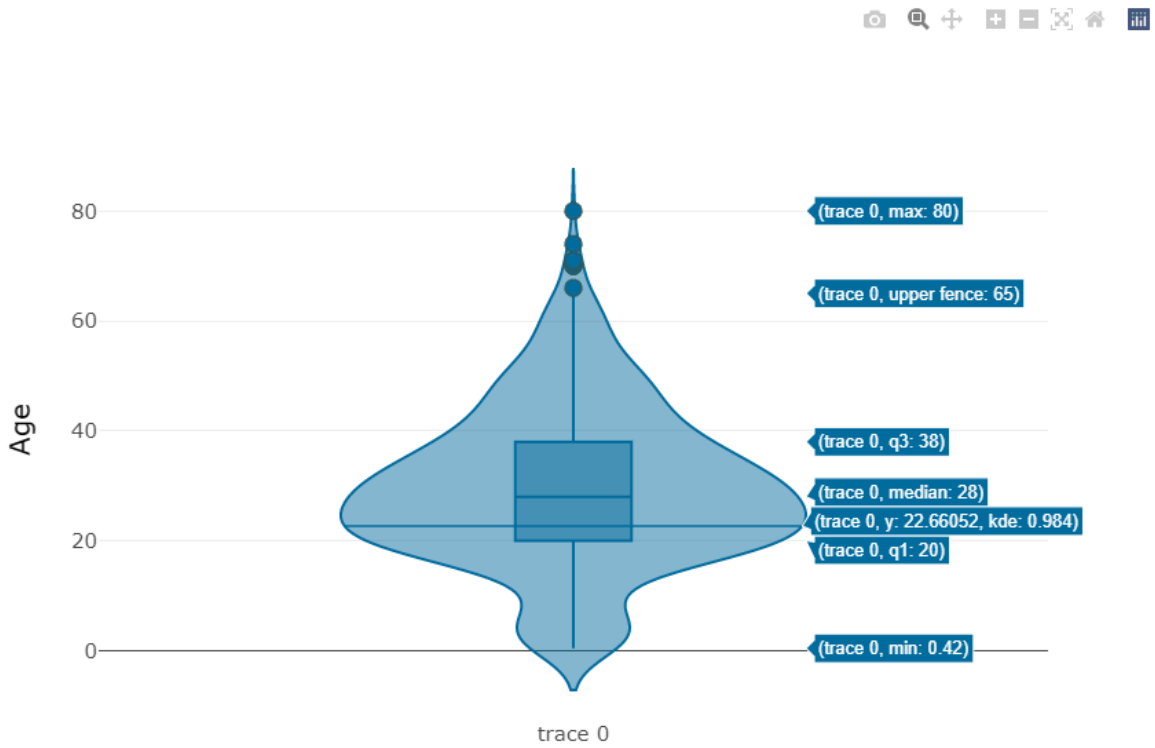
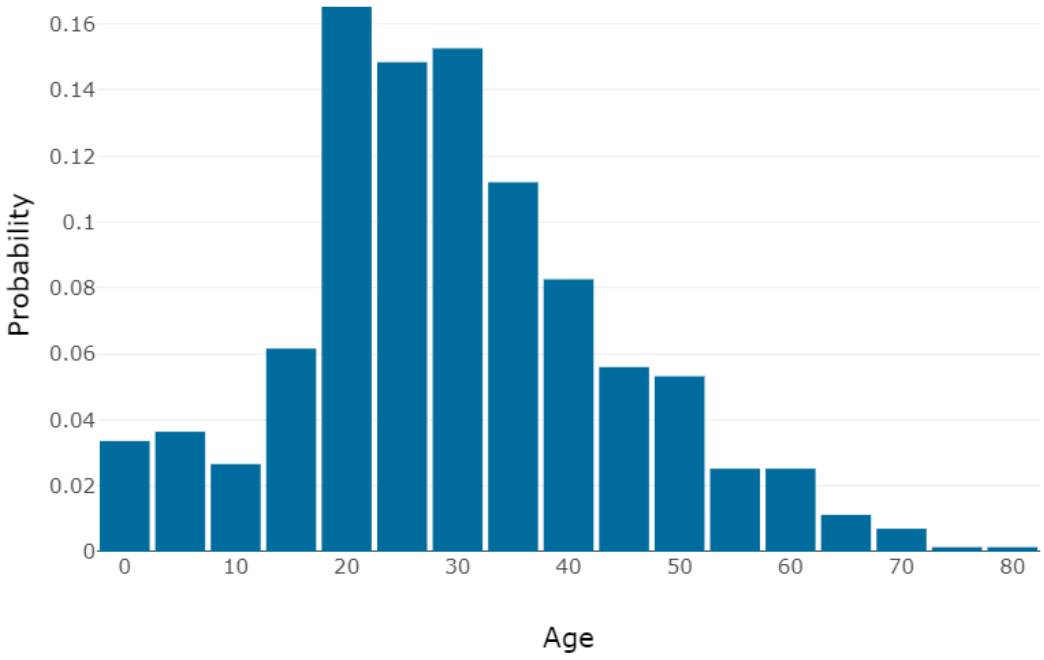
Short data summary statistics

	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	

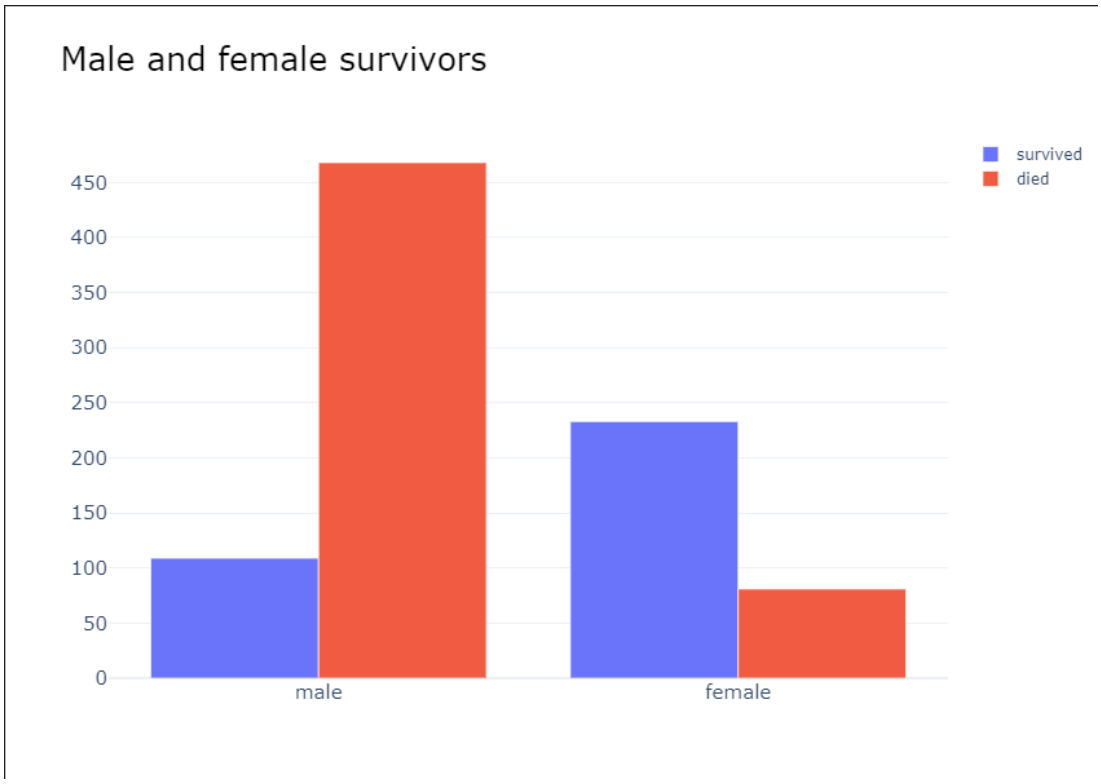
	Parch	Fare
count	891.000000	891.000000
mean	0.381594	32.204208
std	0.806057	49.693429
min	0.000000	0.000000
25%	0.000000	7.910400
50%	0.000000	14.454200
75%	0.000000	31.000000
max	6.000000	512.329200

Age data has normal-like distibution. Mean - 29.7, Median - 28.

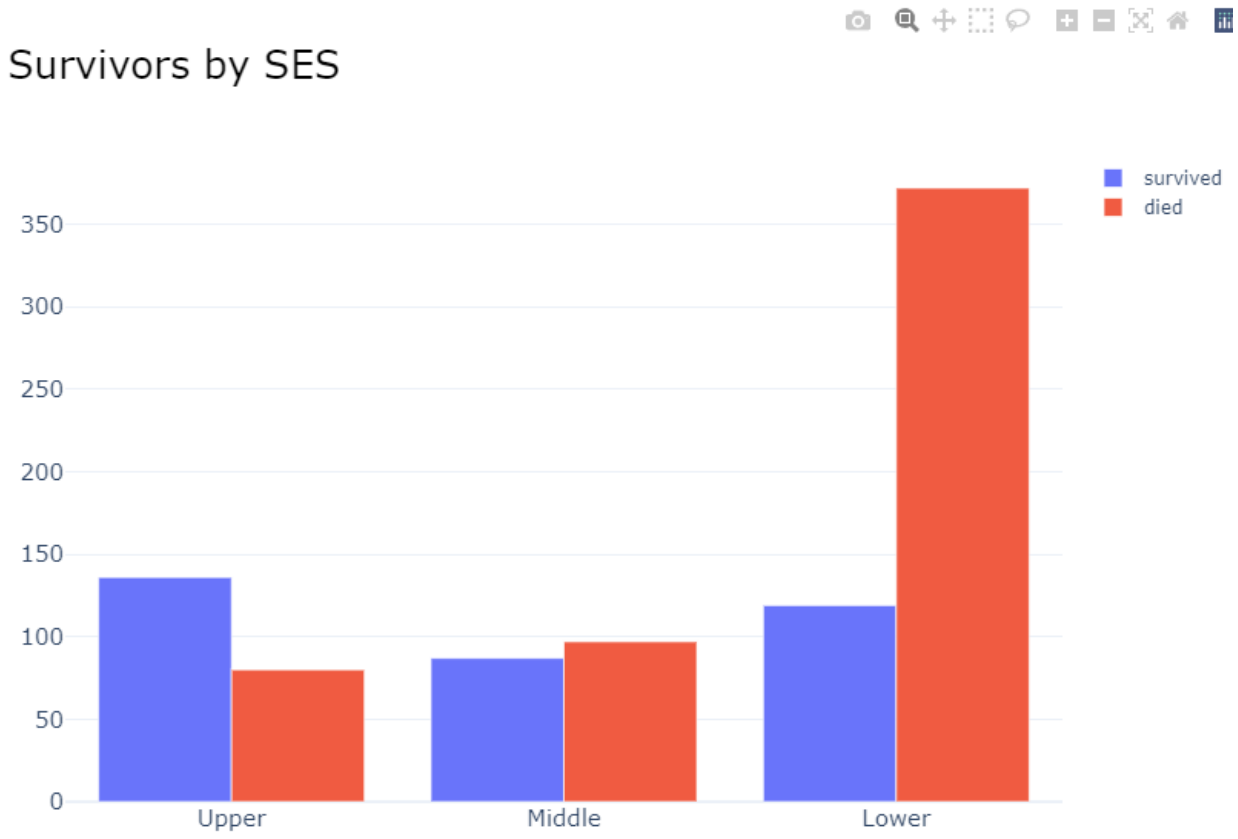


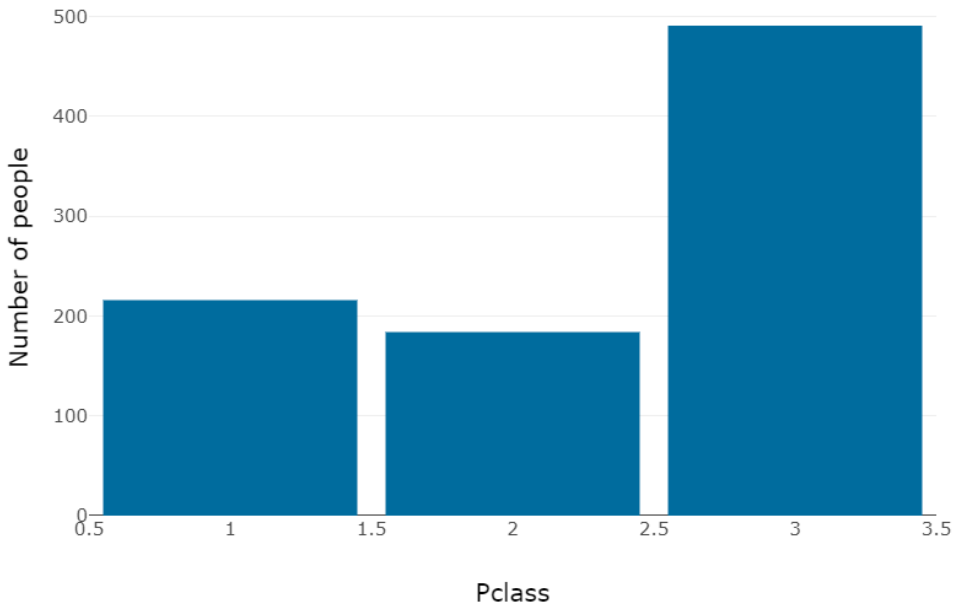


Females were much more likely to survive.

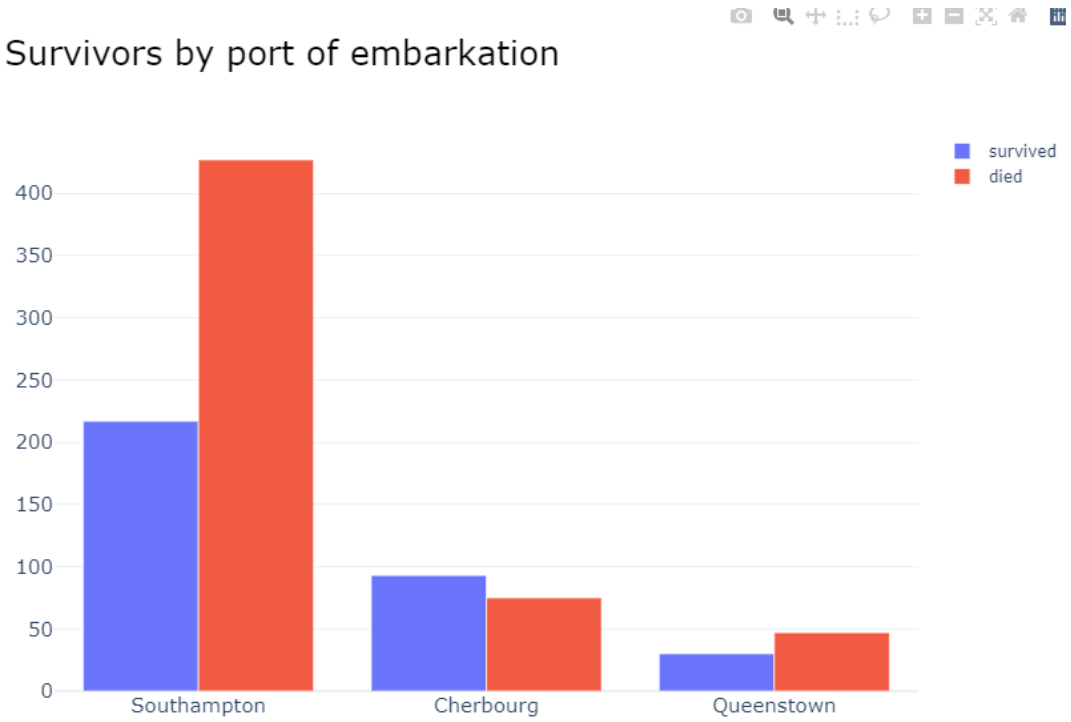


Upper and middle class people had a great advantage in terms of surviving, but lower class were the dominant group on board.

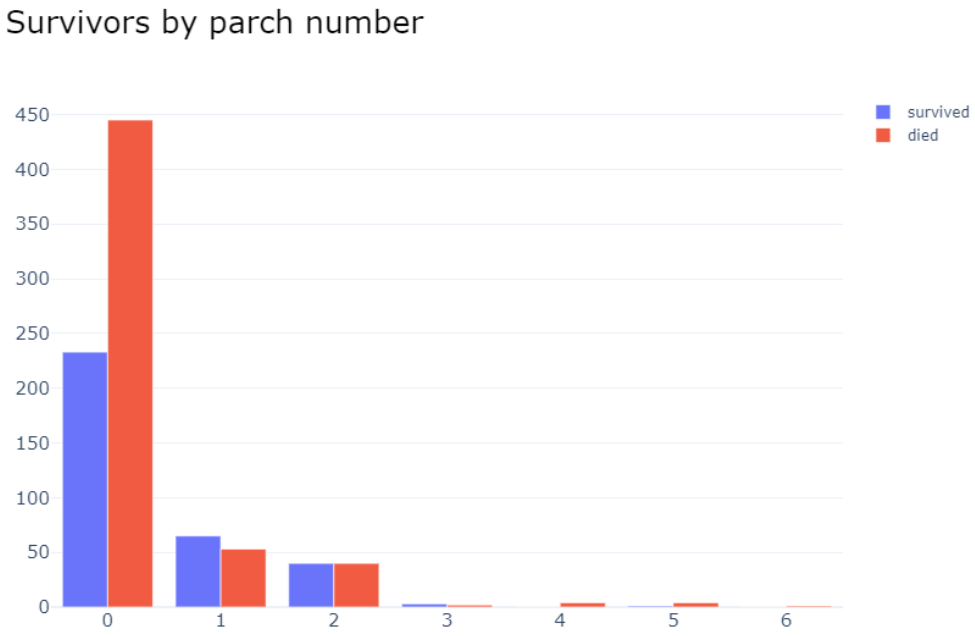


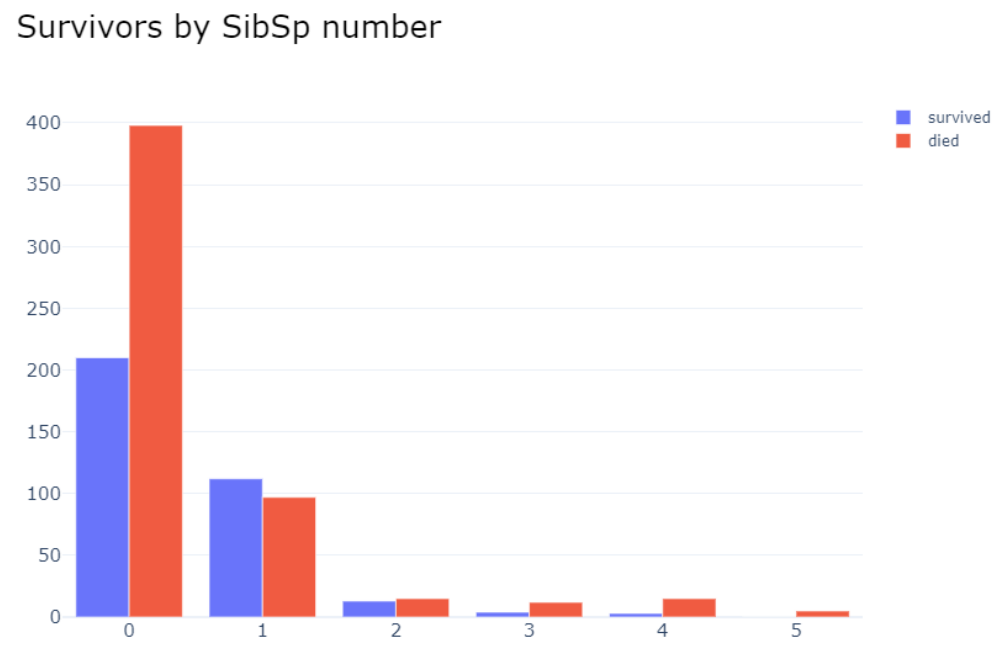


People from Southampton had the lowest chance of survival. This may have been due to the way rooms were allocated.

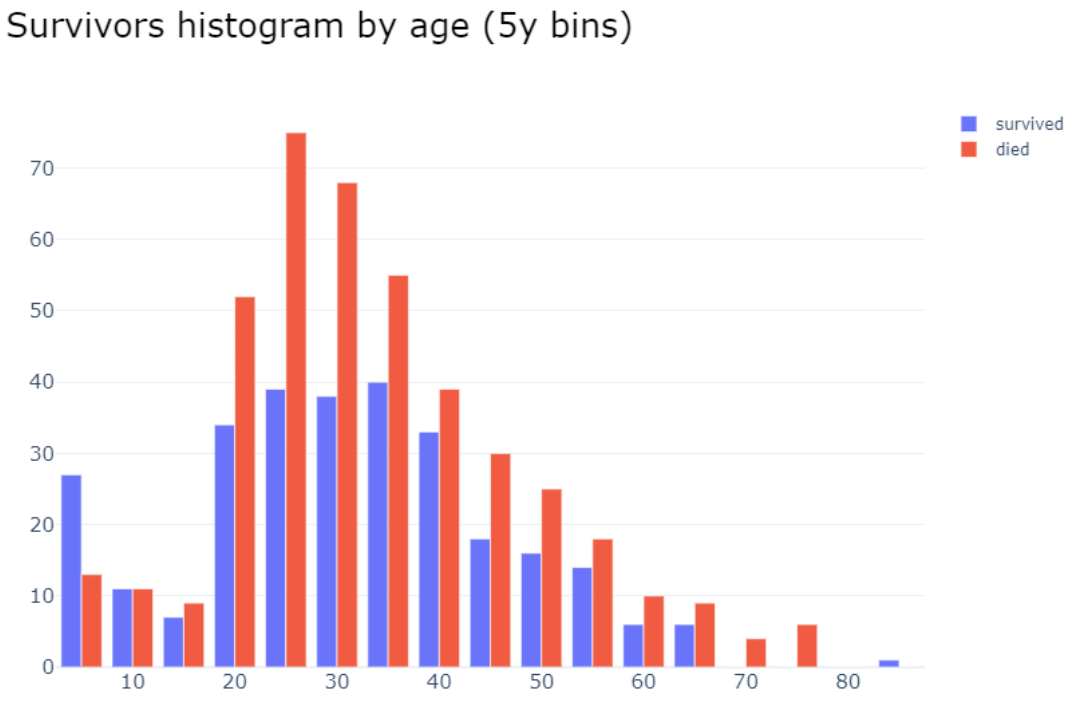


Only a third of those without family on board survived. Having 1 or 2 relatives on board gave the best chance of survival.

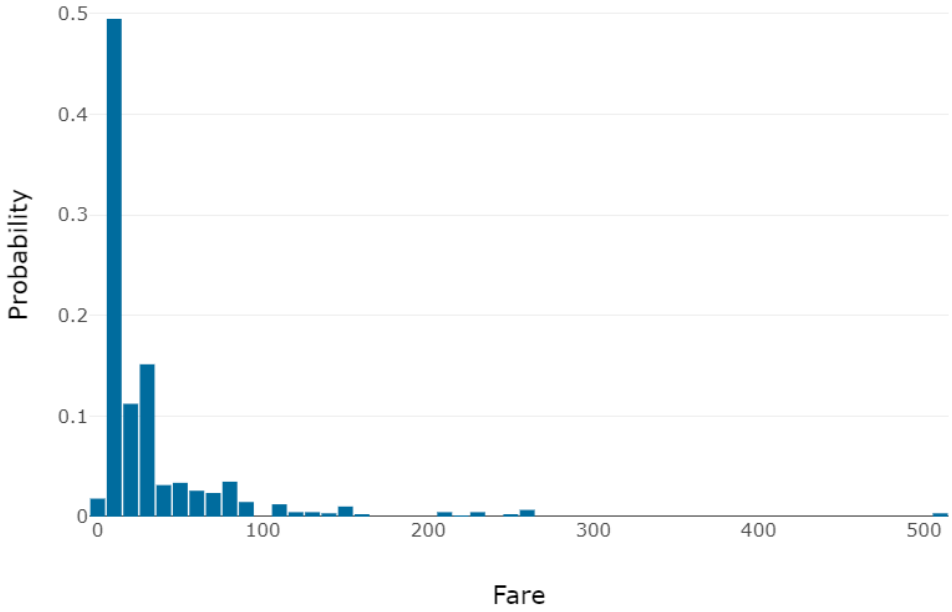




Age histogram shows that most of the children were saved.

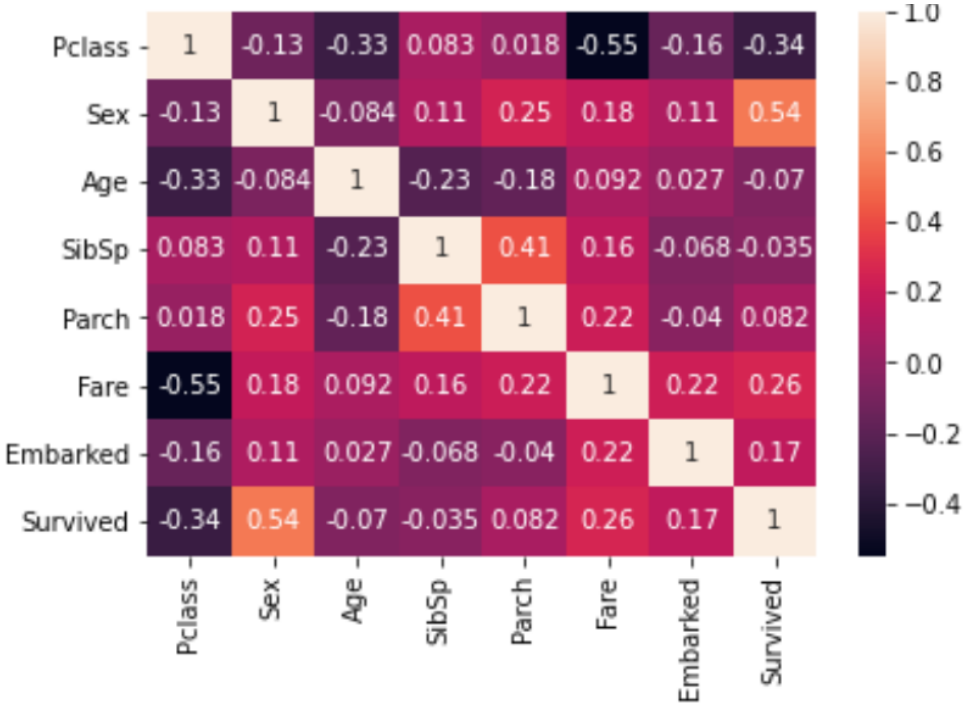


I honestly don’t know what the „Fare” data means in this context



Name, Cabin and Ticket data are descriptive and PassengerId is artificialy generated, so I skipped those columns while preparing final data. NA’s in Age were replaced by mean column age. In gender column females were replaced with 0’s and males with 1’s. In Embarked column S,Q,C were replaced with 0,1,2 respectively.

Correlation plot:



Final data summary

	Pclass	Sex	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	2.308642	0.352413	29.699118	0.523008	0.381594	32.204208
std	0.836071	0.477990	13.002015	1.102743	0.806057	49.693429
min	1.000000	0.000000	0.420000	0.000000	0.000000	0.000000
25%	2.000000	0.000000	22.000000	0.000000	0.000000	7.910400
50%	3.000000	0.000000	29.699118	0.000000	0.000000	14.454200
75%	3.000000	1.000000	35.000000	1.000000	0.000000	31.000000
max	3.000000	1.000000	80.000000	8.000000	6.000000	512.329200

	Embarked	Survived
count	891.000000	891.000000
mean	0.463524	0.383838
std	0.791503	0.486592
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	1.000000	1.000000
max	2.000000	1.000000

Models

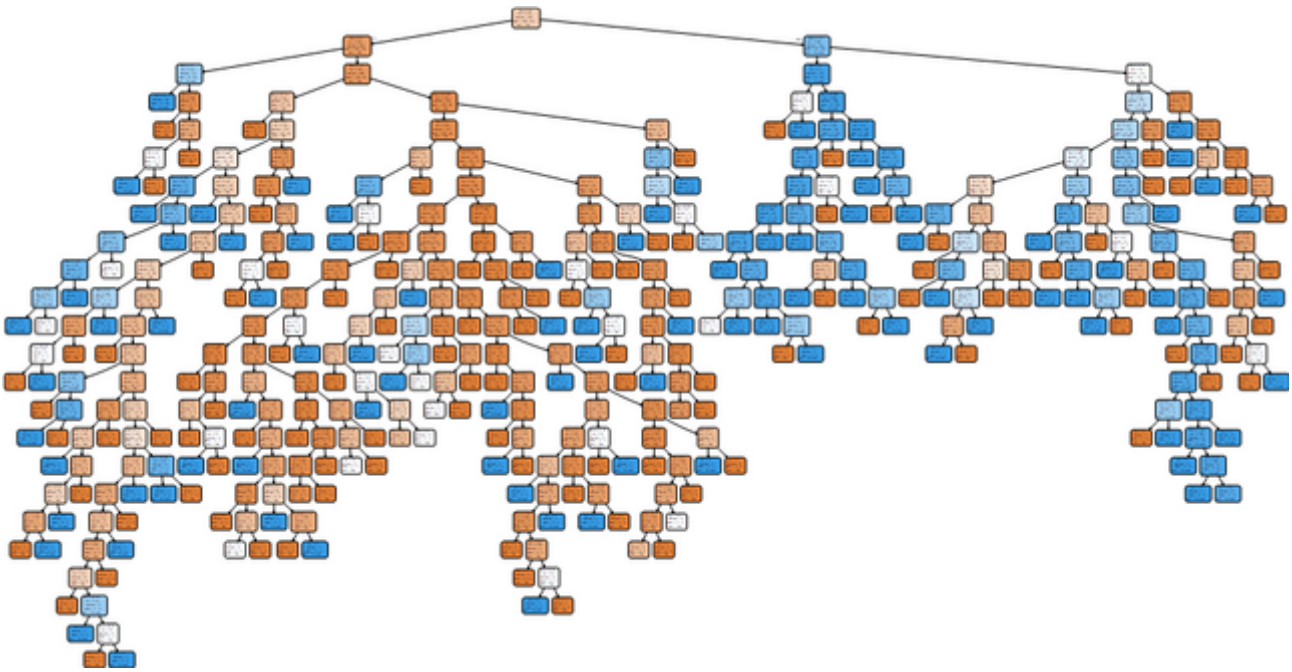
I tested four different Decision Tree classifiers to predict Survived class using default settings, and [default, 2, 5, 10] max depths. Models were evaluated using *Accuracy, Balanced Accuracy, F-measure, Presicion and ROCS* metrics. However, it is important to mention that they were measured on the training data, which is not very informative. It was impossible to test it on the test dataset, because there is no ‘Survived’ column in it, which is the whole point of using this dataset.

Results:

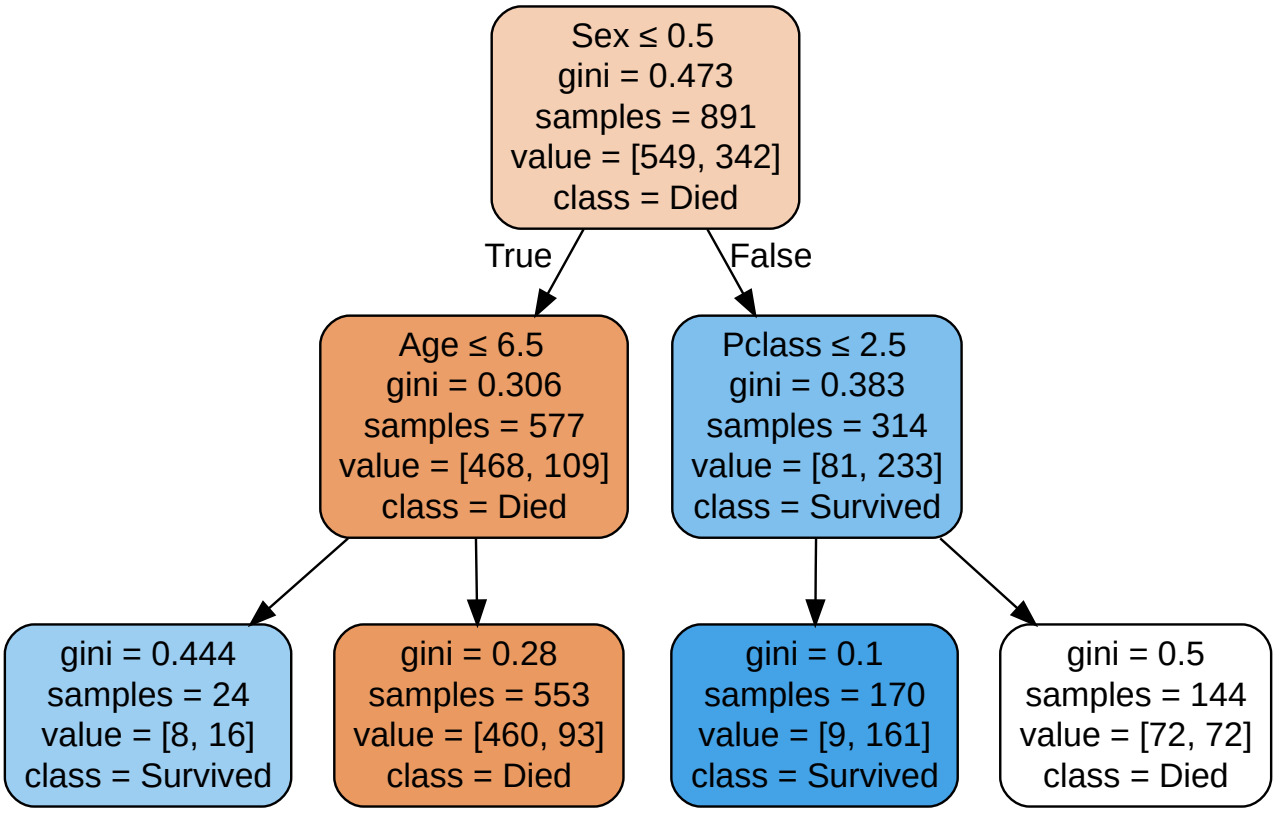
```
--------------------------------------
     Default settings
    Accuracy: 0.9820426487093153
    Balanced accuracy: 0.9777106701179177
    F-measure: 0.976190476190476
    Presicion: 0.9939393939393939
    Receiver Operating Characteristic Curve: 0.9777106701179178
--------------------------------------
     Max depth = 2
    Accuracy: 0.7957351290684624
    Balanced accuracy: 0.7432892340139967
    F-measure: 0.6604477611940299
    Presicion: 0.9123711340206185
    Receiver Operating Characteristic Curve: 0.7432892340139968
--------------------------------------
     Max depth = 5
    Accuracy: 0.8417508417508418
    Balanced accuracy: 0.8219729651998849
    F-measure: 0.7813953488372092
    Presicion: 0.8316831683168316
    Receiver Operating Characteristic Curve: 0.821972965199885
--------------------------------------
     Max depth = 10
    Accuracy: 0.9248035914702581
    Balanced accuracy: 0.9092129230179273
    F-measure: 0.895800933125972
    Presicion: 0.9568106312292359
    Receiver Operating Characteristic Curve: 0.9092129230179273
```

Trees (better resolution can be found in „tree_[number].svg” files):

Default:

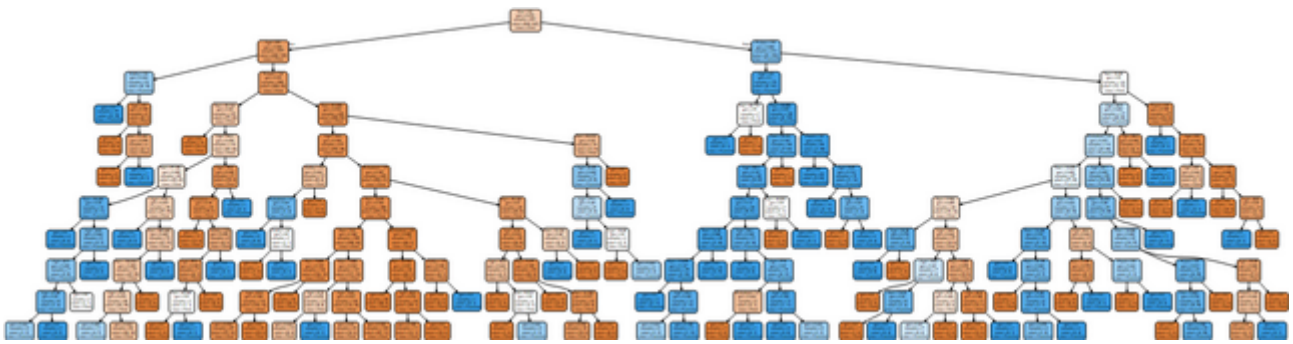


Max_depth = 2



Max_depth = 5

Max_depth = 10



Then I’ve trained Random Forest model with default parameters. It turned out that there was no significant change in best fit compared to the Decision Tree model.

```
--------------------------------------
     Default settings
    Accuracy: 0.9820426487093153
    Balanced accuracy: 0.9793643945930399
    F-measure: 0.976401179941003
    Presicion: 0.9851190476190477
    Receiver Operating Characteristic Curve: 0.9793643945930399
```

I’ve trained KNN models with 7 different k values: [2,3,5,10,25,50,70]. As expected, lower values returned the best fits.

```
--------------------------------------
     2-NN
    Accuracy: 0.8406285072951739
    Balanced accuracy: 0.7935001438021283
    F-measure: 0.73992673992674
    Presicion: 0.9901960784313726
    Receiver Operating Characteristic Curve: 0.7935001438021283
--------------------------------------
     3-NN
    Accuracy: 0.8361391694725028
    Balanced accuracy: 0.8201754385964912
    F-measure: 0.7787878787878788
    Presicion: 0.8081761006289309
    Receiver Operating Characteristic Curve: 0.8201754385964912
--------------------------------------
     5-NN
    Accuracy: 0.8092031425364759
    Balanced accuracy: 0.7867414437733677
    F-measure: 0.735202492211838
    Presicion: 0.7866666666666666
    Receiver Operating Characteristic Curve: 0.7867414437733679
--------------------------------------
     10-NN
    Accuracy: 0.7665544332210998
    Balanced accuracy: 0.7317371297095197
    F-measure: 0.6567656765676567
    Presicion: 0.7537878787878788
    Receiver Operating Characteristic Curve: 0.7317371297095198
```

```
--------------------------------------
     25-NN
    Accuracy: 0.7261503928170595
    Balanced accuracy: 0.6829642412041032
    F-measure: 0.5821917808219178
    Presicion: 0.7024793388429752
    Receiver Operating Characteristic Curve: 0.682964241204103
--------------------------------------
     50-NN
    Accuracy: 0.7037037037037037
    Balanced accuracy: 0.6504170261719873
    F-measure: 0.5217391304347826
    Presicion: 0.6857142857142857
    Receiver Operating Characteristic Curve: 0.6504170261719874
--------------------------------------
     70-NN
    Accuracy: 0.6879910213243546
    Balanced accuracy: 0.6371153293068738
    F-measure: 0.5070921985815602
    Presicion: 0.6441441441441441
    Receiver Operating Characteristic Curve: 0.6371153293068736
```

SVM model was tested using GridSearchCV with this parameter grid:

```
param_grid = {'C': [0.1, 1, 10, 100, 1000],
              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
              'kernel': ['rbf']}
```

Best parameters fit: {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}.

Results:

```
Accuracy: 0.8597081930415263
Balanced accuracy: 0.8508771929824561
F-measure: 0.8164464023494861
Presicion: 0.8200589970501475
Receiver Operating Characteristic Curve: 0.8508771929824561
```

I’ve trained MLPClassifier using 4 different activation and solver combinations. Results:

```
-------------------------------------
     solver: adam, activation: relu
    Accuracy: 0.8204264870931538
    Balanced accuracy: 0.8030150512894259
    F-measure: 0.756838905775076
    Presicion: 0.7879746835443038
    Receiver Operating Characteristic Curve: 0.8030150512894256
-------------------------------------
     solver: adam, activation: tanh
    Accuracy: 0.8170594837261503
    Balanced accuracy: 0.8102051577030007
    F-measure: 0.7661406025824964
    Presicion: 0.752112676056338
    Receiver Operating Characteristic Curve: 0.8102051577030006
-------------------------------------
     solver: lbfgs, activation: relu
    Accuracy: 0.8540965207631874
    Balanced accuracy: 0.8336449046112548
    F-measure: 0.7968750000000001
    Presicion: 0.8557046979865772
    Receiver Operating Characteristic Curve: 0.8336449046112548
-------------------------------------
     solver: lbfgs, activation: tanh
    Accuracy: 0.898989898989899
    Balanced accuracy: 0.8877145048413383
    F-measure: 0.8644578313253012
    Presicion: 0.8913043478260869
    Receiver Operating Characteristic Curve: 0.8877145048413382
```

lbfgs and tanh combination was the best fit.

Conclusion

It turns out that simple Decision Tree model did better than all MLPs. SVM and 2/3-NN returned predictions similar to MLPs. However, caution should be exercised as the metrics are not representative on the training data.