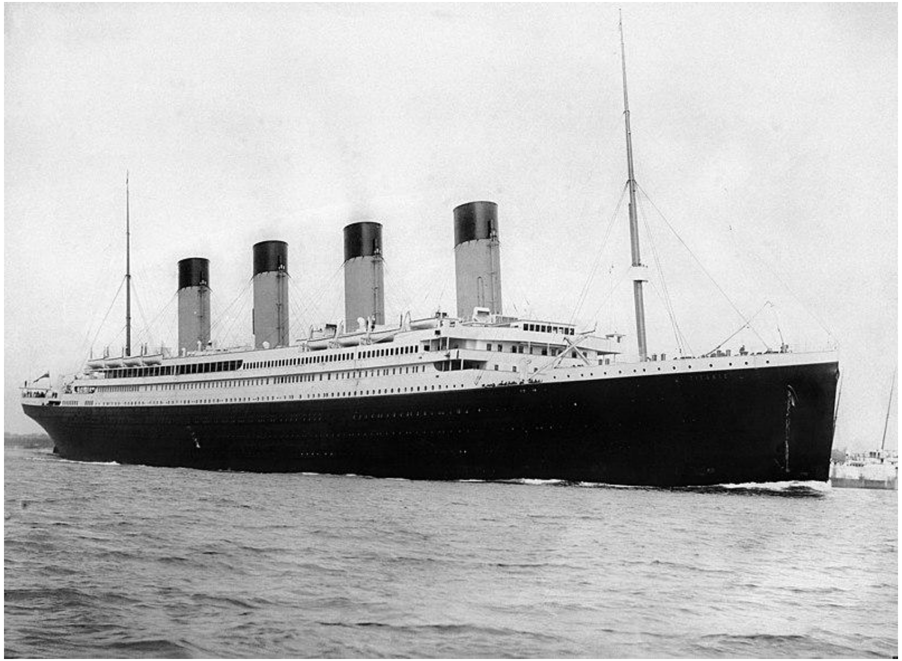
Data Mining

Assignment 1

Wojciech Nagórka, Andrii Chmutov, Vasyl Korzavatykh, Kuba Czech

Dataset



Titanic

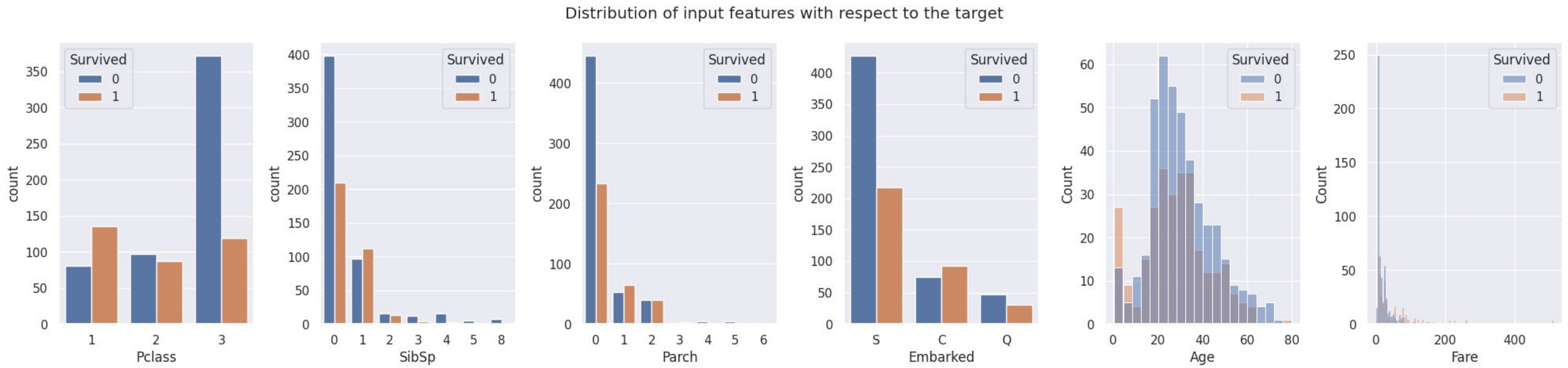
- Dataset contains information on passengers of a ship which sunk by hitting an iceberg in April 1912
- Survival is a target variable

Features in the dataset

- *PassengerID*
- Survived
- Pclass
- *Lname*
- Name
- Sex
- Age
- SibSp
- Parch
- Ticket
- Fare
- Cabin
- Embarked

```
<class 'pandas.core.frame.DataFrame'>
Index: 891 entries, 1 to 891
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Name      891 non-null    object
 3   Sex       891 non-null    object
 4   Age       714 non-null    float64
 5   SibSp     891 non-null    int64
 6   Parch     891 non-null    int64
 7   Ticket    891 non-null    object
 8   Fare      891 non-null    float64
 9   Cabin     204 non-null    object
 10  Embarked  889 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 83.5+ KB
```

Data analysis and problems at data



Missing values

- The first potential problem with the data
- Exist for attributes: Cabin, Age and Embarkment
- Could influence the result unpredictably

```
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

Dealing with missing attributes

Solution:

Cabin -> separate category

Age -> median

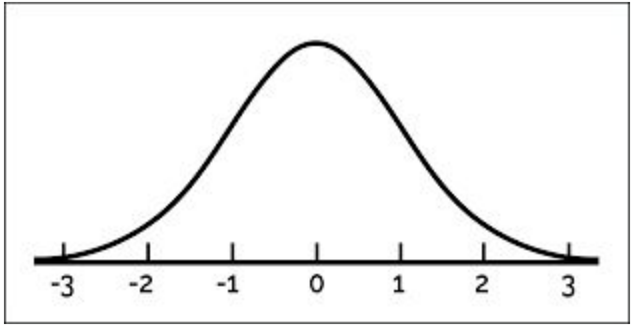
Embarkment -> mode

Continuous data

- **Age** [0.42, 80] and **Fare** [0, 512.3]
- wide range of values
- if used in raw form it may have negative effect on the results

Standardization

- changing the range of an attribute
- we used **StandardScaler()** from sklearn
- it results in reducing the impact of outliers



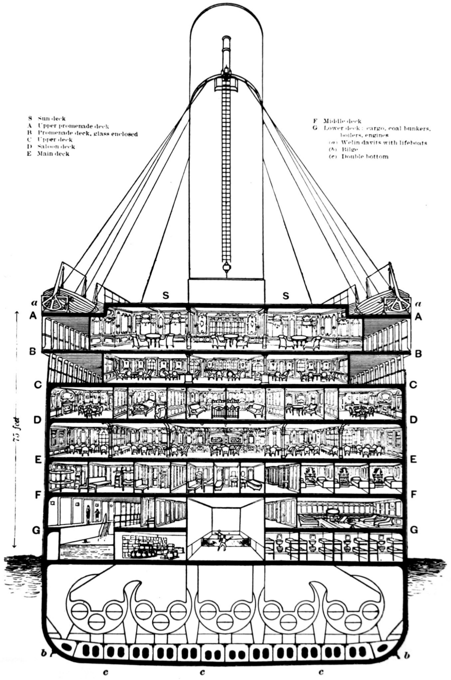
Further data analysis

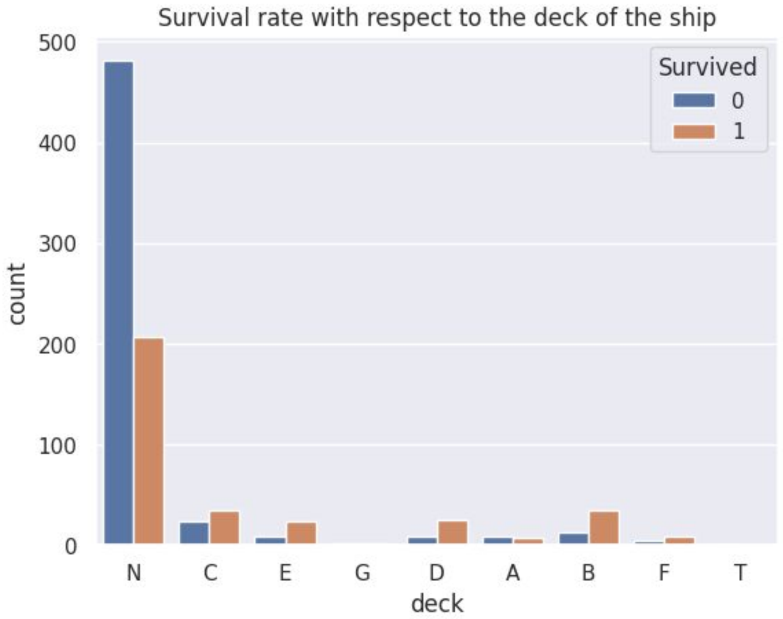
Deck vs Survival

```
data['Cabin'].unique()
```

```
array([nan, 'C85', 'C123', 'E46', 'G6', 'C103', 'D56', 'A6',
       'C23 C25 C27', 'B78', 'D33', 'B30', 'C52', 'B28', 'C83', 'F33',
       'F G73', 'E31', 'A5', 'D10 D12', 'D26', 'C110', 'B58 B60', 'E101',
       'F E69', 'D47', 'B86', 'F2', 'C2', 'E33', 'B19', 'A7', 'C49', 'F4',
       'A32', 'B4', 'B80', 'A31', 'D36', 'D15', 'C93', 'C78', 'D35',
       'C87', 'B77', 'E67', 'B94', 'C125', 'C99', 'C118', 'D7', 'A19',
       'B49', 'D', 'C22 C26', 'C106', 'C65', 'E36', 'C54',
       'B57 B59 B63 B66', 'C7', 'E34', 'C32', 'B18', 'C124', 'C91', 'E40',
       'T', 'C128', 'D37', 'B35', 'E50', 'C82', 'B96 B98', 'E10', 'E44',
       'A34', 'C104', 'C111', 'C92', 'E38', 'D21', 'E12', 'E63', 'A14',
       'B37', 'C30', 'D20', 'B79', 'E25', 'D46', 'B73', 'C95', 'B38',
       'B39', 'B22', 'C86', 'C70', 'A16', 'C101', 'C68', 'A10', 'E68',
       'B41', 'A20', 'D19', 'D50', 'D9', 'A23', 'B50', 'A26', 'D48',
       'E58', 'C126', 'B71', 'B51 B53 B55', 'D49', 'B5', 'B20', 'F G63',
       'C62 C64', 'E24', 'C90', 'C45', 'E8', 'B101', 'D45', 'C46', 'D30',
       'E121', 'D11', 'E77', 'F38', 'B3', 'D6', 'B82 B84', 'D17', 'A36',
       'B102', 'B69', 'E49', 'C47', 'D28', 'E17', 'A24', 'C50', 'B42',
       'C148'], dtype=object)
```

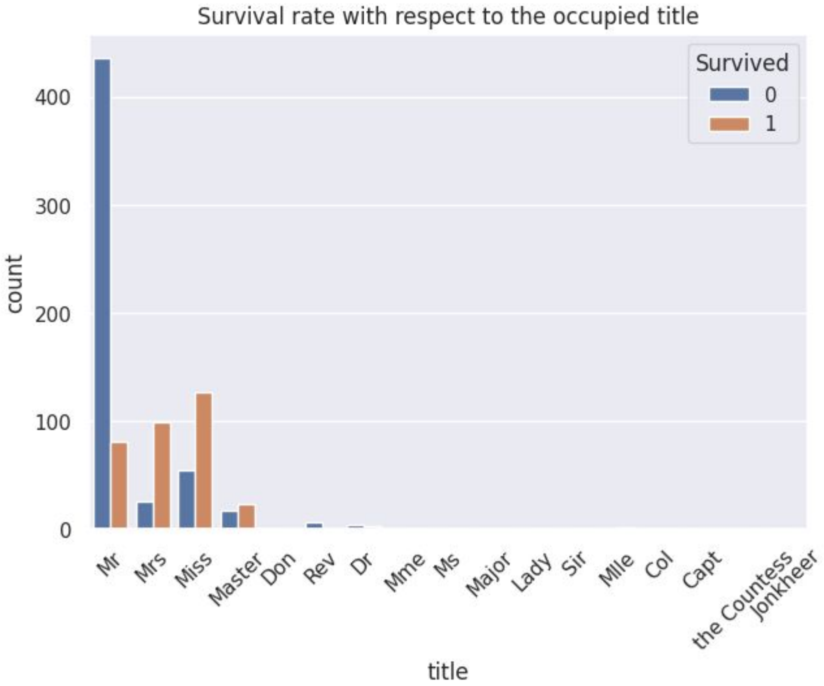
Deck vs Survival





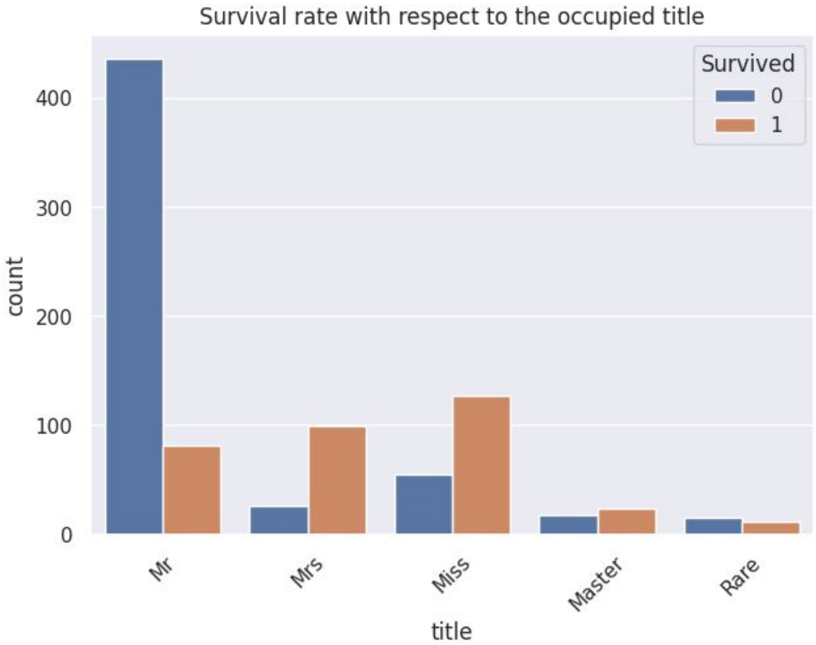
Title vs Survival

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

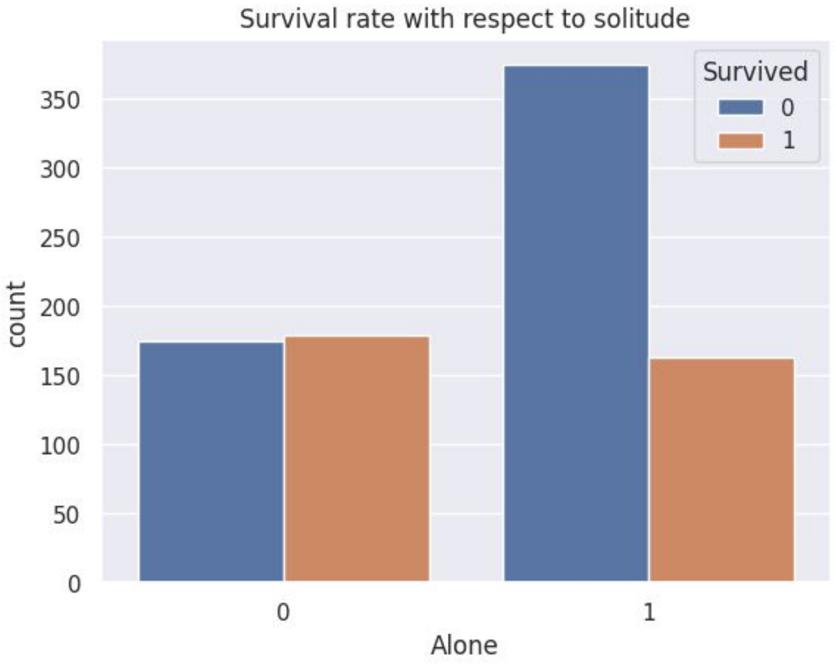


Title vs Survival

```
Name
Mr              517
Miss            182
Mrs             125
Master           40
Dr                7
Rev               6
Col               2
Mlle              2
Major             2
Ms                1
Mme               1
Don               1
Lady              1
Sir               1
Capt              1
the Countess      1
Jonkheer          1
Name: count, dtype: int64
```



FamilySize / Solitude vs Survival



Pre-processing techniques

1. **Derived attributes**
- information hidden in data
- *Title*, *Deck*, and *FamilySize* attributes
2. **Binarization**
- focus on the condition instead of the exact value
- *Alone* attribute
3. **Label encoding**
- convert categorical variable into dummy variables
- we used *get_dummies()* function from pandas

Features after pre-processing

1. **Standardized**
- Age
- Fare
2. **Derived**
- Deck
- FamilySize
3. **Binarization**
- Alone
4. **Label encoding**
- Embarked_C, Embarked_Q, Embarked_S
- Title_Master, Title_Miss, Title_Mr, Title_Mrs, Title_Rare
- Sex_male, Sex_female

Repository structure

Files

Each file in the repository has it's own purpose.

File name	Meaning
analysis.ipynb	Exploratory analysis of the features
preprocessing.py	The library itself
samples.ipynb	A set of use cases from the library
tests.ipynb	The comparison of cross-validation metric with respect to how data was prepared
train.csv	The training dataset. If you want to download or to play around with testing data as well, you should visit the following section of the competition.
improved_dataset.csv	The improved dataset

The code follows the schema of Transformer Mixin class from `scikit-learn` library to be support the Pipeline API.

Results

Before pre-processing

```
SVC
Scores: [0.68, 0.69, 0.67, 0.67, 0.7]
Mean score: 0.68

Perceptron
Scores: [0.73, 0.71, 0.66, 0.57, 0.61]
Mean score: 0.66

Random Forest
Scores: [0.84, 0.8, 0.75, 0.83, 0.82]
Mean score: 0.81
```

After pre-processing

```
SVC
Scores: [0.83, 0.8, 0.8, 0.84, 0.84]
Mean score: 0.82

Perceptron
Scores: [0.82, 0.8, 0.7, 0.78, 0.87]
Mean score: 0.79

Random Forest
Scores: [0.84, 0.81, 0.8, 0.85, 0.83]
Mean score: 0.83
```

Conclusions

Pre-processing techniques improved accuracy of classification on the dataset.

This highlights the importance of pre-processing in data analysis.