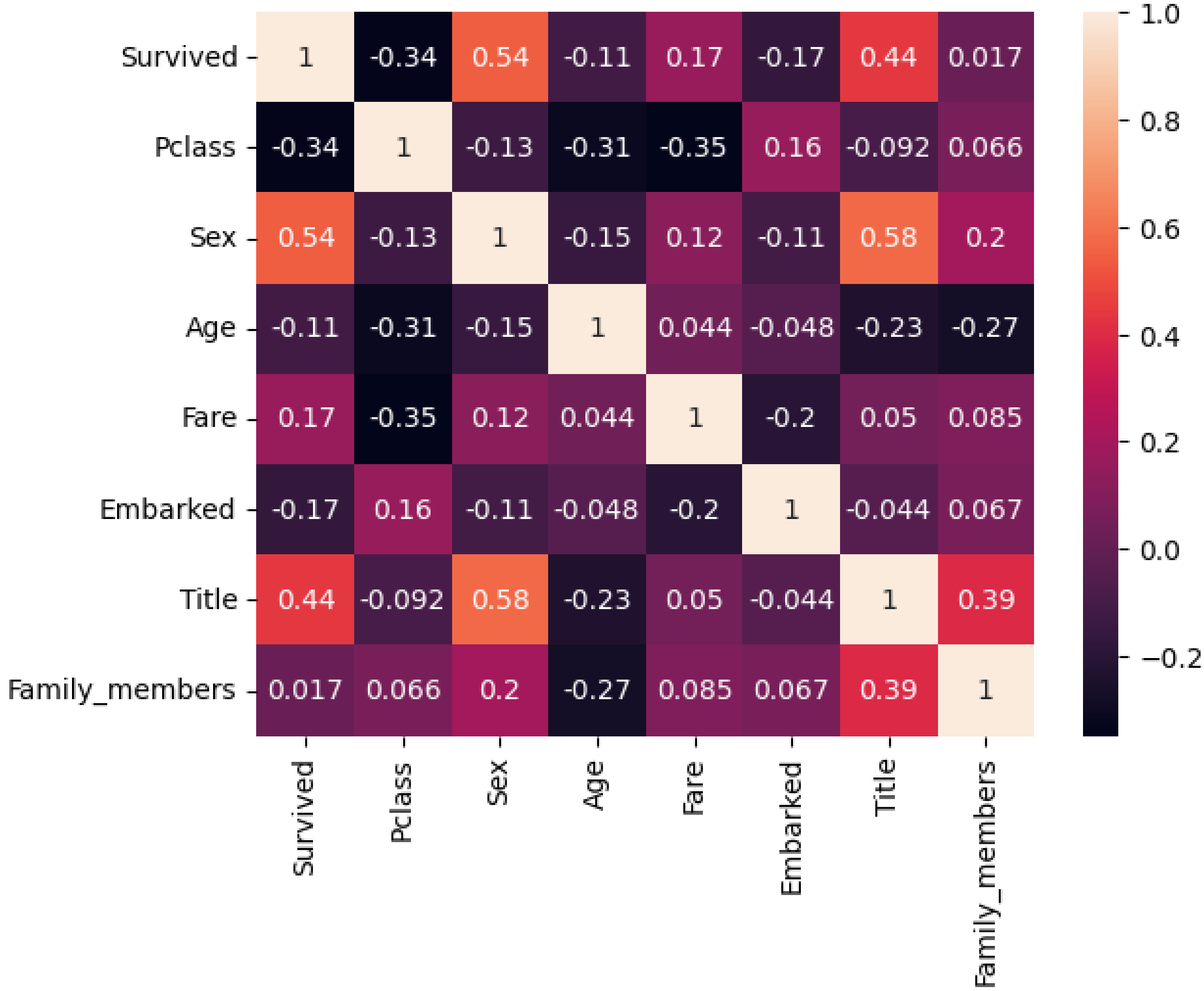


'SibSp' is correlated with 'Parch' significally, and they don't weight too much to 'Survived'. Hence sum them.

```
   Survived  Pclass  Sex Age Fare  Embarked  Title  Family_members
0         0       3    0   2    1         2      1               1
1         1       1    1   3    1         0      3               1
2         1       3    1   2    1         2      2               0
3         1       1    1   3    1         2      3               1
4         0       3    0   3    1         2      1               0
```

```
C:\Users\santi\AppData\Local\Temp\ipykernel_10036\199106421.py:2: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  members= df[['SibSp','Parch']].apply(lambda row: row[0] + row[1], axis=1)
C:\Users\santi\AppData\Local\Temp\ipykernel_10036\199106421.py:2: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  members= df[['SibSp','Parch']].apply(lambda row: row[0] + row[1], axis=1)
```

Out[18]: <Axes: >



Modelation

For mixed features, i.e cathegorical and numerical these models work: Decision Trees, Naive Bayes (with Gaussian distribution with numeric attributes),KNN. Otherwise it's necessary to try ensemble techniques.

```
Test set score for SVC model: 0.6388
Confusion matrix for SVC model:
[[266   0]
 [151   1]]
--------------------------------------------------
Test set score for LinearSVC model: 0.6053
Confusion matrix for LinearSVC model:
[[229  37]
 [128  24]]
--------------------------------------------------
Test set score for DecisionTreeClassifier model: 0.7560
Confusion matrix for DecisionTreeClassifier model:
[[198  68]
 [ 34 118]]
--------------------------------------------------
Test set score for Perceptron model: 0.5981
Confusion matrix for Perceptron model:
[[236  30]
 [138  14]]
--------------------------------------------------
Test set score for SGDClassifier model: 0.6388
Confusion matrix for SGDClassifier model:
[[266   0]
 [151   1]]
```

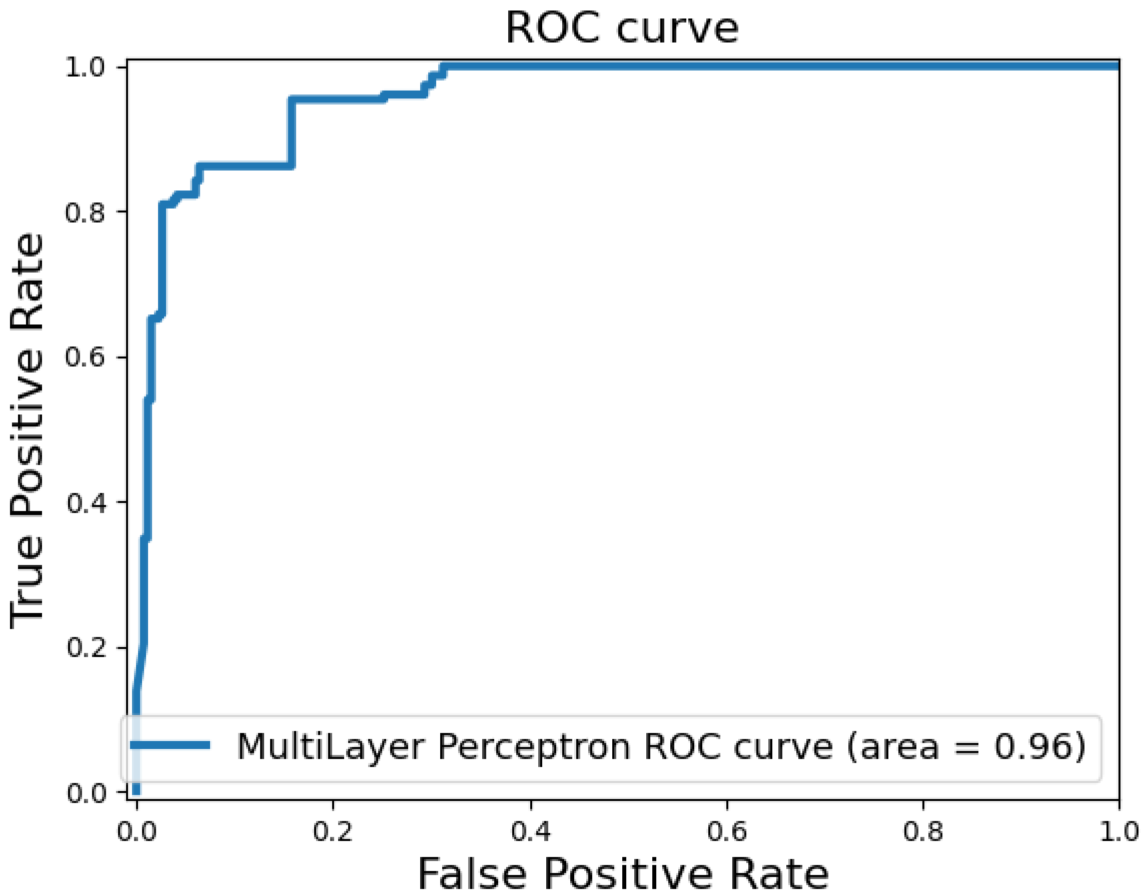
```
--------------------------------------------------
Test set score for GaussianNB model: 0.3636
Confusion matrix for GaussianNB model:
[[  0 266]
 [  0 152]]
--------------------------------------------------
Test set score for KNeighborsClassifier model: 0.5239
Confusion matrix for KNeighborsClassifier model:
[[168  98]
 [101  51]]
--------------------------------------------------
Test set score for RandomForestClassifier model: 0.9091
Confusion matrix for RandomForestClassifier model:
[[234  32]
 [  6 146]]
--------------------------------------------------
Test set score for MLPClassifier model: 0.5861
Confusion matrix for MLPClassifier model:
[[214  52]
 [121  31]]
--------------------------------------------------
Test set score for LogisticRegression model: 0.6029
Confusion matrix for LogisticRegression model:
[[239  27]
 [139  13]]
--------------------------------------------------
```

```
c:\Users\santi\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\neural_
network\_multilayer_perceptron.py:690: ConvergenceWarning: Stochastic Optimizer: Maximum
iterations (200) reached and the optimization hasn't converged yet.
  warnings.warn(
```

The Random Forest and Decision Tree Clasifier with default parameters are the best fits founded. However it's not yet ready, must be superior to 95%.

```
0.8971291866028708 [200, 'sqrt', 'gini']
```

Know that I know the best parameters, let's see the validation accuracy or AUC and ROC metrics to know if it's overfitted.


ROC curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
MultiLayer Perceptron ROC curve (area = 0.96)