

# Predykcja przeżycia pasażerów katastrofy Titanica

Klaudia Szczygieł

# Opis projektu

RMS Titanic był największym i najbardziej luksusowym statkiem pasażerskim swoich czasów. W swój pierwszy i ostatni rejs wypłynął do Nowego Jorku z brytyjskiego Southampton 10 kwietnia 1912 r. W nocy z 14 na 15 kwietnia statek zderzył się z górą lodową, w wyniku czego zatonął. W katastrofie zginęło 1502 z 2224 osób na pokładzie. Wydarzenie to jest postrzegane przez ludzi na całym świecie jako najbardziej wstrząsająca tragedia morska wszech czasów. Jedną z głównych przyczyn był brak wystarczającej ilości kamizelek ratunkowych dla wszystkich pasażerów. Czynnikami, które miały istotny wpływ na przeżycie katastrofy były:

1. **Wiek** - dzieci miały większe szanse na przeżycie niż osoby dorosłe
2. **Płeć** - kobiety miały większe szanse przeżycia niż mężczyźni
3. **Klasa** - pasażerowie z pierwszej klasy mieli większe szanse na przeżycia katastrofy niż osoby z niższych sfer

# **Kolejność omawiania projektu**

- 1. Pozyskanie danych wraz z ich opisem.**
- 2. Biblioteki użyte podczas analizy danych oraz modelu predykcyjnego.**
- 3. Przygotowanie danych do modelowania oraz analizy danych.**
- 5. Modele predykcyjne.**

# Pozyskanie danych

1. Rzeczywiste dane dotyczące pasażerów Titanica zostały  
ściągnięte z platformy Kaggle (<https://www.kaggle.com/c/titanic/data>)
2. Dane składają się ze zbioru treningowego oraz testowego.  
Zbiór testowy składa się z 981 wierszy oraz 12 atrybutów  
natomiast zbiór testowy składa się z 418 wierszy i 11  
atrybutów.

# Opis atrybutów

1. **PassengerId** - numer identyfikacyjny pasażera
2. **Survived** - zmienna binarna, przyporządkowująca wartość równą 0 pasażerom, którzy nie przeżyli katastrofy oraz przyporządkowująca wartość 1 pasażerom, którzy katastrofę przeżyli. Jest to zmienna, którą chcemy modelować, zatem nie jest ona dostępna w zbiorze testowym
3. **Pclass** - informacja o klasie z której pochodził dany pasażer (1 klasa, 2 klasa, 3 klasa)
4. **Name** - imię oraz nazwisko pasażera (wartość indywidualna dla każdego pasażera)
5. **Sex** - płeć (kobieta/mężczyzna)
6. **Age** - wiek
7. **SibSp** - informacja o liczbie rodzeństwa/kuzynów podróżujących statkiem
8. **Parch** - informacja o liczbie dzieci/rodziców podróżujących statkiem
9. **Ticket** - numer biletu
10. **Fare** - opłata za bilet (ludzie z wyższej klasy płacili drożej)
11. **Cabin** - numer kabiny
12. **Embarked** - miasto z którego dany pasażer wypływał (S - Southampton, C - Cherbourg, Q - Quennstown)

# Biblioteki użyte w projekcie

Cała analiza danych wraz z zastosowanie modeli uczenia maszynowego były przygotowywane w Pythonie. Standardowe biblioteki jakie zostały użyte podczas wstępnej analizy danych to bibliotek pandas i numpy. Z biblioteki scikit-learn, korzystaliśmy w celu modelowania oraz ewoluowania modelu predykcyjnego.

```
#plotly
import plotly.plotly as py
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
from plotly import tools

from sklearn.metrics import roc_auc_score

#Models
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression, Perceptron, SGDClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC, LinearSVC
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier

from sklearn.model_selection import cross_val_score,train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.grid_search import GridSearchCV
from sklearn.cross_validation import KFold
```

# Przygotowanie danych do modelowania oraz analizy

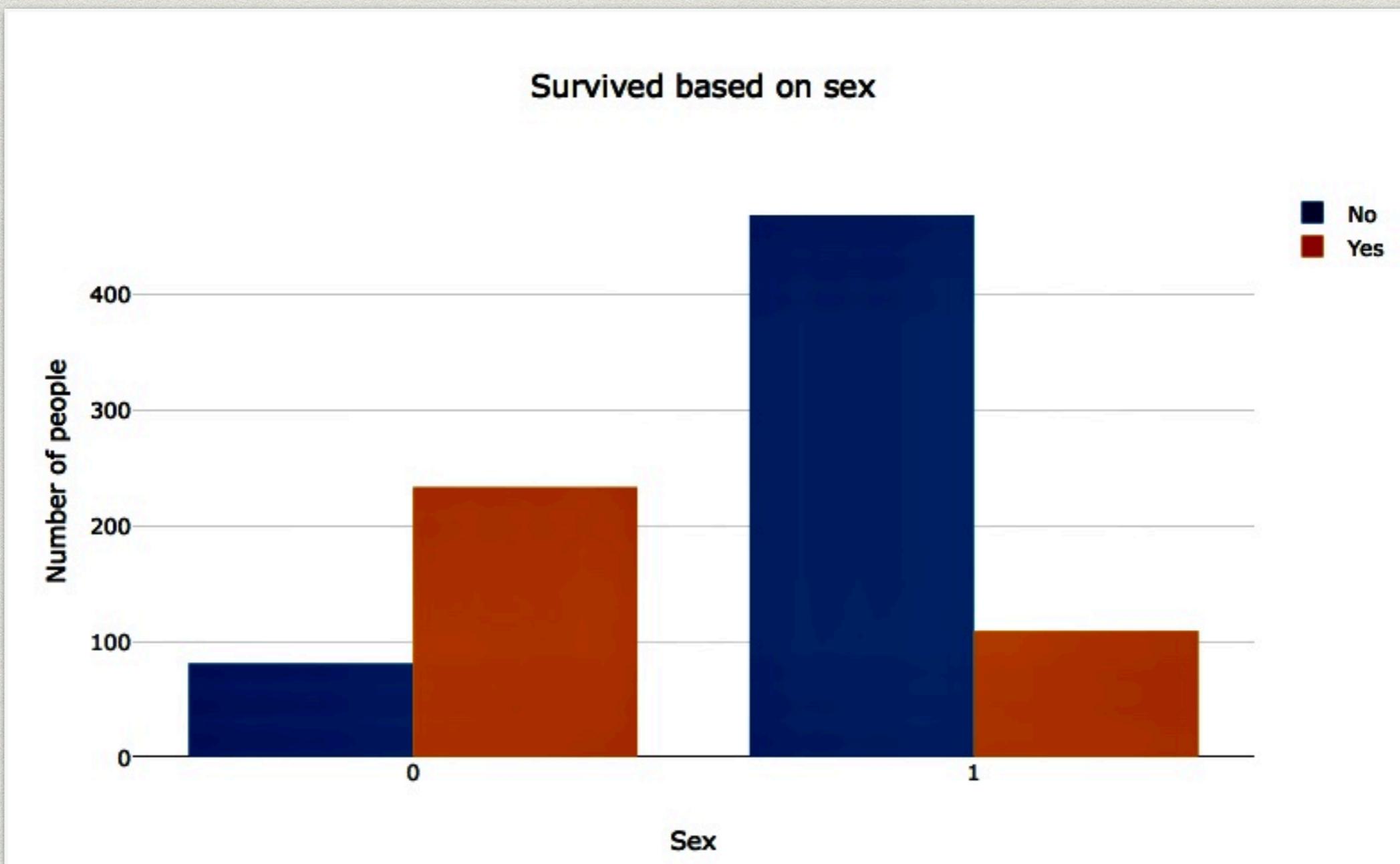
Za nim przejdziemy do modelowania zmiennej Survival, zajmiemy się następującymi problemami:

1. Brak niektórych wartości w zbiorze danych.
2. Feature engineering.
3. Korelacja pomiędzy zmienną, którą chcemy prognozować a pozostałymi zmiennymi.
4. Zamiana zmiennych kategorycznych na zmienne numeryczne.

# Płeć

1. Nie ma brakujących wartości
2. Jest zmienną kategoryczną w związku z czym trzeba będzie ją „zakodować” binarnie (0 - odpowiada kobietą, 1 - mężczyzną)
3. Na podstawie wykresu widać, że większe szanse na przeżycie miały kobiety, niż mężczyźni, dlatego jest to istotna cecha dla naszego modelu

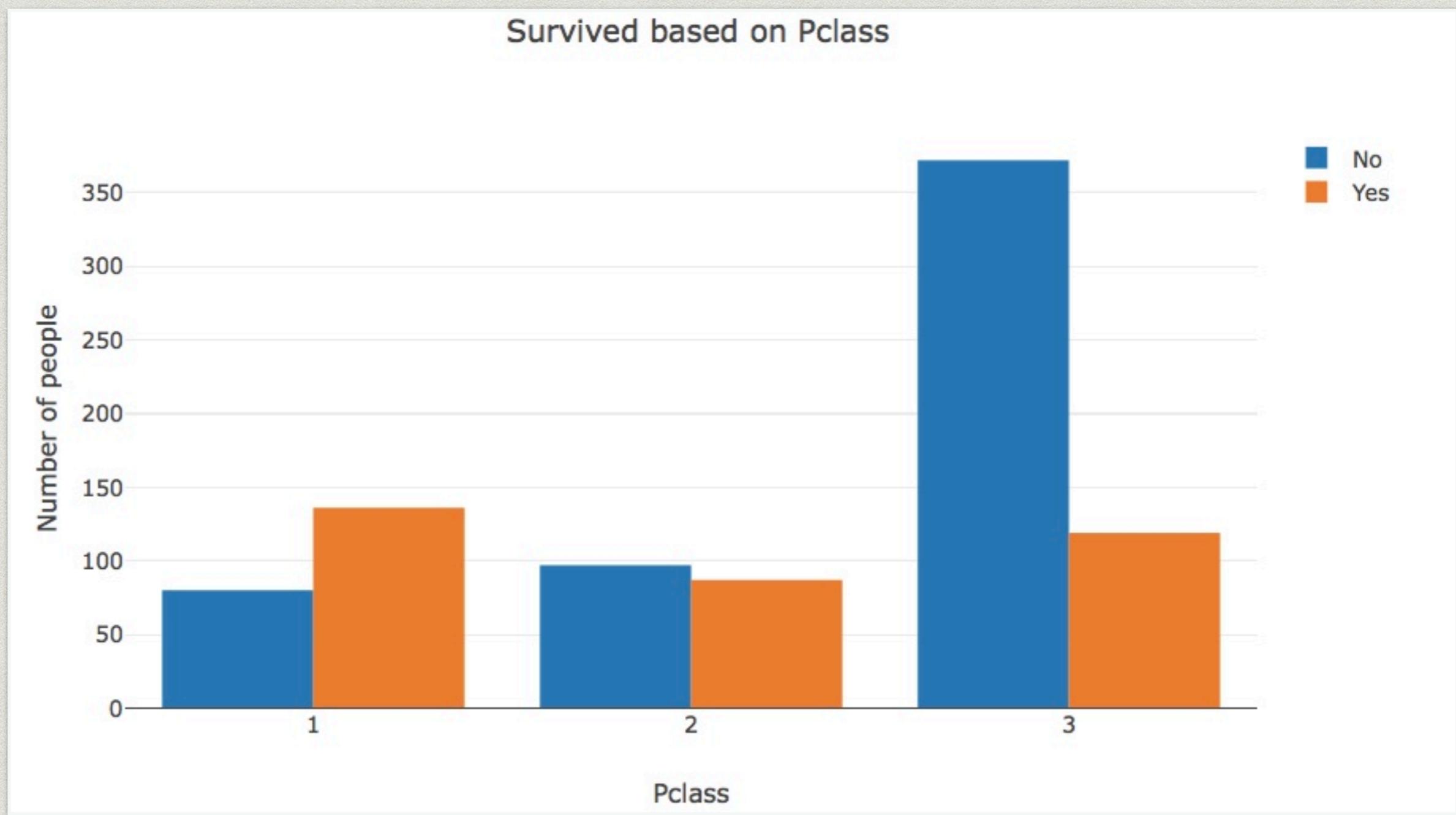
Wykres przedstawiający zależność pomiędzy płcią a przeżyciem katastrofy



# Przynależność do danej klasy

1. Nie ma brakujących wartości
2. Zmienna kategoryczna (1 - odpowiada pierwszej klasie, 2 - odpowiada drugiej klasie, 3 - odpowiada trzeciej klasie)
3. Na podstawie wykresu widać, że jest ona dość mocno skorelowana ze zmienną **Survival**. Większość osób z pierwszej klasy przeżyła katastrofę. Natomiast pasażerowie z drugiej i trzeciej w większości przypadków zginęły. Z wykresu widać również dość dobrze, że największą liczbę ofiar spośród pasażerów stanowiły osoby z klasy trzeciej.

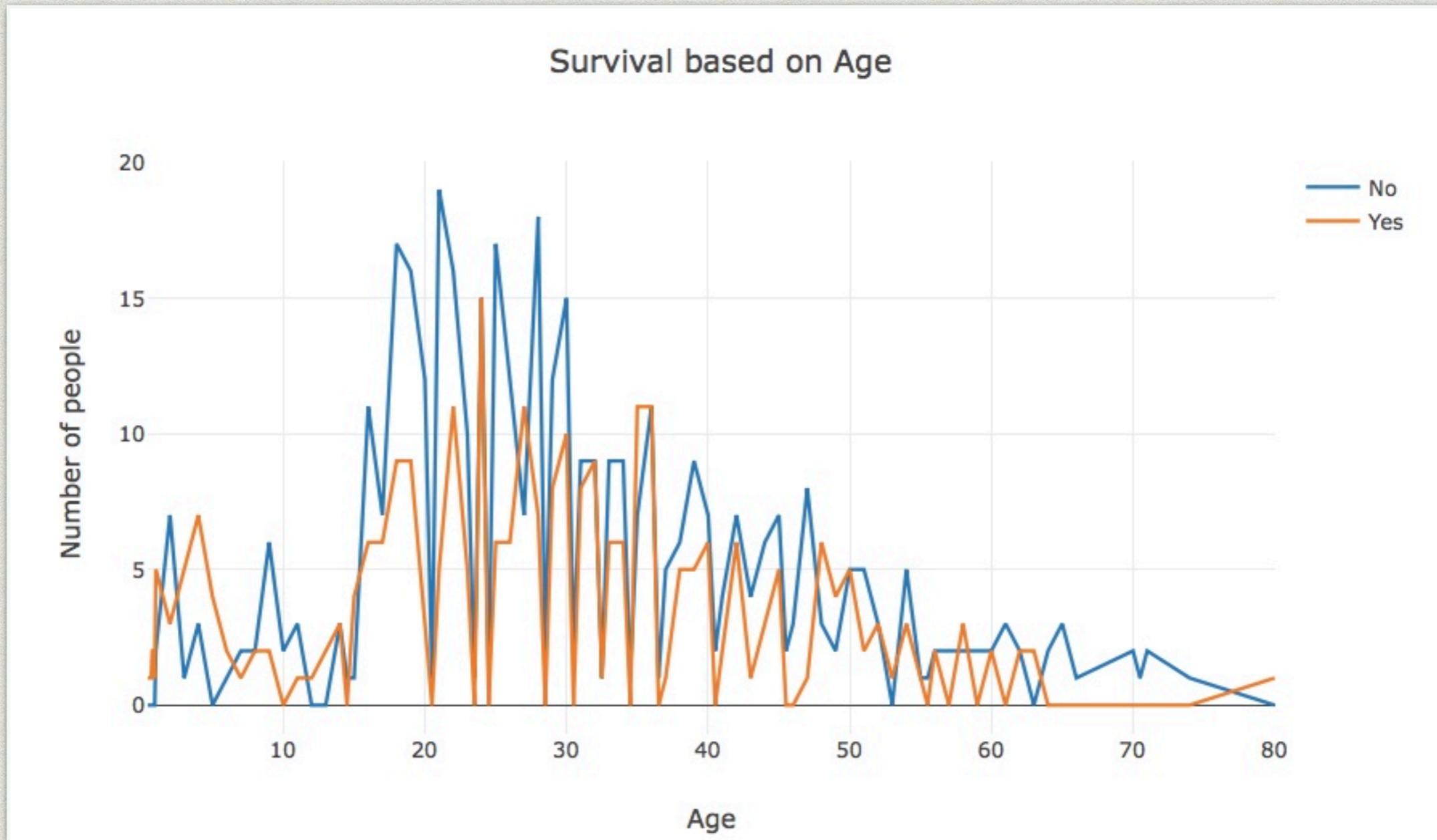
Wykres przedstawiający zależność pomiędzy przeżyciem katastrofy a przynależnością do danej klasy



# Wiek

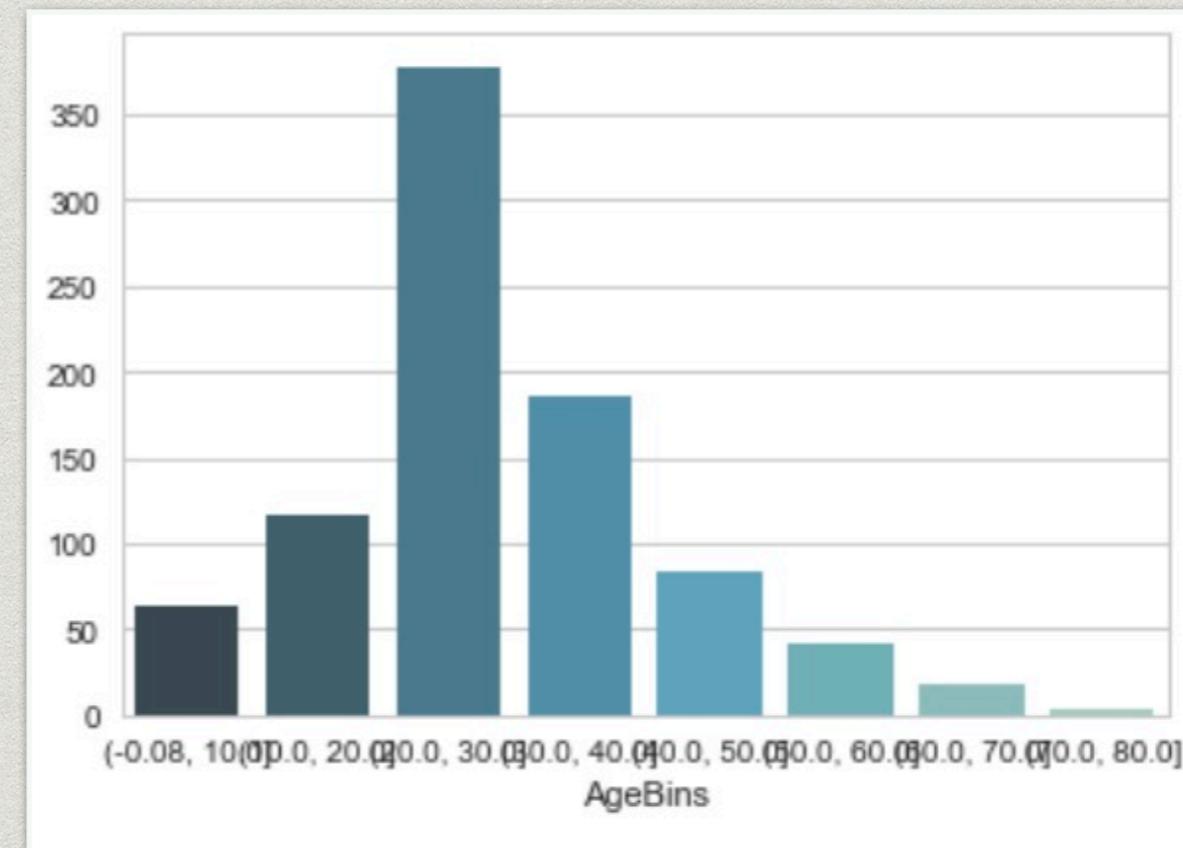
1. W tym przypadku mamy do czynienia z wybrakowanymi zmiennymi. W celu wykonania modelu prognozującego musimy sobie jakoś poradzić z tak wybrakowanymi danymi. Powszechnymi sposobami na poradzenie sobie z wybrakowanymi danymi jest
  - usunięcie wierszy, które nie posiadają danych (co jest raczej dość drastycznym posunięciem)
  - uzupełnienie wybrakowanych wartości średnią, medianą, najczęściej powtarzającą się wartością itd.
2. W naszym przypadku mamy do czynienia z 177 wierszami, które nie mają informacji o wieku. W tym przypadku napewno nie możemy sobie pozwolić na usunięcie tych obserwacji. Działanie jakie zostało tutaj podjęte to uzupełnienie tych obserwacji medianą oparta na płci oraz klasie.
3. Na podstawie wykresu widać, że dzieci miały większe szanse przeżycia.

## Zależność pomiędzy wiekiem a przeżyciem katastrofy

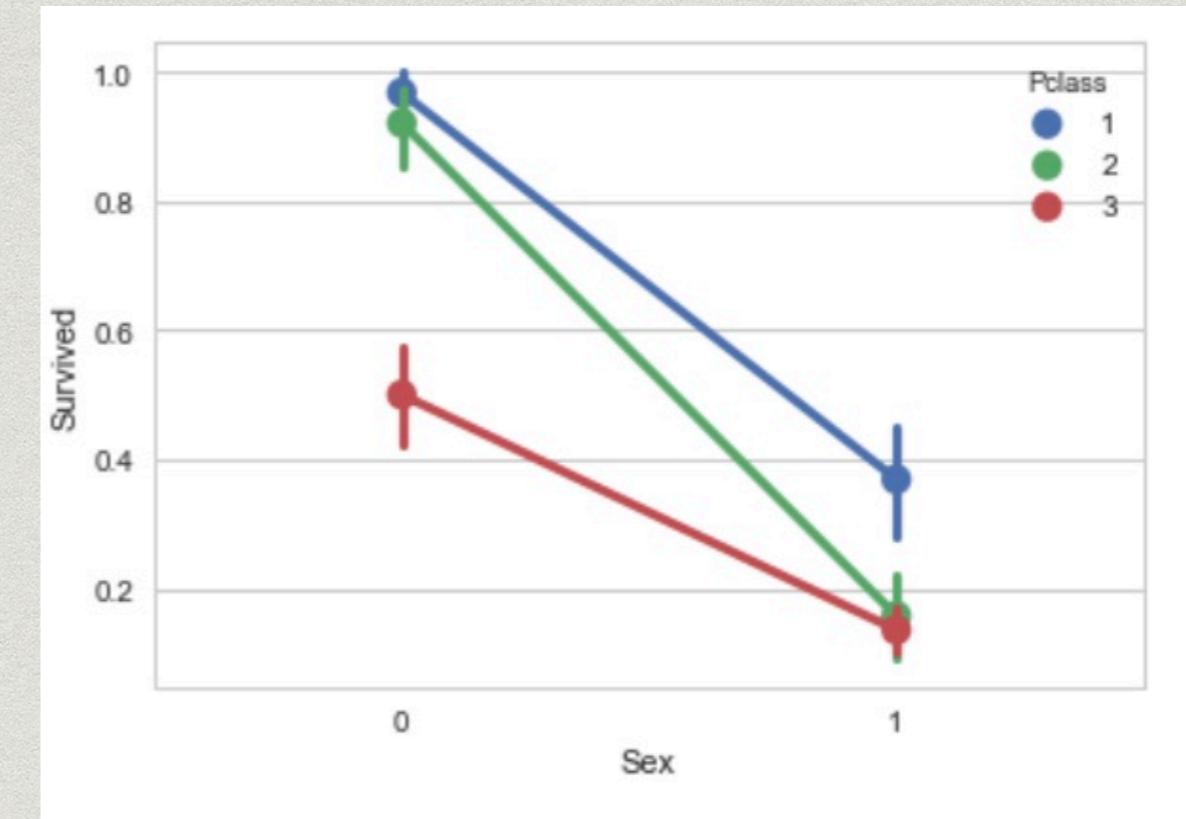
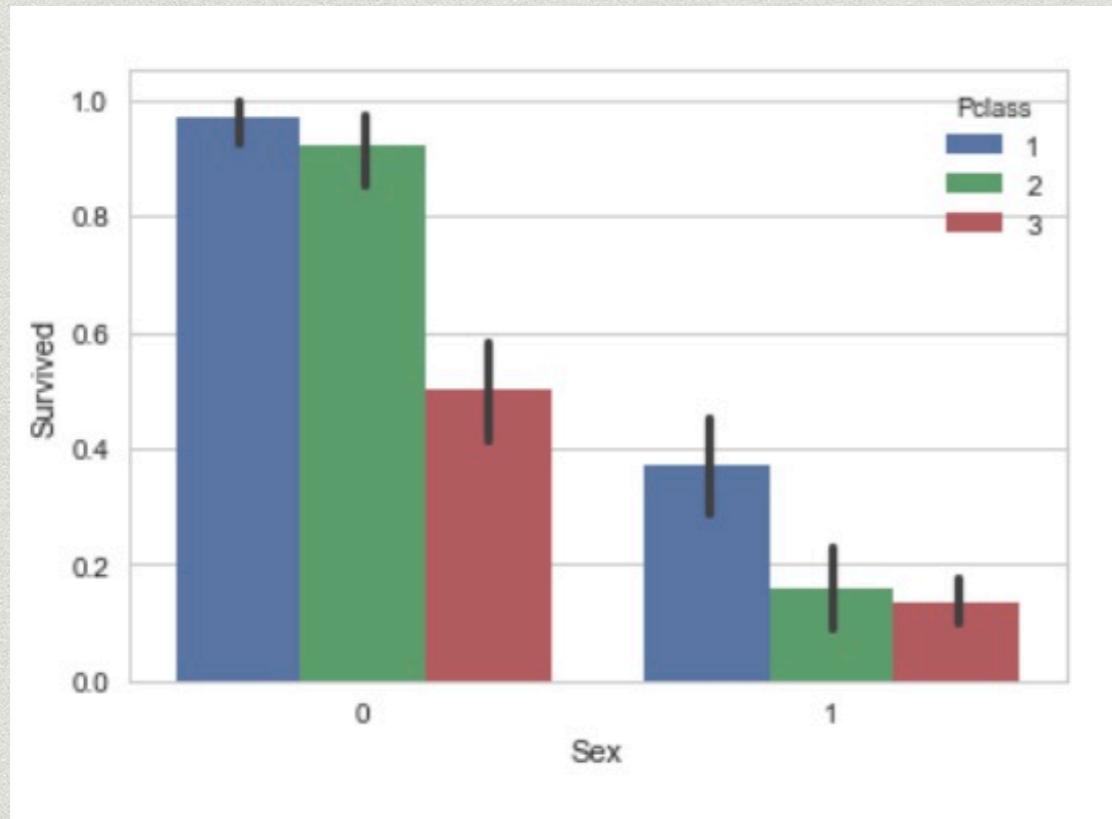


Wiemy, że wiek miał silny wpływ na przeżycie katastrofy. W celu dalszej analizy oraz sprawniejszego działania algorytmów uczenia maszynowego zamienimy zmienną wiek na przedziały wiekowe

|   | AgeBins       | Survived |
|---|---------------|----------|
| 6 | (60.0, 70.0]  | 0.222222 |
| 7 | (70.0, 80.0]  | 0.250000 |
| 2 | (20.0, 30.0]  | 0.322751 |
| 1 | (10.0, 20.0]  | 0.379310 |
| 4 | (40.0, 50.0]  | 0.392857 |
| 5 | (50.0, 60.0]  | 0.404762 |
| 3 | (30.0, 40.0]  | 0.448649 |
| 0 | (-0.08, 10.0] | 0.593750 |

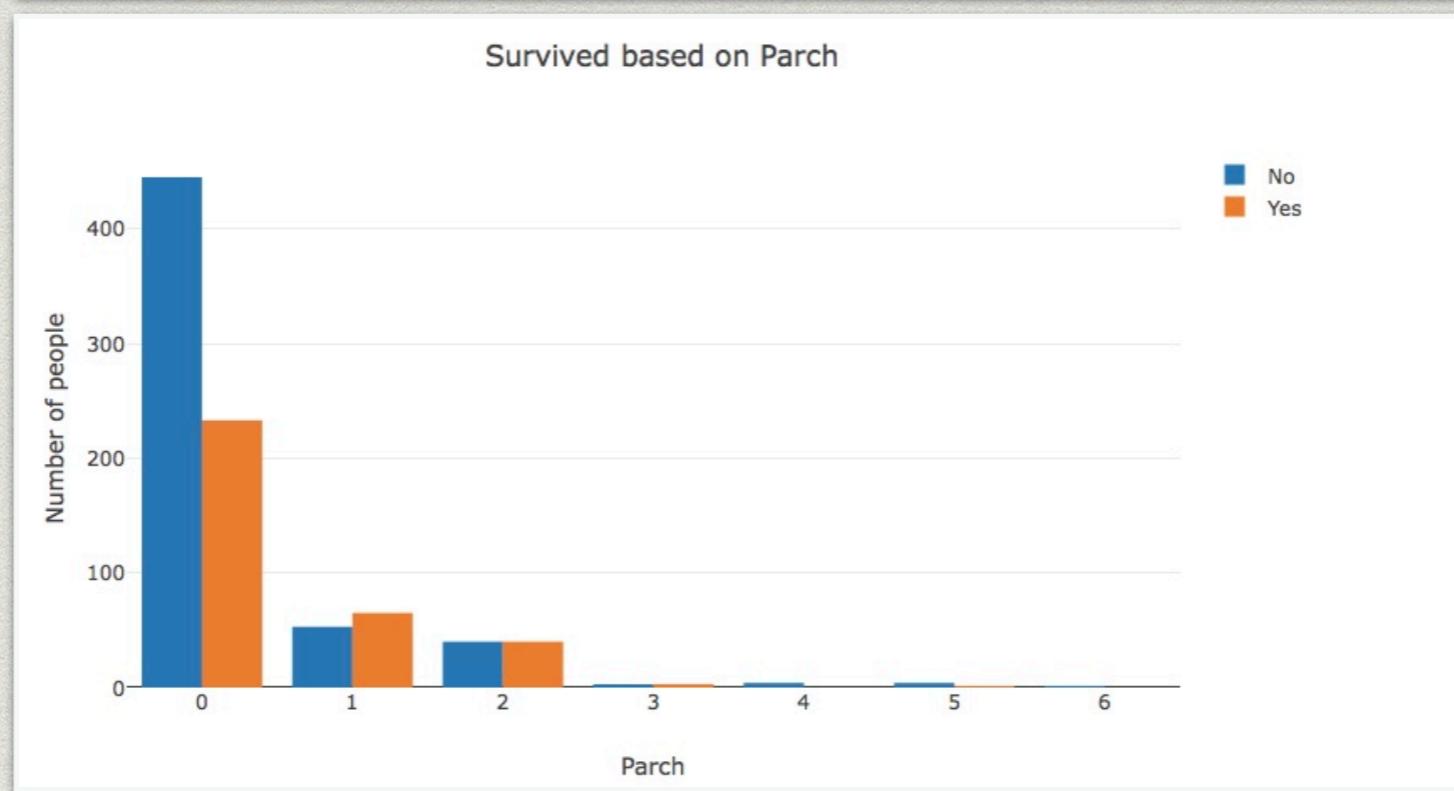
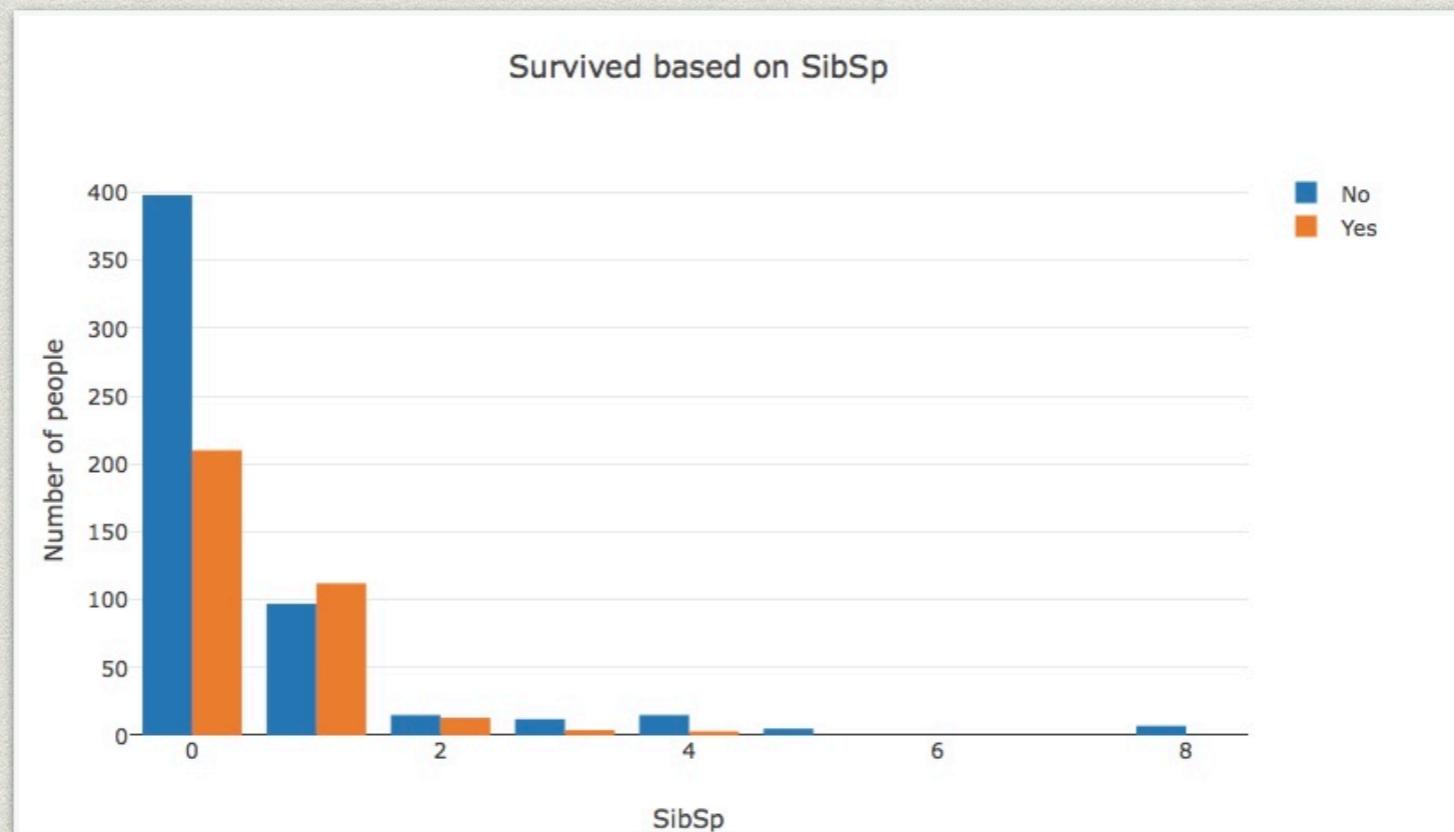


# Związek pomiędzy płcią, klasą a przeżyciem

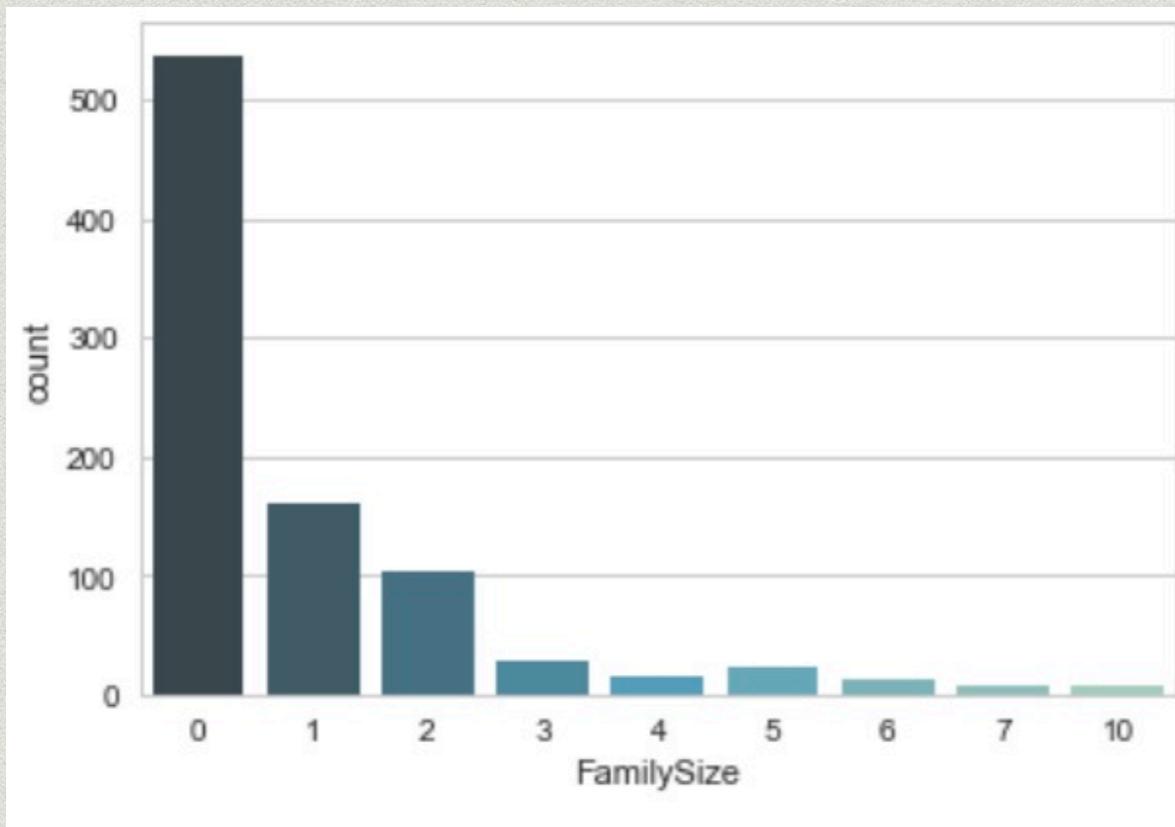


Na podstawie powyższych wykresów widzimy, że kobiety ze wszystkich klas miały większą szansę przeżycia.

# Rodzina (liczba rodzeństwa/kuzynów + liczba dzieci/rodziców na pokładzie)



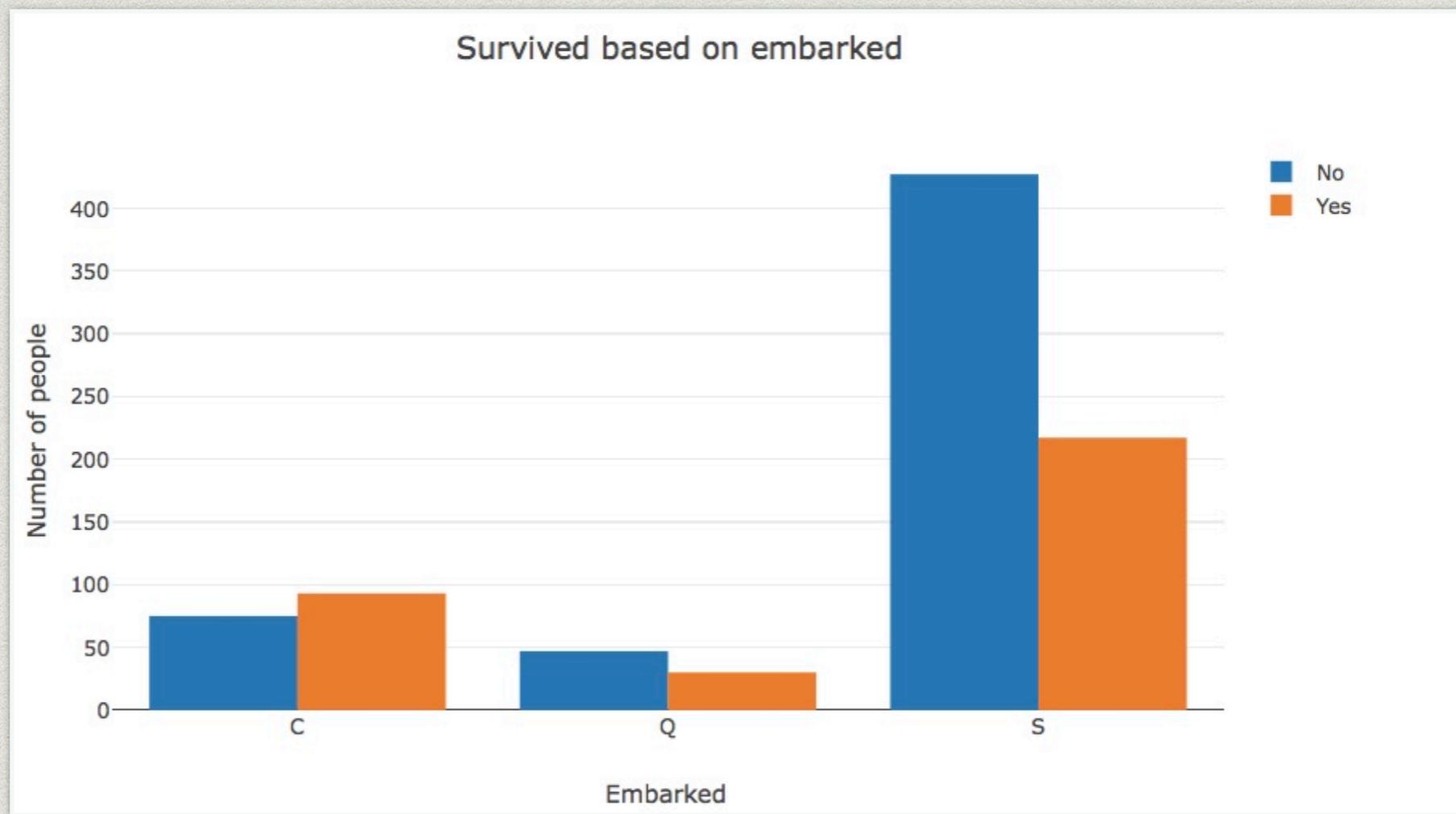
Zarówno w zmiennej **SibSp** jak i **Parch** nie mamy do czynienia z wybrakowanymi danymi. W przypadku tych dwóch zmiennych śmiało możemy wysunąć hipotezę, że przeżycie zależało od tego czy się podróżowało z kimś spokrewnionym czy też nie. W związku z tym zsumowaliśmy dwie zmienne w jedną, tworząc nową zmienną **FamilySize**, która jest łączną liczbą osób spokrewnionych ze sobą.



Na podstawie wykresu widać, że większość osób podróżowała pojedynczo

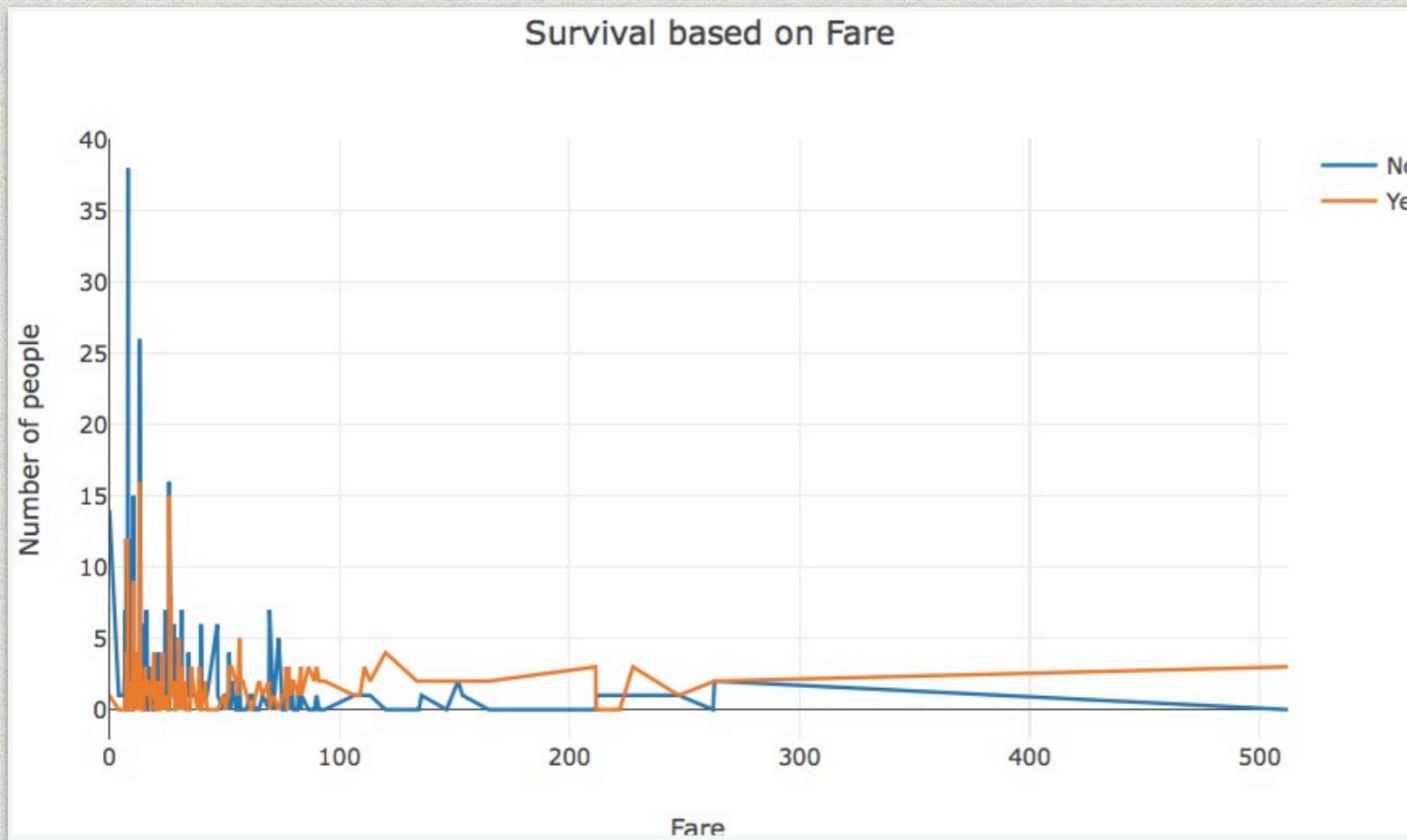
# Port

- 2 wartości brakujące, które zastąpimy średnią z pozostałych
- zmienne kategoryczne (S, Q, C - odpowiadające portą z którego pasażerowie wypływali) brakujące, które zastąpimy średnią z pozostałych



# Opłata za bilet

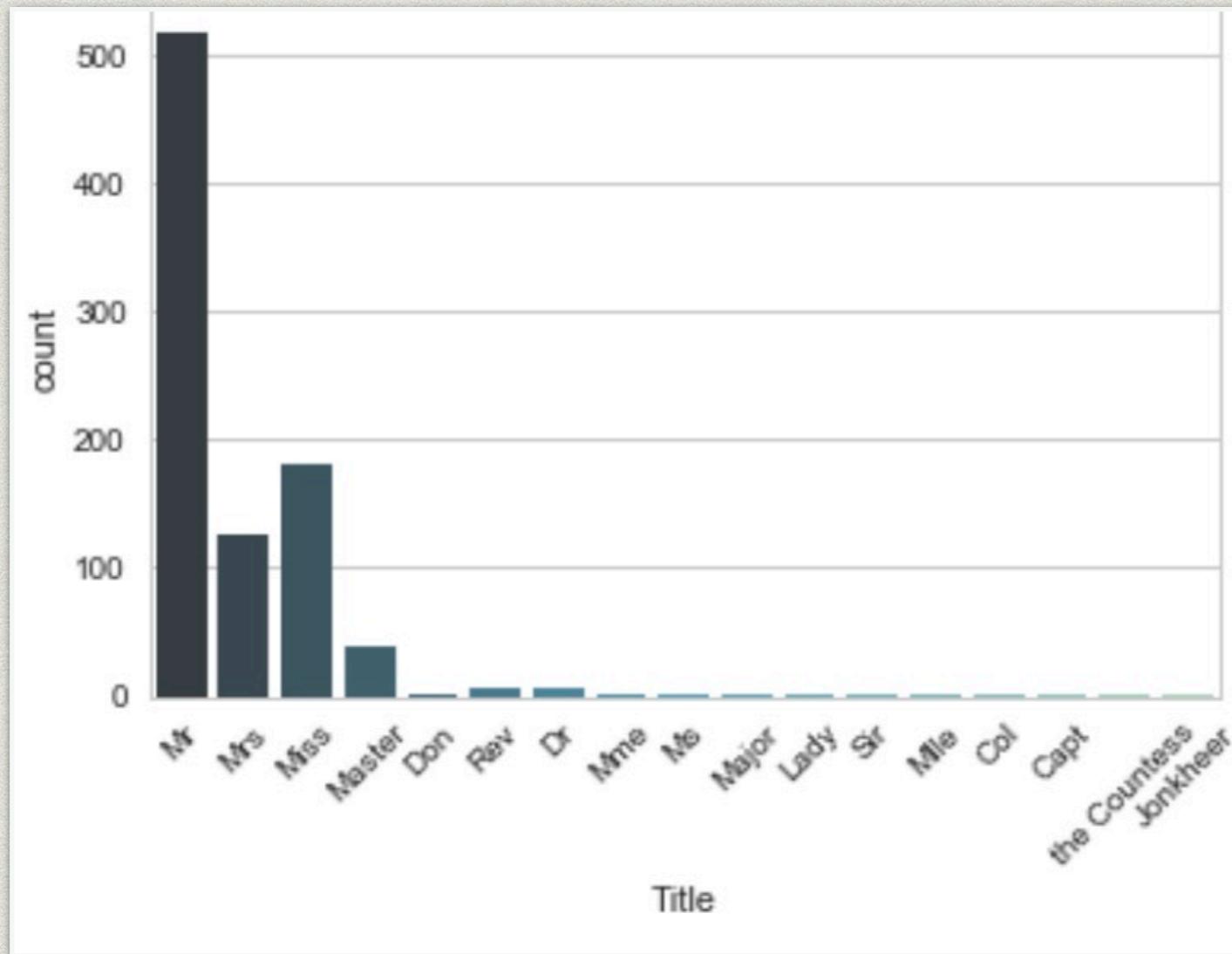
- nie mamy żadnej brakującej wartości
- podobnie jak z wiekiem, stworzymy zmienną wartość, która będzie zawierała przedziały cenowe opłat



|   | FareBins          | Survived |
|---|-------------------|----------|
| 1 | (7.854, 10.5]     | 0.201087 |
| 0 | (-0.001, 7.854]   | 0.217877 |
| 2 | (10.5, 21.679]    | 0.424419 |
| 3 | (21.679, 39.688]  | 0.444444 |
| 4 | (39.688, 512.329] | 0.642045 |

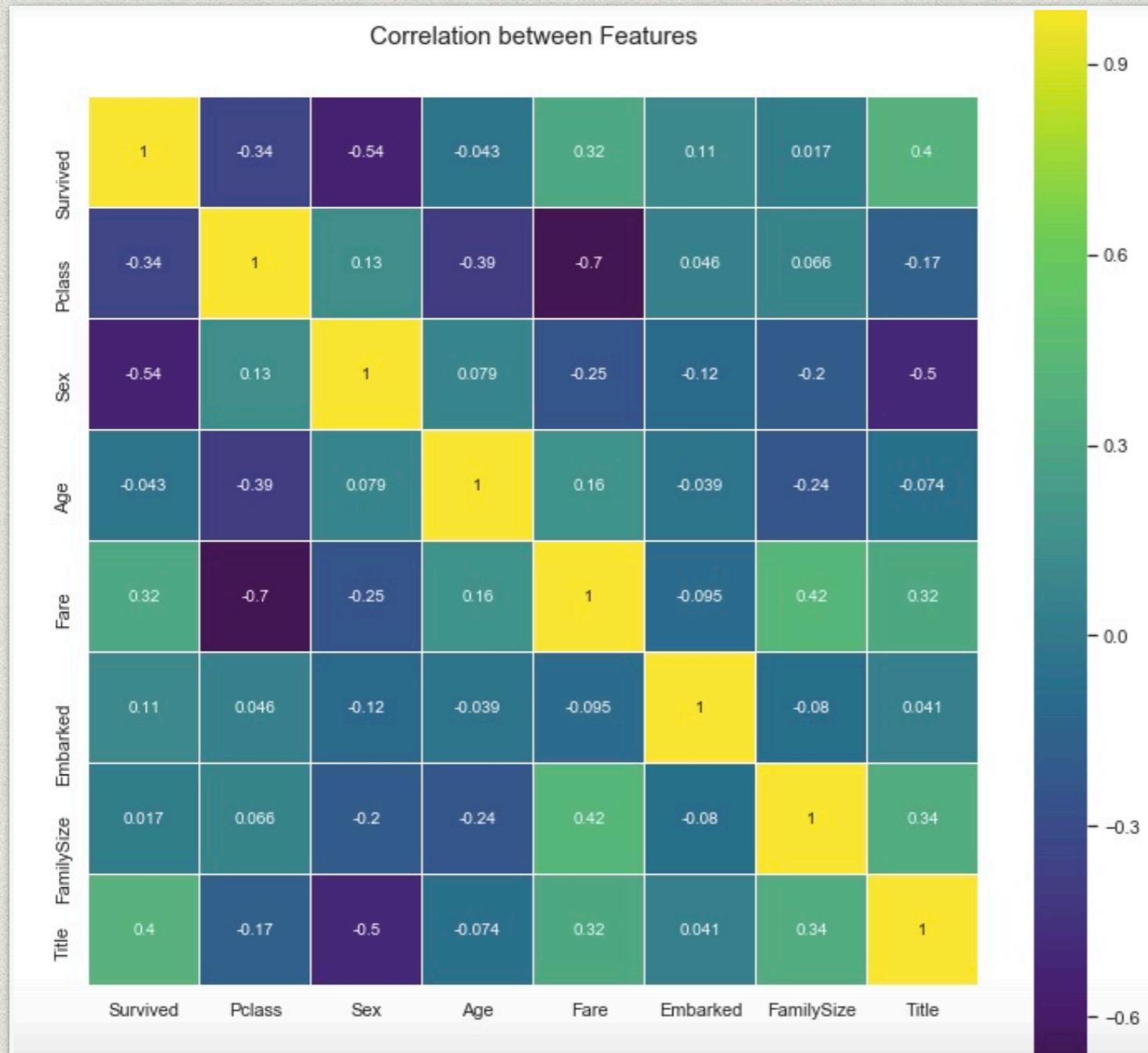
# Tytuł

Samo imię oraz nazwisko większego znaczenia w predykcji przeżycia nie ma. Natomiast tytuł jak najbardziej wpływa na ocalenie z katastrofy. Dlatego stworzymy sobie dodatkową zmienną, która będzie zawierała informacje o tytule danej osoby.



|   | Title  | Survived |
|---|--------|----------|
| 4 | Mrs    | 0.792000 |
| 2 | Miss   | 0.705882 |
| 1 | Master | 0.575000 |
| 0 | Lamped | 0.318182 |
| 3 | Mr     | 0.156673 |

# Macierz korelacji pomiędzy zmiennymi



Z macierzy korelacji widać, że największy wpływ na zmienną Survival mają płeć, przynależność do klasy, opłata za bilet oraz tytuł.

# Uczenie maszynowe

W celu predykcji przeżycia pasażerów Titanica, użyjemy następujących modeli klasyfikujących:

1. Logistic regression
2. Support vector machine
3. Lasy losowe
4. KNN
5. Naive Bayes
6. Drzewo decyzyjne
7. Perceptron
8. Stochastic Gradient Descent

# Podział zbioru na część treningową i testową

```
# Split data to be used in the models
# Create matrix of features
x = train.drop('Survived', axis = 1) # grabs everything else but 'Survived'

# Create target variable
y = train['Survived'] # y is the column we're trying to predict

# Use x and y variables to split the training data into train and test set
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = .20, random_state = 101)
```

Porównanie modeli będziemy przeprowadzać, za pomocą  
**roc\_auc\_score, accuracy, false positive and true positive rate**

# Logistic Regression

```
Classification Report:  
precision    recall   f1-score   support  
  
      0       0.78      0.91      0.84      99  
      1       0.86      0.68      0.76      80  
  
avg / total       0.81      0.80      0.80     179  
  
-----  
ROC : 0.871 %
```

# SVM

```
Accuracy for linear SVM is 0.7988826815642458  
-----  
Classification Report  
precision    recall   f1-score   support  
  
      0       0.78      0.88      0.83      99  
      1       0.82      0.70      0.76      80  
  
avg / total       0.80      0.80      0.80     179  
  
-----  
ROC : 0.869 %
```

## Drzewo losowe

```
The accuracy of the Decision Tree is 0.7988826815642458
```

```
-----  
Classification Report
```

|             | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0           | 0.77      | 0.90   | 0.83     | 99      |
| 1           | 0.84      | 0.68   | 0.75     | 80      |
| avg / total | 0.81      | 0.80   | 0.80     | 179     |

```
-----  
ROC : 0.798 %
```

## KNN

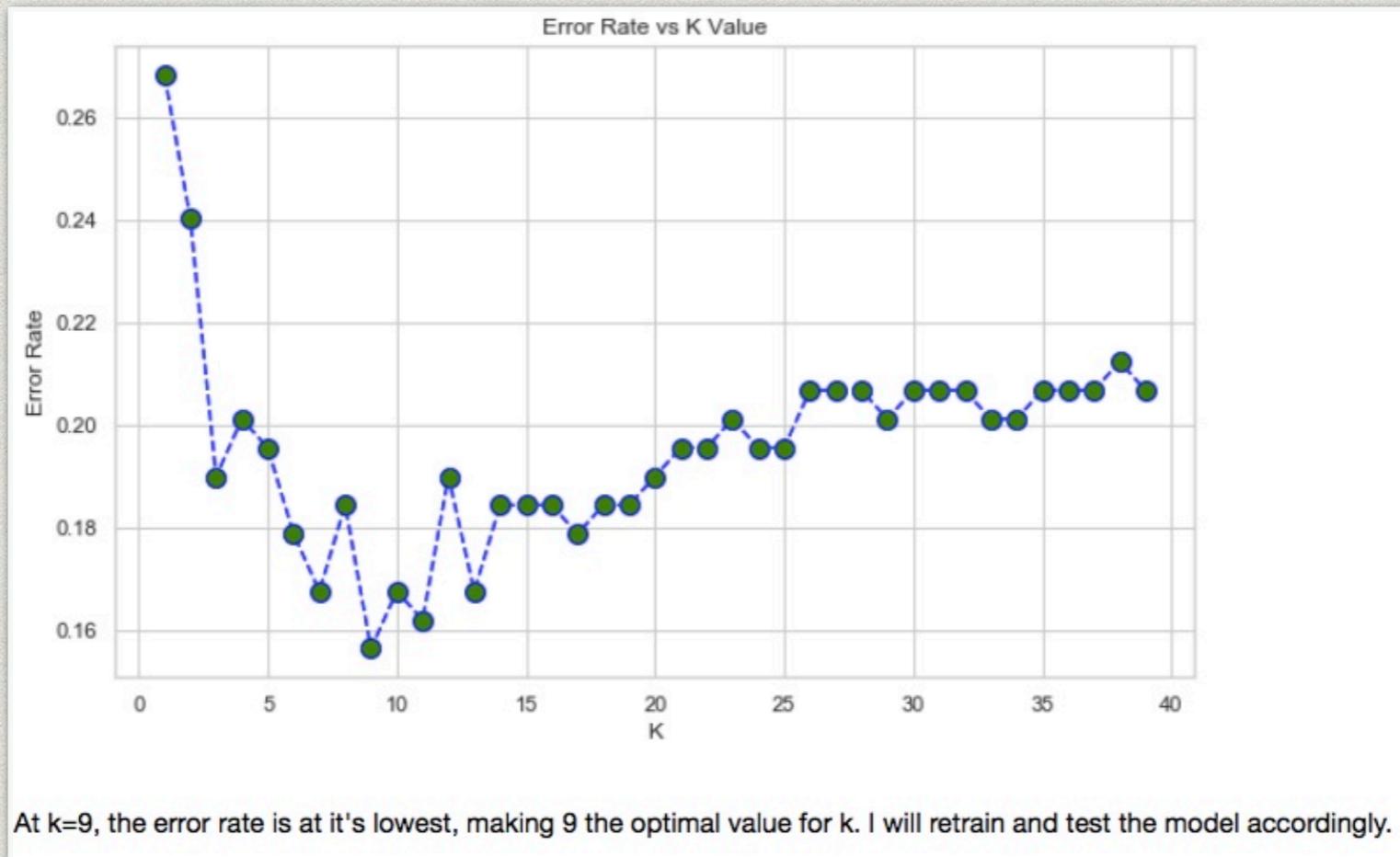
```
The accuracy of the KNN is 0.8044692737430168
```

```
-----  
Classification Report
```

|             | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0           | 0.78      | 0.91   | 0.84     | 99      |
| 1           | 0.86      | 0.68   | 0.76     | 80      |
| avg / total | 0.81      | 0.80   | 0.80     | 179     |

```
-----  
ROC : 0.870 %
```

# KNN z najlepszym parametrem



```
K-Nearest Neighbors(KNN)
k = 9
-----
The accuracy of the KNN is 0.8435754189944135
-----
Classification Report
              precision    recall   f1-score   support
          0       0.80      0.96      0.87      99
          1       0.93      0.70      0.80      80
avg / total       0.86      0.84      0.84     179
-----
ROC : 0.882 %
```

# Gaussian Naive Bayes

```
The accuracy of the NaiveBayes is 0.8044692737430168
```

---

```
Classification Report
```

|             | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0           | 0.81      | 0.84   | 0.83     | 99      |
| 1           | 0.79      | 0.76   | 0.78     | 80      |
| avg / total | 0.80      | 0.80   | 0.80     | 179     |

---

```
ROC : 0.859 %
```

# Las losowy

```
The accuracy of the Random Forests is 0.8156424581005587
```

---

```
Classification Report
```

|             | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0           | 0.80      | 0.89   | 0.84     | 99      |
| 1           | 0.84      | 0.72   | 0.78     | 80      |
| avg / total | 0.82      | 0.82   | 0.81     | 179     |

---

```
ROC : 0.867 %
```

## Perceptron

```
The accuracy of the Perceptron is 0.6871508379888268
```

```
-----  
Classification Report
```

|             | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0           | 0.73      | 0.69   | 0.71     | 99      |
| 1           | 0.64      | 0.69   | 0.66     | 80      |
| avg / total | 0.69      | 0.69   | 0.69     | 179     |

```
-----  
ROC : 0.746 %
```

## Stochastic Gradient Descent

```
The accuracy of the SGD is 0.6480446927374302
```

```
-----  
Classification Report
```

|             | precision | recall | f1-score | support |
|-------------|-----------|--------|----------|---------|
| 0           | 0.63      | 0.87   | 0.73     | 99      |
| 1           | 0.70      | 0.38   | 0.49     | 80      |
| avg / total | 0.66      | 0.65   | 0.62     | 179     |

```
-----  
ROC : 0.697 %
```

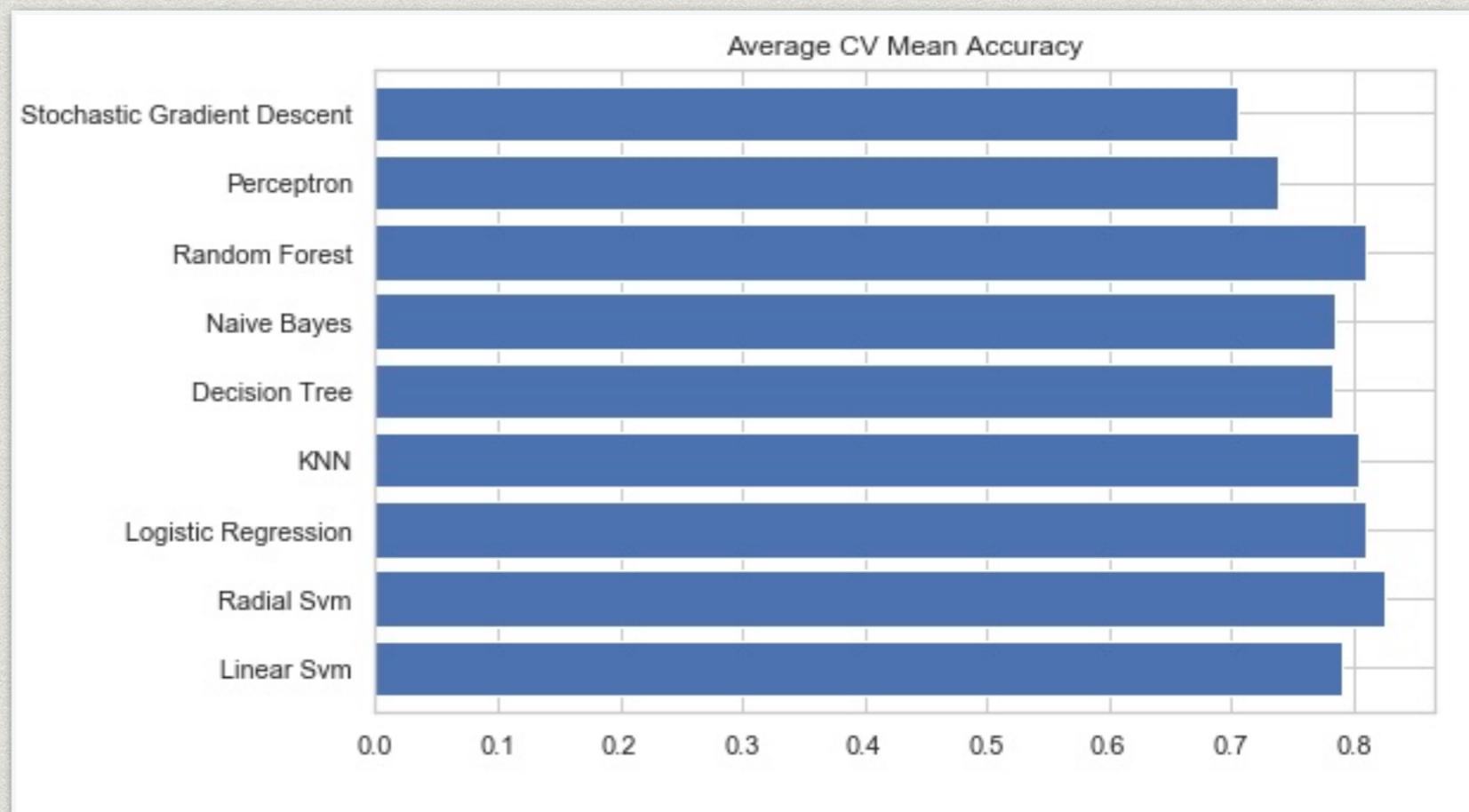
# Ewoluowanie modelu predykcyjnego

|   | Model                       | Score    |
|---|-----------------------------|----------|
| 4 | K-Nearest Neighbours        | 0.843575 |
| 2 | Rbf SVC                     | 0.832402 |
| 8 | Stochastic Gradient Descent | 0.832402 |
| 6 | Random Forests              | 0.815642 |
| 0 | Logistic Regression         | 0.804469 |
| 5 | Gaussian Naive Bayes        | 0.804469 |
| 1 | Linear SVC                  | 0.798883 |
| 3 | Decision Tree               | 0.798883 |
| 7 | Perceptron                  | 0.687151 |

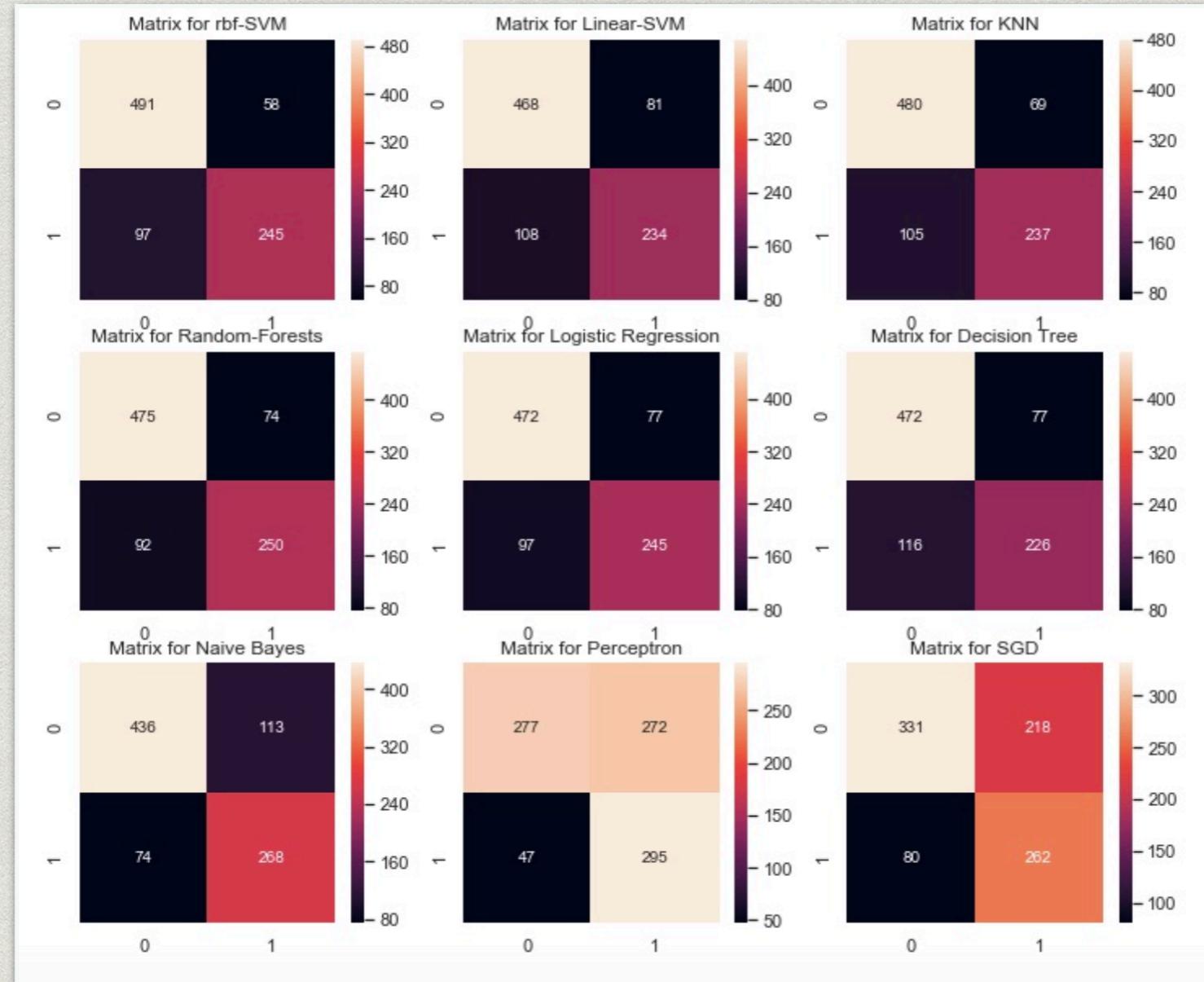
Najlepszy model okazał się **Linear SVC**, następnie **Rbf SVC** oraz **Drzewa decyzyjne**. W kolejnej części projektu do ewoluowania modelu zostało zastosowane **Cross Validation** w celu poprawieniu wyniku modelowania.

# Cross Validation

|                             | CV Mean  | Std      |
|-----------------------------|----------|----------|
| Linear Svm                  | 0.790089 | 0.029209 |
| Radial Svm                  | 0.824901 | 0.025066 |
| Logistic Regression         | 0.810351 | 0.020264 |
| KNN                         | 0.804745 | 0.018207 |
| Decision Tree               | 0.783397 | 0.017154 |
| Naive Bayes                 | 0.785701 | 0.033231 |
| Random Forest               | 0.810320 | 0.018312 |
| Perceptron                  | 0.738642 | 0.073207 |
| Stochastic Gradient Descent | 0.705926 | 0.045993 |



# Macierz pomyłek



**Rbf SVM** ze wszystkich modeli ma największą szansę na przewidzenie poprawnie zaklasyfikowanych pasażerów jako tych którzy katastrofy nie przeżyli. Natomiast **NaiveBayes**, **SGD**, **Perceptron** mają największe szanse na pasażerów, którzy przeżyli tragedię.

W kolejnym etapie projektu zajmiemy się ewoluowaniem modelu dla algorytmów takich jak:

1. SVM
2. Las losowy

# Grid Search, czyli poszukiwanie najlepszego parametru dla danego modelu

## SVM

```
from sklearn.model_selection import GridSearchCV
C=[0.05,0.1,0.2,0.3,0.25,0.4,0.5,0.6,0.7,0.8,0.9,1]
gamma=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0]
kernel=['rbf','linear']
hyper={'kernel':kernel,'C':C,'gamma':gamma}
gd=GridSearchCV(estimator=svm.SVC(),param_grid=hyper,verbose=True)
gd.fit(X,Y)
print(gd.best_score_)
print(gd.best_estimator_)
```

## Lasy losowe

```
n_estimators=range(100,1000,100)
hyper={'n_estimators':n_estimators}
gd=GridSearchCV(estimator=RandomForestClassifier(random_state=0),param_grid=hyper,verbose=True)
gd.fit(X,Y)
print(gd.best_score_)
print(gd.best_estimator_)
```

Najlepszy wynik 83.05 % dał nam dla Rbf-SVM parametr C=0.4 oraz gamma=0.3.

Natomiast dla Lasu Losowego dostaliśmy wynik 80.8% z parametrem n\_estimators=600.

# **Metody ensemblingowe**, czyli jak z paru prostych modeli zbudować model o większej dokładności

Metody użyte w projekcie:

1. Voting Classifier
2. Bagging
3. Boosting

# Voting Classifier, czyli kombinacja modeli z ich uśrednieniem

Modele, które wykorzystaliśmy to:

- KNN (z n=7)
- Rbf SVM (z C=0.4, gamma=0.3)
- Las losowy (n\_estimators=500)
- Logistic Regression (C=0.05)
- Drzewo decyzyjne
- Linear SVM

Dokładność jaką uzyskaliśmy dla ensemblerowanego modelu to: **83,24%**

Dokładność przy użyciu Cross Validation to: **82,04%**

# Bagging i boosting dla drzewa decyzyjnego

## Bagging:

Stosując metodę baggingu dla drzewa decyzyjnego otrzymaliśmy dokładność równą: **80,44%**.

Dokładność przy użyciu Cross Validation to: **80,47%**.

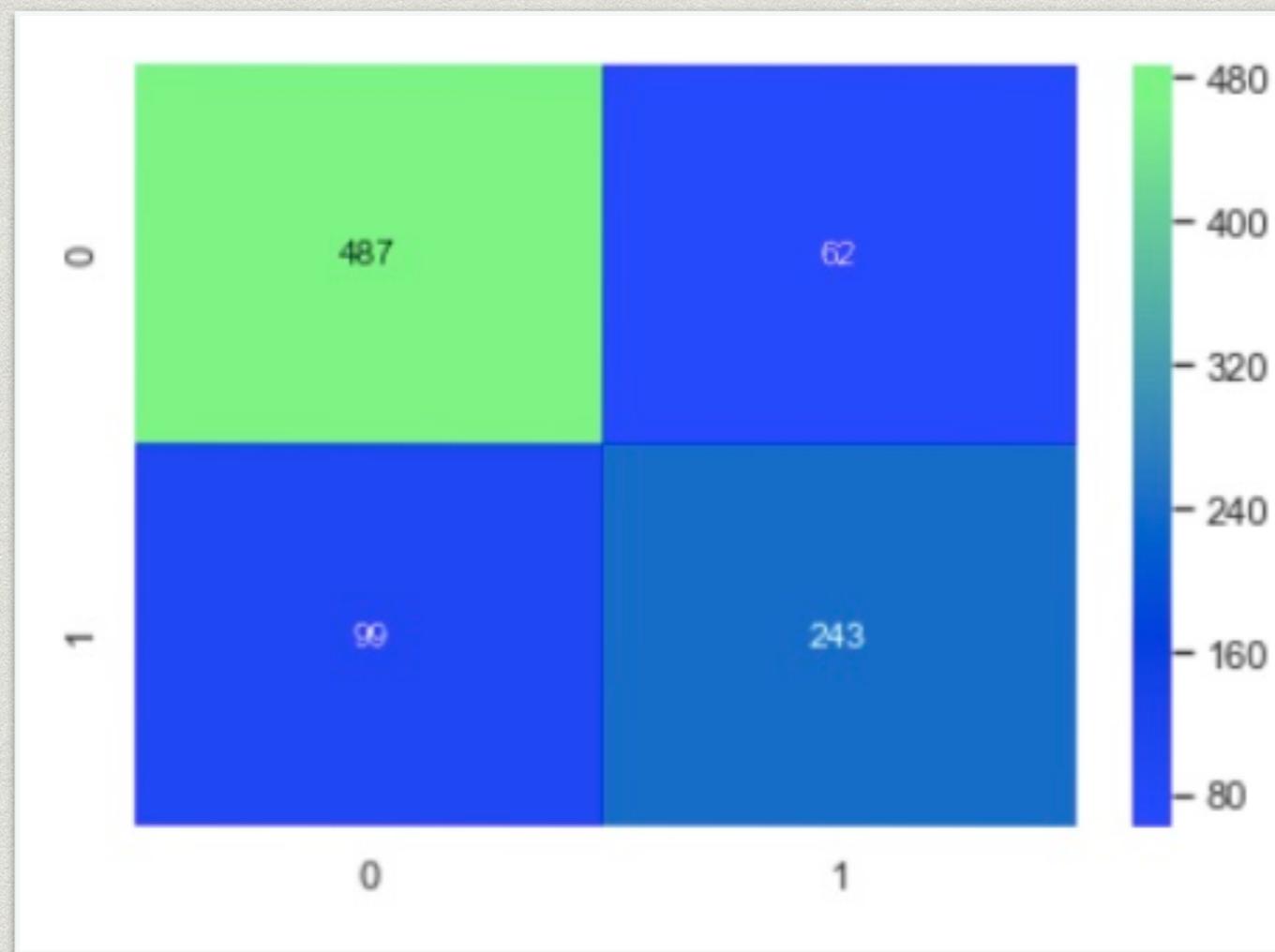
## Boosting:

**AdaBoost:** wynik dla CV to **82,38%**.

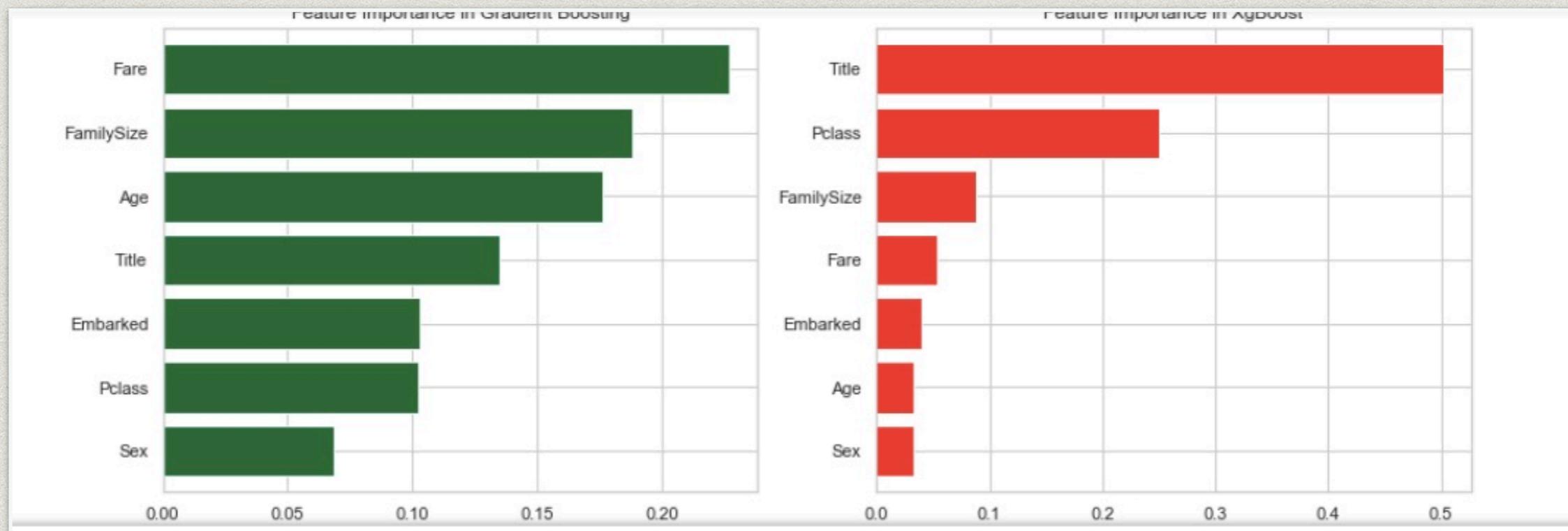
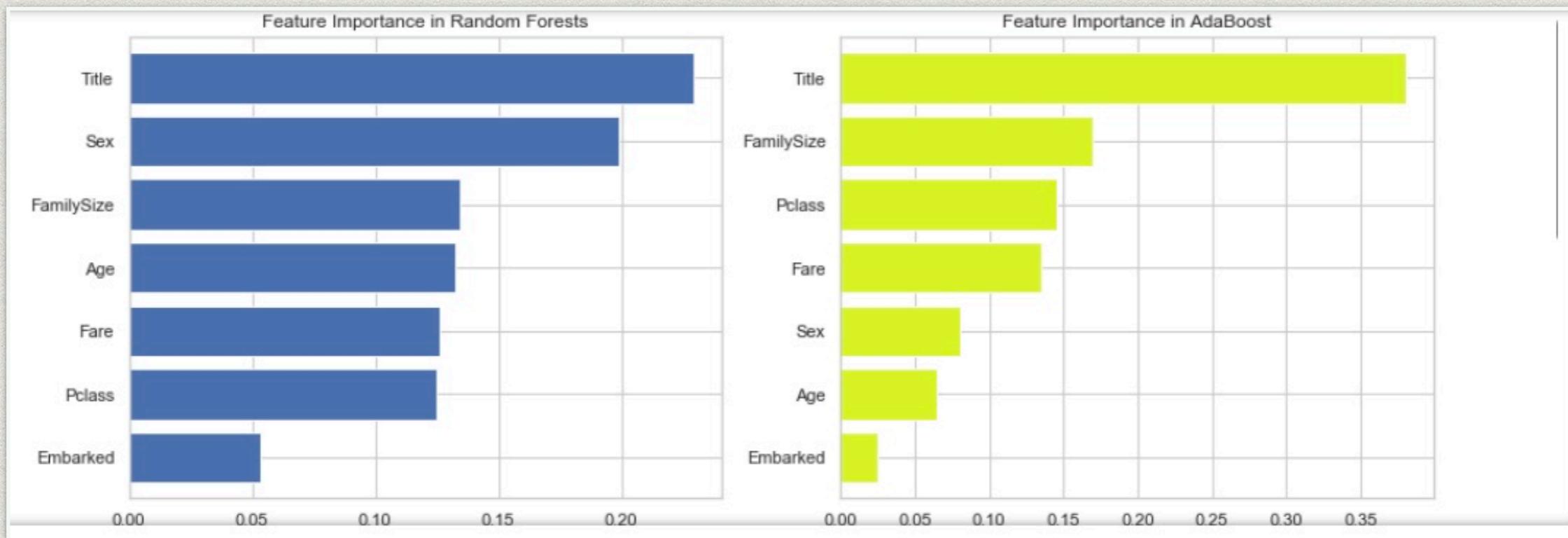
**SGD:** wynik dla CV **81,26%**.

**XGBoost:** wynik to **81,6%**.

# Macierz pomyłek dla najlepszego modelu z boostingiem (AdaBoost)



# Istotność atrybutów



# Obserwacje i wnioski z analizy predykcji przeżycia pasażerów Titanica:

1. Pewne atrybuty są wspólne dla **lasu losowego, AdaBoost, Gradient Boost, XGBoost**. Są to m.in: opłata za bilet, przynależność do klasy, rozmiar rodziny. Czyli czynniki, które od samego początku wydawały się mieć istotną rolę w predykcji.
2. Płeć dla modelu nie ma większego znaczenia, co jest bardzo zaskakującym faktem, ponieważ na początku zaobserwowaliśmy, że kobiety miały większe szanse przeżycia.
3. Najlepszym ze wszystkich modeli okazał się być KNN z n=9, pomimo że dla innych modeli szukaliśmy najlepszego parametru, oraz tworzyliśmy uśrednione modele.