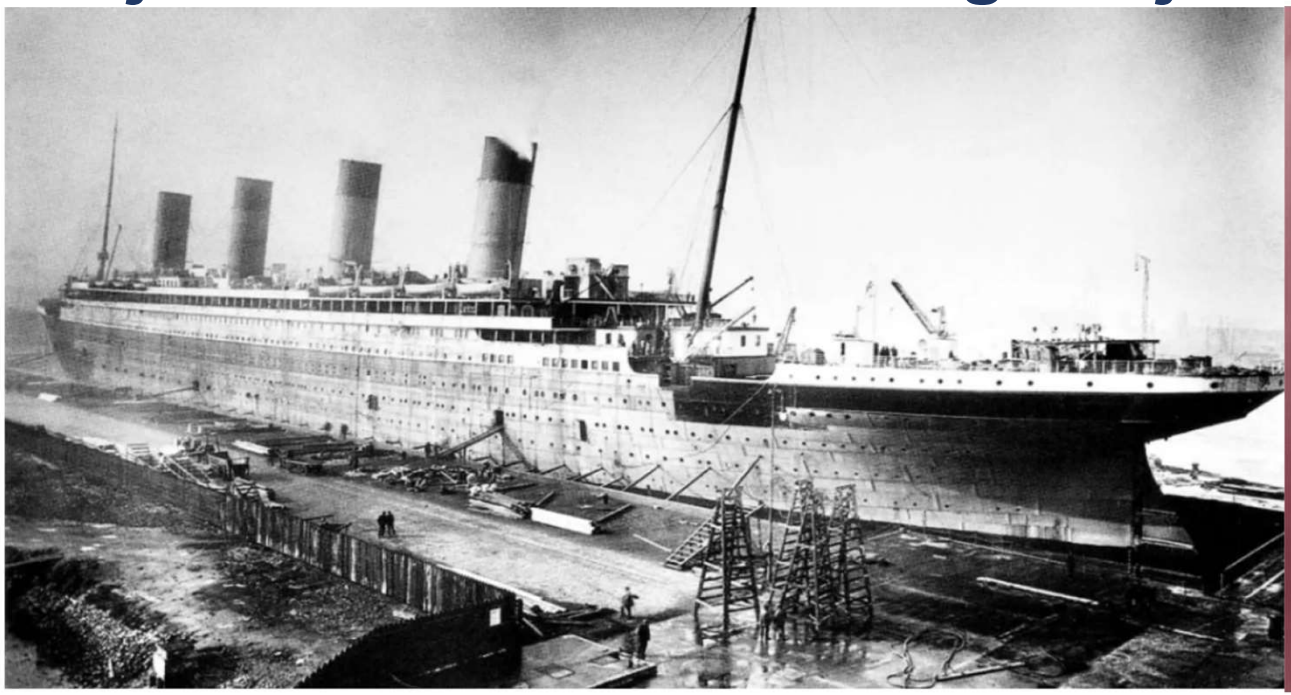
Project on machine learning subject: ’Titanic’



Insurance company must know:

Who will survive the cruise and how to protect company from bankruptcy?

Presenter: Agnieszka Kamińska

INDEX

* ML = Machine Learning

1. Input data

'train.csv' - scrap of training data input. Training file contains 891 rows with passangers data.

	A	B	C	D	E	F	G	H	I	J	K	L
1	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, Mr. Owen Harris	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. William Henry	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr. James	male		0	0	330877	8.4583		Q

Parameters explanation:

'Survived' - Indicator of survival (0 = No, 1 = Yes),

'Pclass' - Type of class in which passengers traveled (1 = 'The Richest' Class, 2 = 'The Middle Class', 3 = 'The Poorest Class'),

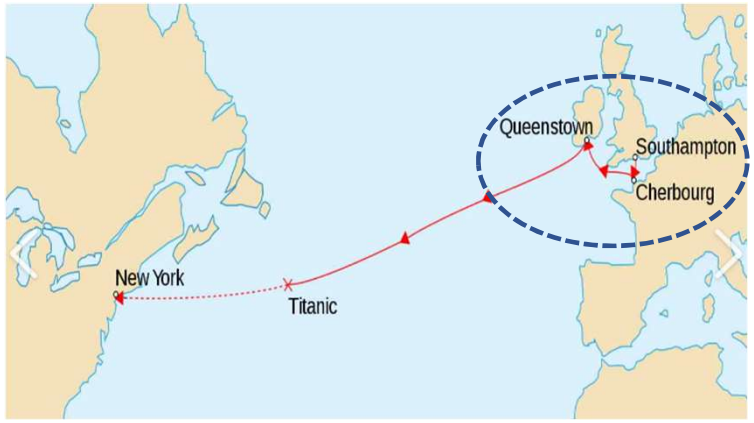
'Sex' - Gender of the passenger,

'Age' - Passenger's age,

'SibSp' - Number of siblings/spouses aboard the Titanic,

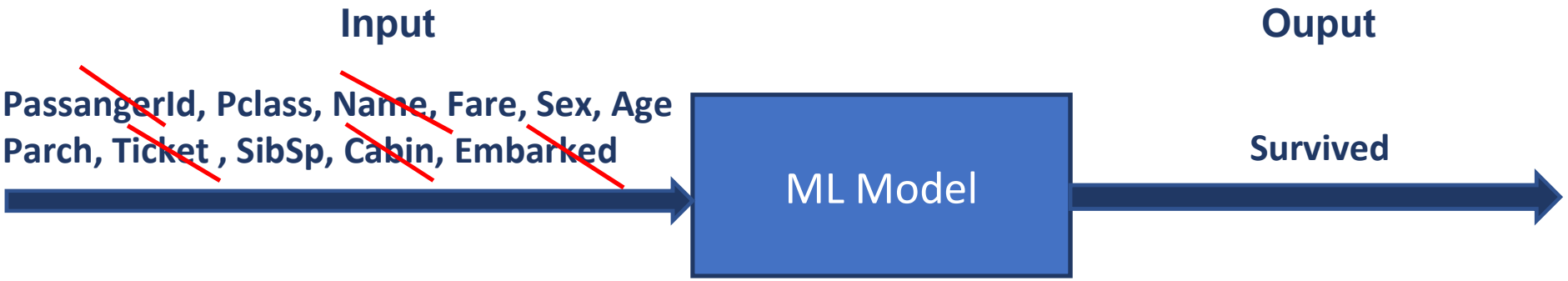
'Parch' - Number of parents/children aboard the Titanic,

'Fare' - The amount of the cruise ticket fee paid by the passenger in $,



'Embarked' - Port of embarkation (C = Cherbourg(FR), Q = Queenstown(IE), S = Southampton(UK).

2. ML model



3. Data Cleansing

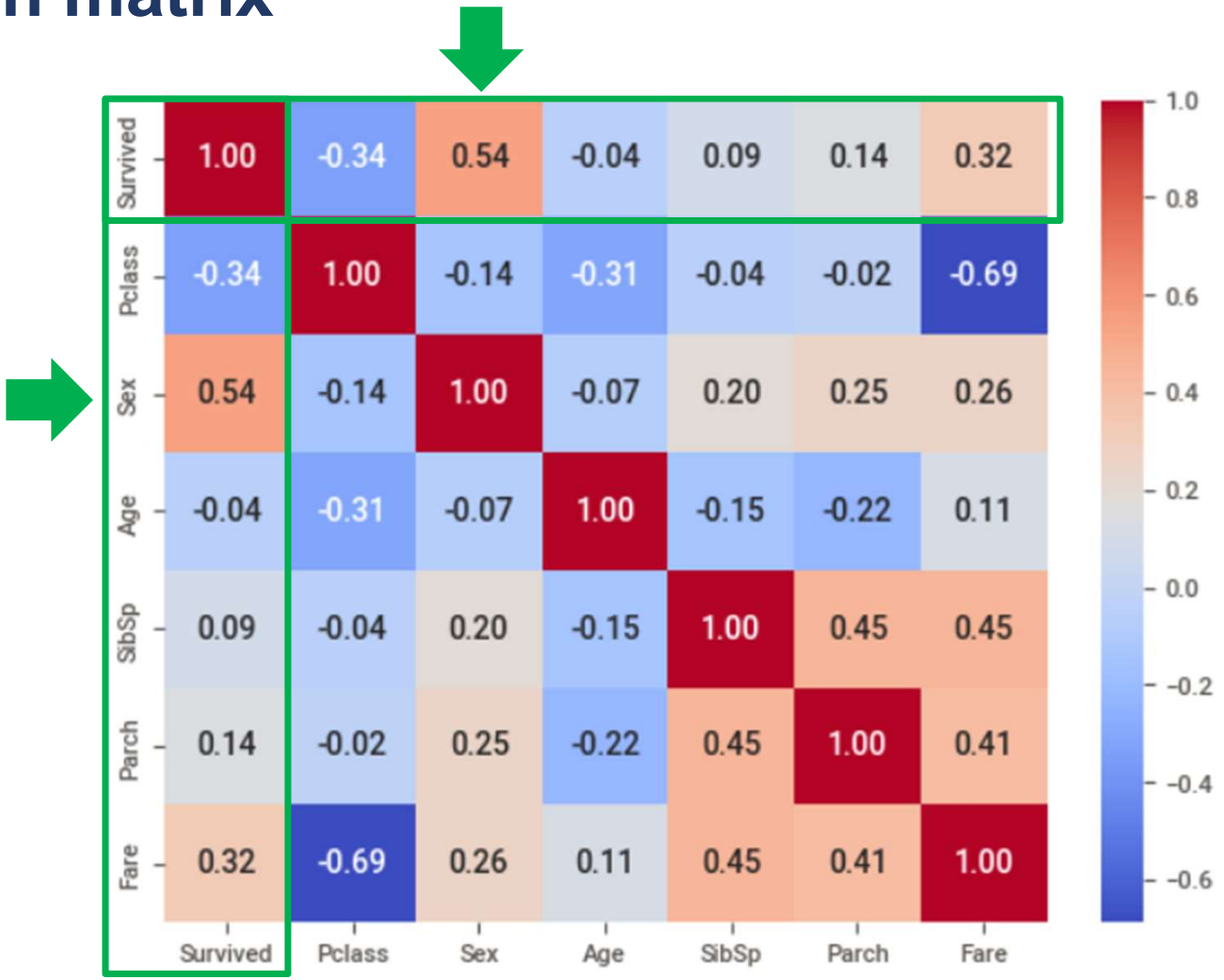
- Some of the input parameters have been removed from the 'train.csv' file, as data of negligible importance or or reducing the performance of the model:
- **PassengerId, Name, Ticket, Cabin, Embarked.**

- Among remaining input parameters:
- **Pclass, Fare, Sex, Age, Parch, SibSp** analysis showed, that only 'Age' parameter had 20% of NA-s in data.

- Modifications done on final 'train.csv':

  Sex: changed**: 'male' → '0'** , **'female' →'1'**
  Age: blanks replaced by 'Median' value

4. Correlation matrix

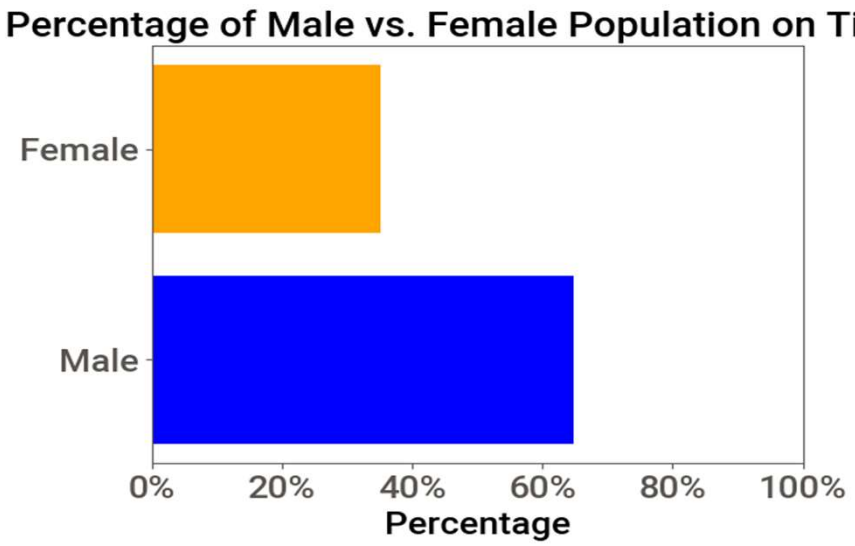


- We can observe **the strongest correlation coefficient between 'Sex' vs.'Survived'**

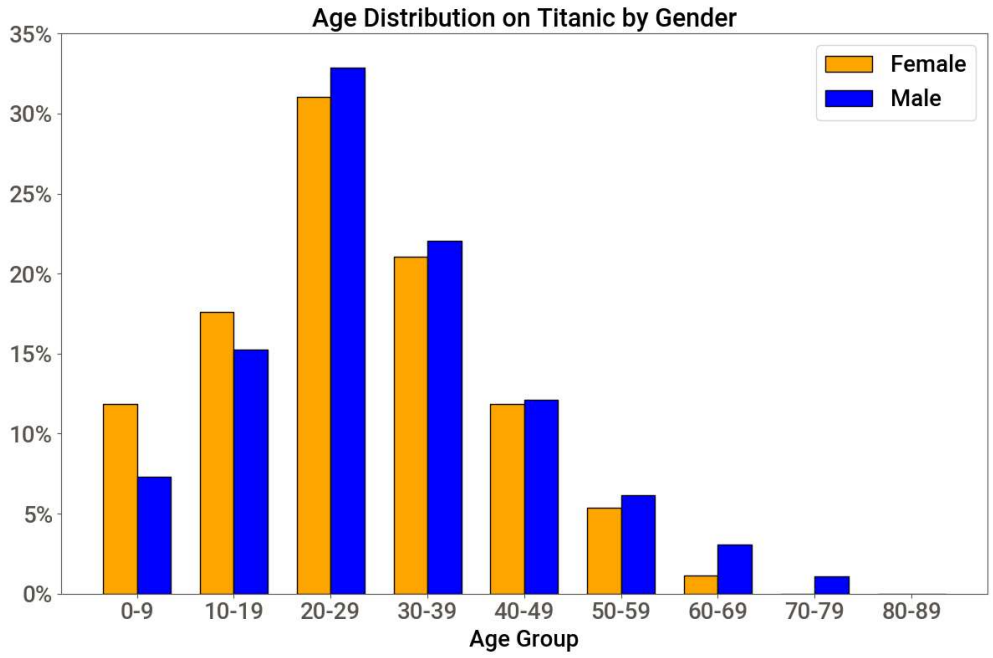
5. Charts with interesting parameters from input data

- The population of the input data from the perspective of gender division

- **Females in total: 314 (35%)**
- **Males in total: 577 (65%)**



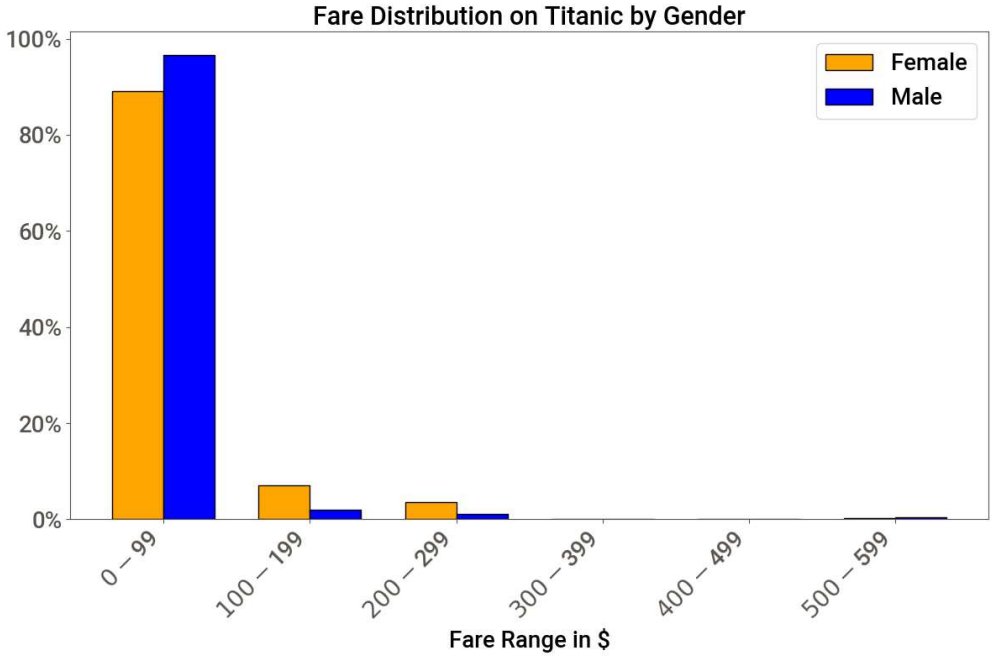
- **Females Age: Min: 1, Max: 63, Mean:28**
- **Males Age: Min: 0, Max: 80, Mean:30**



5. Charts with interesting parameters from input data

- The population of the input data from the perspective of gender division

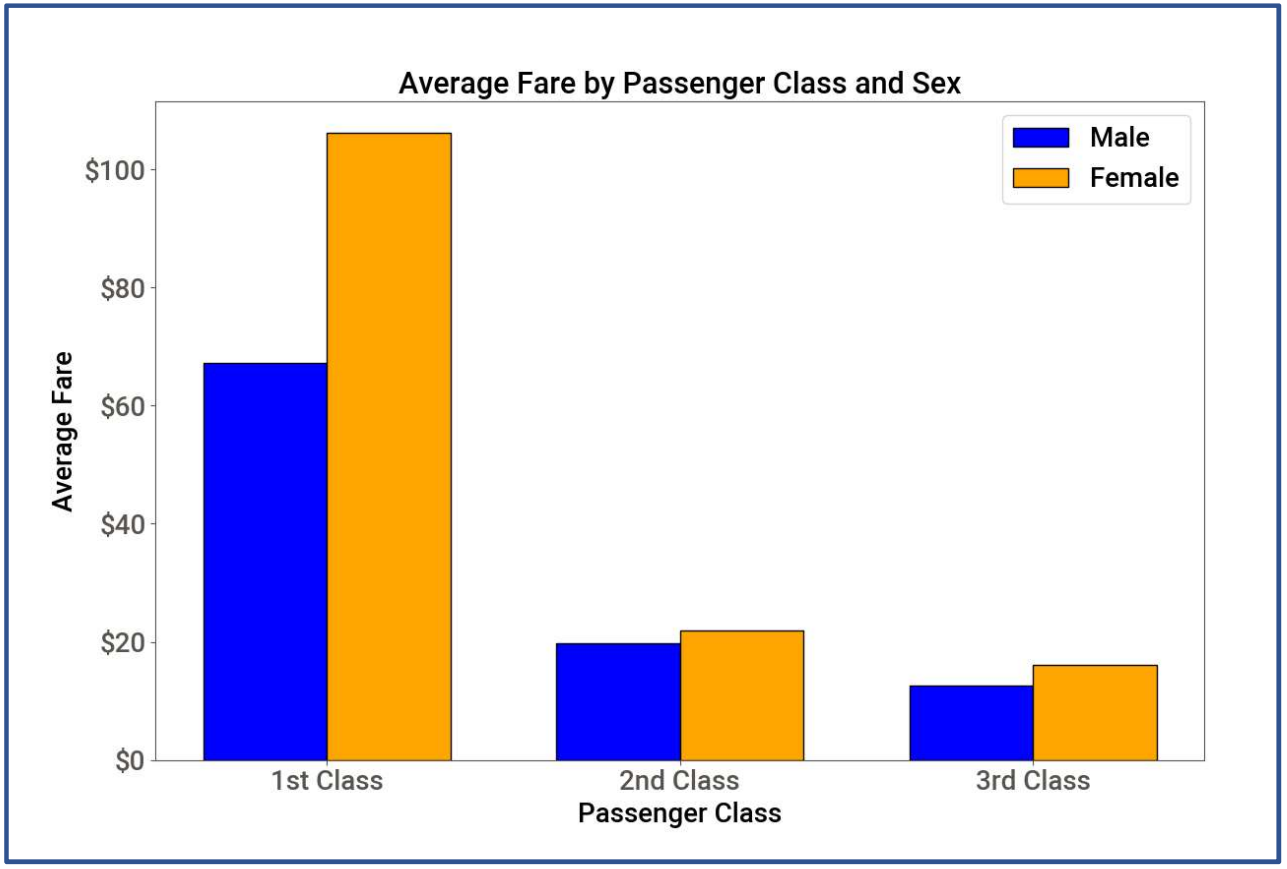
- **Females Fare: Min: 7$, Max: 512$, Mean:44$**
- **Males Fare: Min: 0$, Max: 512$, Mean:26$**



- **Females: Class:1(30%), Class2:(24%), Class3:(46%)**
- **Males : Class:1(21%), Class2:(19%), Class3:(60%)**

5. Charts with interesting parameters from input data

- The population of the input data from the perspective of gender division



6. Metrix influencing the final choice of ML model

√ **Accuracy = 0.8956** The ratio of correctly predicted instances to the total instances.

√ **Precision =** $\frac{true\ positives}{true\ positives + \boldsymbol{false\ positives}}$

Precision is crucial if you are to predict survivors and you want to minimize the number of non-survivors predicted as survivors (**FP**)

√ **Recall** = $\frac{true\ positives}{true\ positives + \boldsymbol{false\ negatives}}$

Recall is crucial if you want to ensure that as many actual survivors as possible are identified (minimizing **FN**).

√ **Model confusion matrix**

		Predicted	
		Negative (N) -	Positive (P) +
Actual	Negative -	True Negative (TN)	False Positive (FP) Type I Error
	Positive +	False Negative (FN) Type II Error	True Positive (TP)

How predicted chosen model

Predicted	0	1
Actual		
0	520	29
1	66	276

FP
False positives

False negatives FN

Model rates

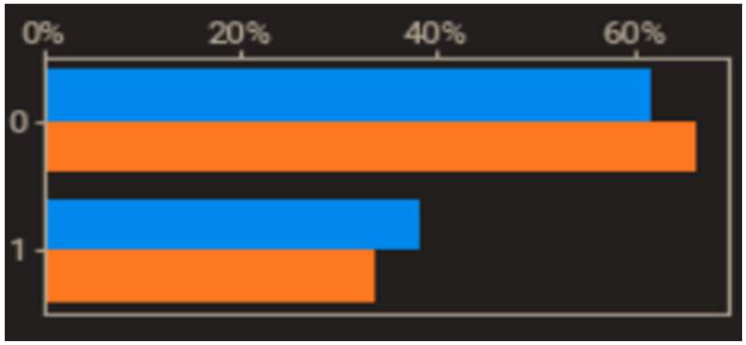
```
print(f'Precision 1: {278/(278+29):.2f}')
print(f'Recall 1: {278/(278+66):.2f}')

print(f'Precision 0: {520/(520+66):.2f}')
print(f'Recall 0: {520/(520+29):.2f}')
```

```
Precision 1: 0.91
Recall 1: 0.81
Precision 0: 0.89
Recall 0: 0.95
```

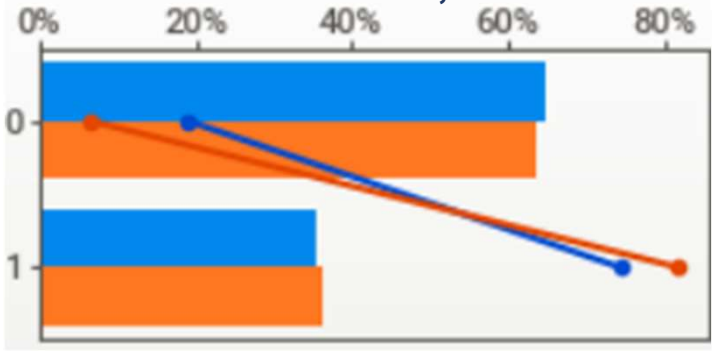
7. ML model performance

Train		Test
891	ROWS	418

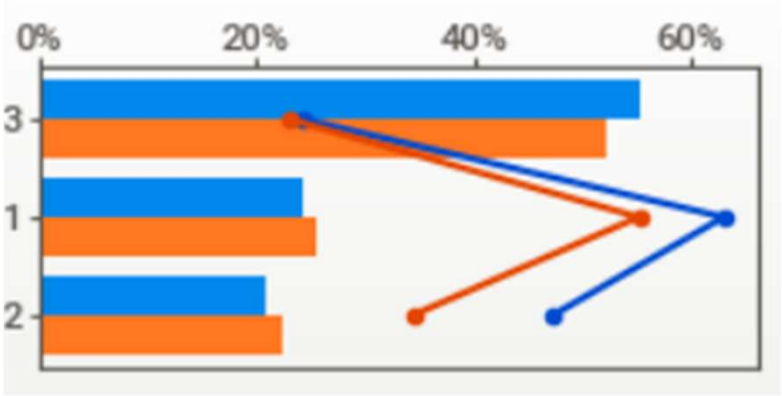
Non - Survived = 0, Survived =1



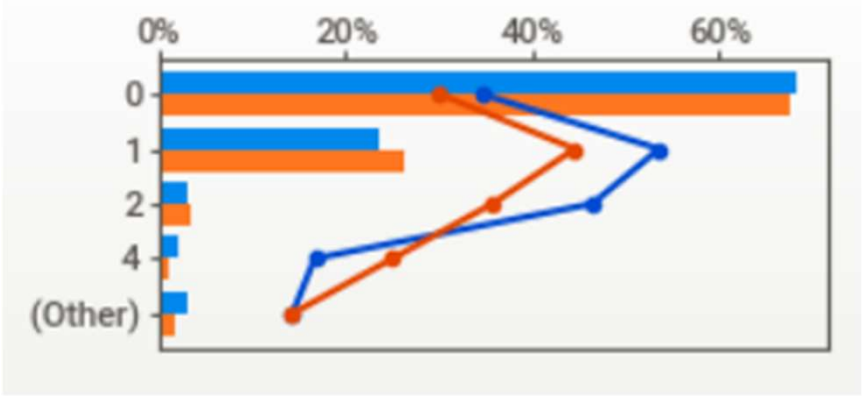
Non-Survived. Sex = 0 - Males, Sex =1- Females



Pclass



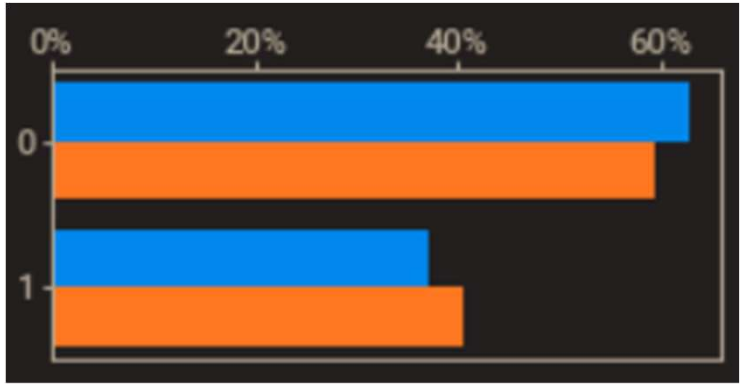
SibSp



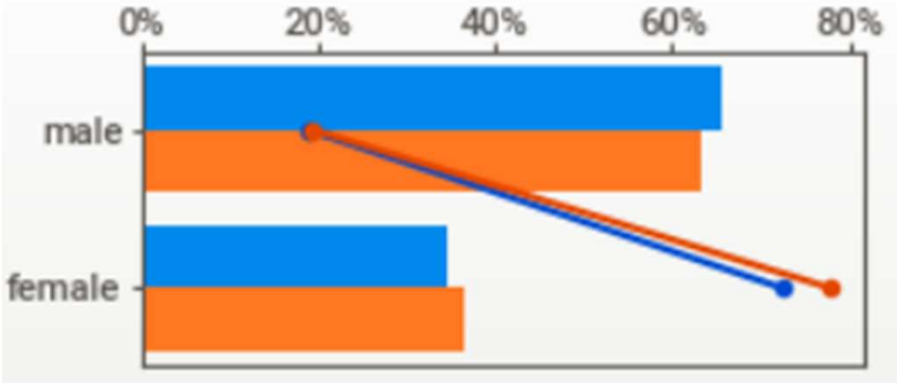
7. ML model performance

Train		Test
596	ROWS	295

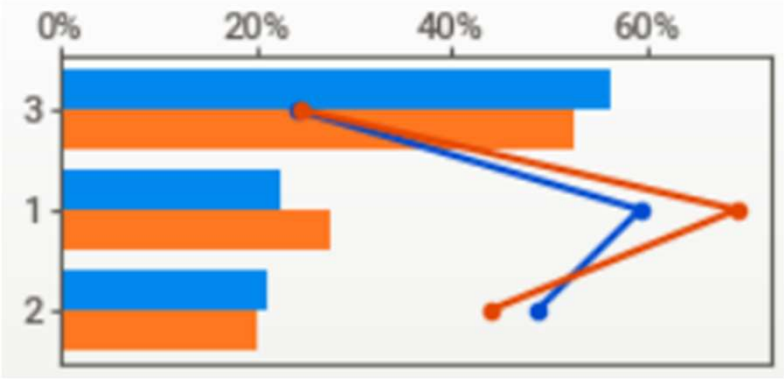
Non - Survived = 0, Survived =1



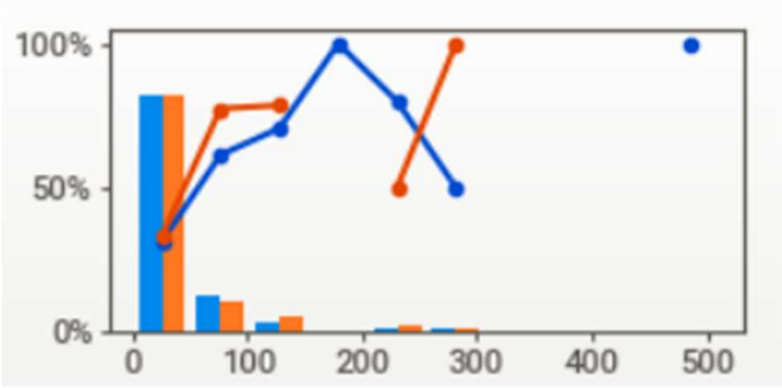
Sex, Non-Survived



Pclass



Fare



8. Recommendations for insurance company

```
Feature Pclass: 0.189
Feature Sex: 0.714
Feature Age: 0.020
Feature SibSp: 0.039
Feature Parch: 0.015
Feature Fare: 0.023
```

- **Output from the final ML model** - XGBoost, shows **the two key parameters with their percentage impact on survival**:

  1. Sex: 71.4%,
  2. Pclass: 18.9%.

  The rest stands for: SibSp: 3.9%, Fare: 2.3%, Age: 2.0%, Parch: 1.5%

- We can assess our **'Survived' set of passengers recommended** for being insured as:

- **Almost All the Women:** ~75% Survived subtract the poorest one, which didn't survive and they couldn't afford insurance anyway,

- **Rich people**: if someone paid for a ticket $100+, they have a minimum of 65% chance for surviving and the fact, that the rich will insure themselves for a large amount, it will be a plus for insurance company.
  Survived: 63% of Class-1, 47% of Class-2, 24% of Class-3.

9.

Q&A

Back-up

❑ Models which where trained and checked:

- **Decision tree** Using graph theory nomenclature, a decision tree can be defined as a directed, acyclic, and connected graph that has only one vertex, called the root. It is assumed that the root is located at the top, so the tree grows from top to bottom. The root contains the entire training set in which the objects are defined by a dependent variable (decision attribute) and d independent variables (conditional attributes, or features). The root is a special case of a decision node. At the decision nodes, the set is divided according to the division criterion adopted for it. Tree structure:

✔ Decision node - division point due to the selected division criterion.
✔ Division criterion - a condition for a selected feature according to which division is made in a given node.
✔ Root - the first decision node.
✔ Leaves - terminal nodes.

- **Forest**

A random forest is a group of classification trees working in parallel, based on which the final, joint classification decision is made. The prediction result of the random forest model is the majority vote of the trees included in the forest. That is, in order to classify a new object, it is subjected to independent classification of each tree in the random forest, and then the final classification result is the value of the decision attribute that has been assigned most often. In order to minimize correlations between trees, during the construction of each tree, a random selection of conditional attributes is made, which will be taken into account in a given tree.

- **XGBoost(chosen as the best trained model)**

XGBoost is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.