# Titanic Predictor for Survival

## AI's Daemons

Cristobal Meza - A01661792
Ricardo Campos - A01656898
Diego Zepeda - A01026512

# Methodology

1. Problem Definition
2. Objective definition, success criteria and hypothesis
4. Data Collection (ETL)
5. Data Preprocessing
6. Exploratory Data Analysis
7. Models selection
8. Models training
9. Models evaluation
10. Models optimization
11. Results
12. Conclusion

# 1. Problem Definition

The Titanic dataset is derived from the real-life tragedy of the RMS Titanic, which sank on its maiden voyage in April 1912. This dataset provides information about passengers aboard the Titanic and is often used to predict whether a passenger survived the disaster based on various attributes.

We use the dataset of the Titanic found at:
https://www.kaggle.com/competitions/titanic/overview

**The primary problem is to predict the likelihood of survival for each passenger based on their features.**

**Target class:**

**Survived**
If a passenger survived (0 = No, 1 = Yes)

**Feature classes:**

**PassengerID**
Passenger number

**Survived**
Survived or not

**PClass**
Tiket Class (1st, 2nd, 3rd).

**Name**
Passenger name

**Sex**
'Male' or 'Female'

**Age**
Age in years

**Sibsp**
Number of silbings / spouses abroad the Titanic

**Parch**
Number of parents / children aboard the Titanic

**Ticket**
Ticket number

**Fare**
Passenger fare

**Cabin**
Cabin number

**Embarked**
Port of Embarkation
(C = Cherbourg, Q = Queenstown, S = Southampton)

# 2. Objective definition, success criteria and hypothesis

## Objetives

1. **Predict with 80% Accuracy the survival rate**

2. Apply Methodology to get the top 3 highest Accuracy models and define a winner

## Success criteria

**Achieve at least 80% Accuracy in the training dataset.**

2. As comparing 3 models the winner will have:

- Highest Accuracy
- Highest f1-score

## Hypothesis

**Being a female-child should be the main cause of survival.**

This means:

- Female
- Under 18 years old

Should have the highest survival rate.

# Data Clean & Preprocessing procedure

1. **Overview**
2. **Label Count/check**
3. **Unique Count of each feature values**
4. **NaN Count per feature**
5. **Drop harder/repeated/incomplete values**
6. **NaN recheck**
7. **Distinct values recheck**
8. **Scaling**

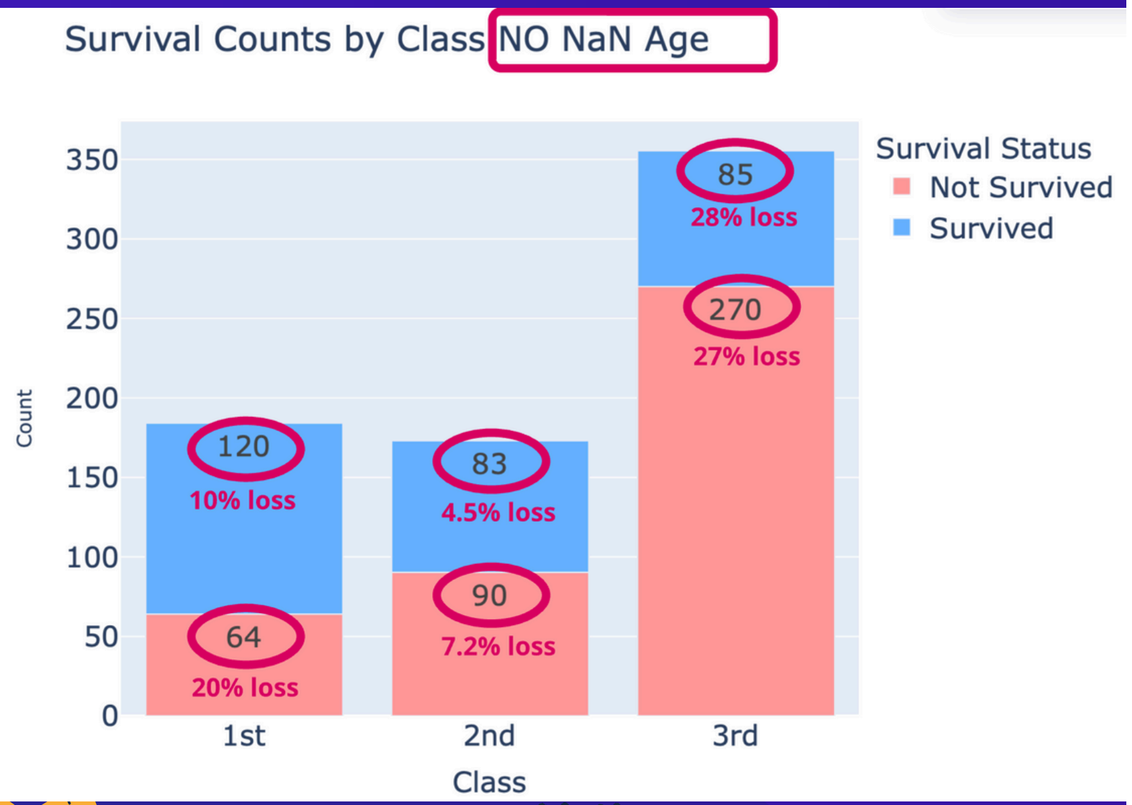
**Issue found:**

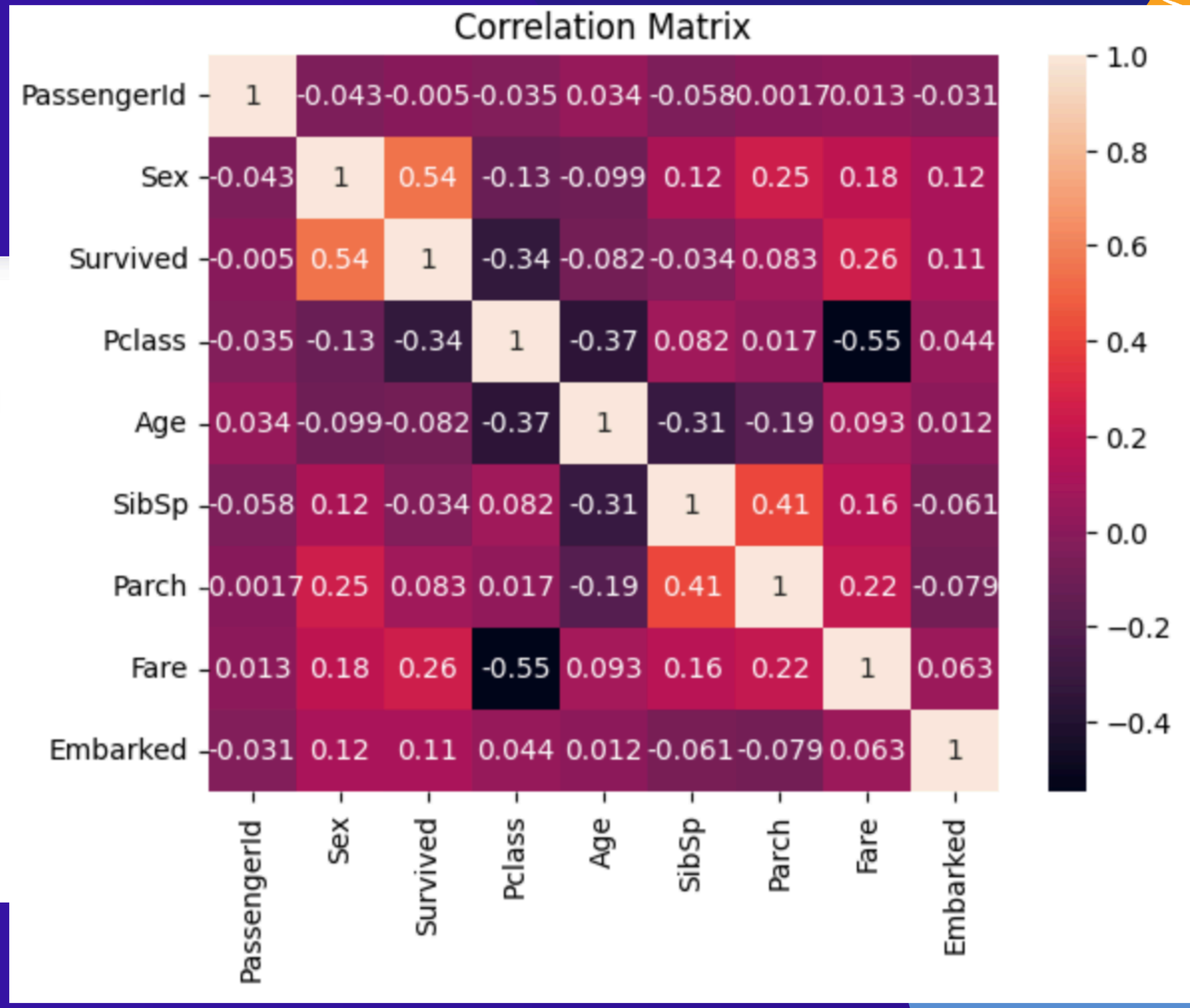
**177 ‘Age’ NaN values identified**

***How’s the impact?***

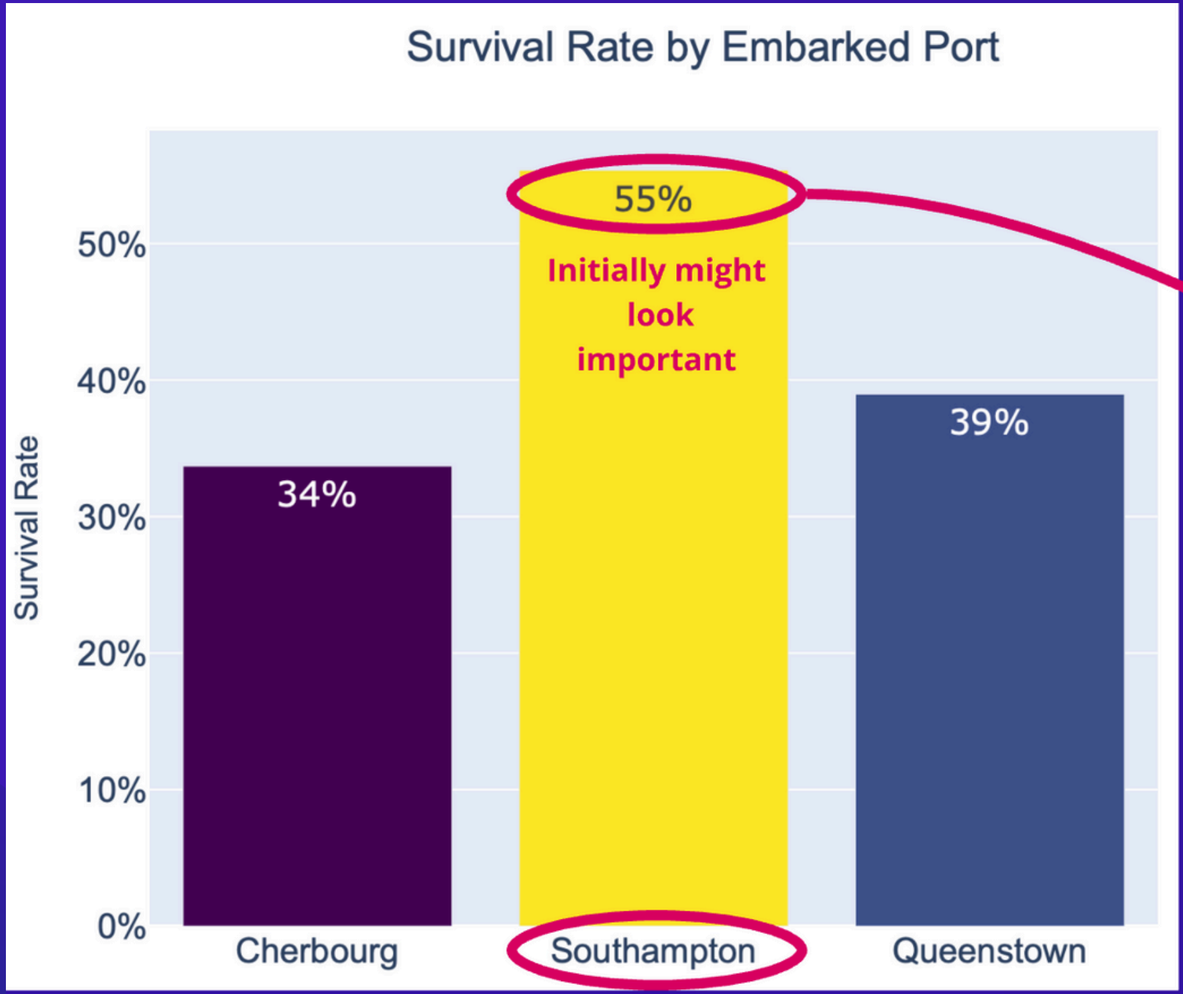
| | PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | Fare | Embarked |
|---|---|---|---|---|---|---|---|---|---|
| count | 889.000000 | 889.000000 | 889.000000 | 889.000000 | 712.000000 | 889.000000 | 889.000000 | 889.000000 | 889.000000 |
| mean | 446.000000 | 0.382452 | 2.311586 | 0.350956 | 29.642093 | 0.524184 | 0.382452 | 32.096681 | 0.362205 |
| std | 256.998173 | 0.486260 | 0.834700 | 0.477538 | 14.492933 | 1.103705 | 0.806761 | 49.697504 | 0.636157 |
| min | 1.000000 | 0.000000 | 1.000000 | 0.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 224.000000 | 0.000000 | 2.000000 | 0.000000 | 20.000000 | 0.000000 | 0.000000 | 7.895800 | 0.000000 |
| 50% | 446.000000 | 0.000000 | 3.000000 | 0.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454200 | 0.000000 |
| 75% | 668.000000 | 1.000000 | 3.000000 | 1.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000000 | 1.000000 |
| max | 891.000000 | 1.000000 | 3.000000 | 1.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329200 | 2.000000 |

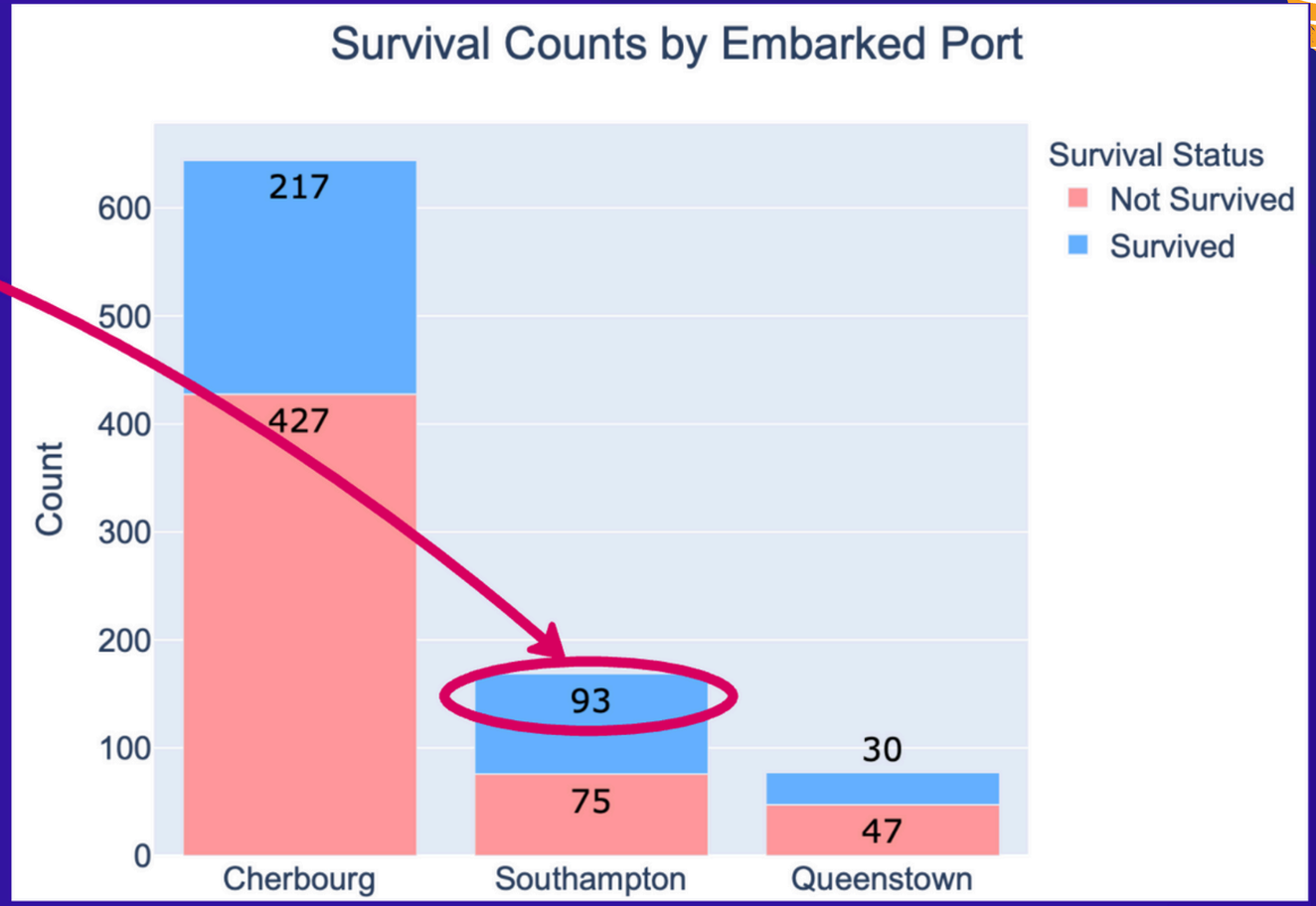
# Deletion of 177 NaN ‘Age’ rows (not column) approved



# Embarked feature check

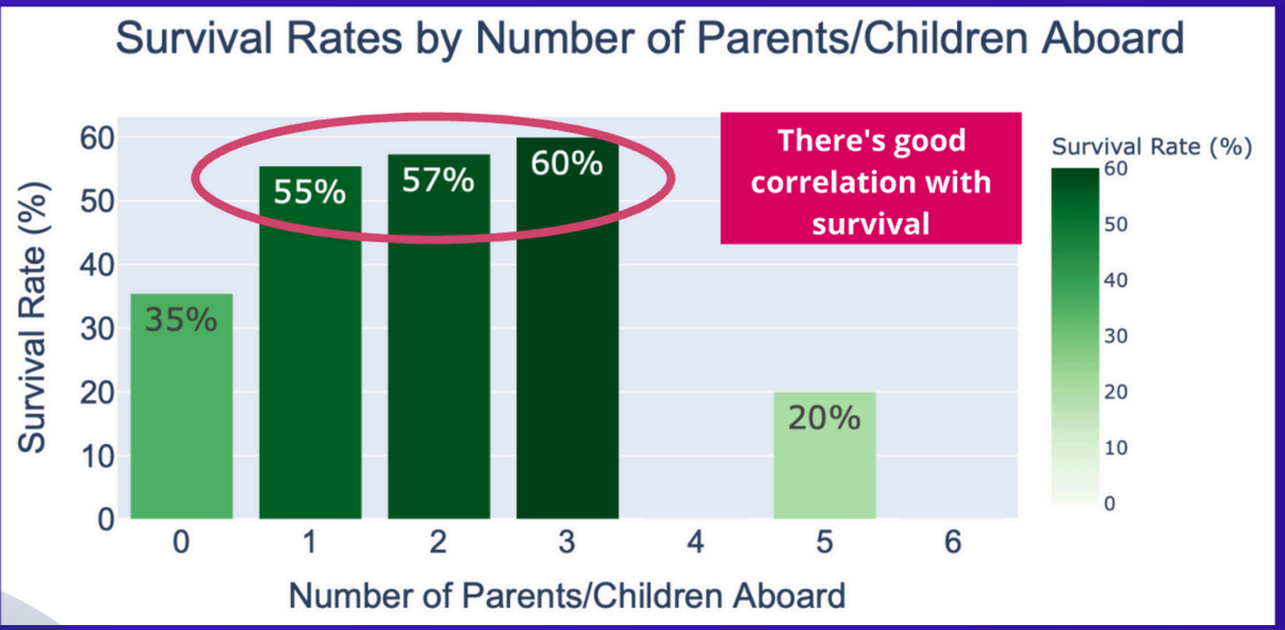


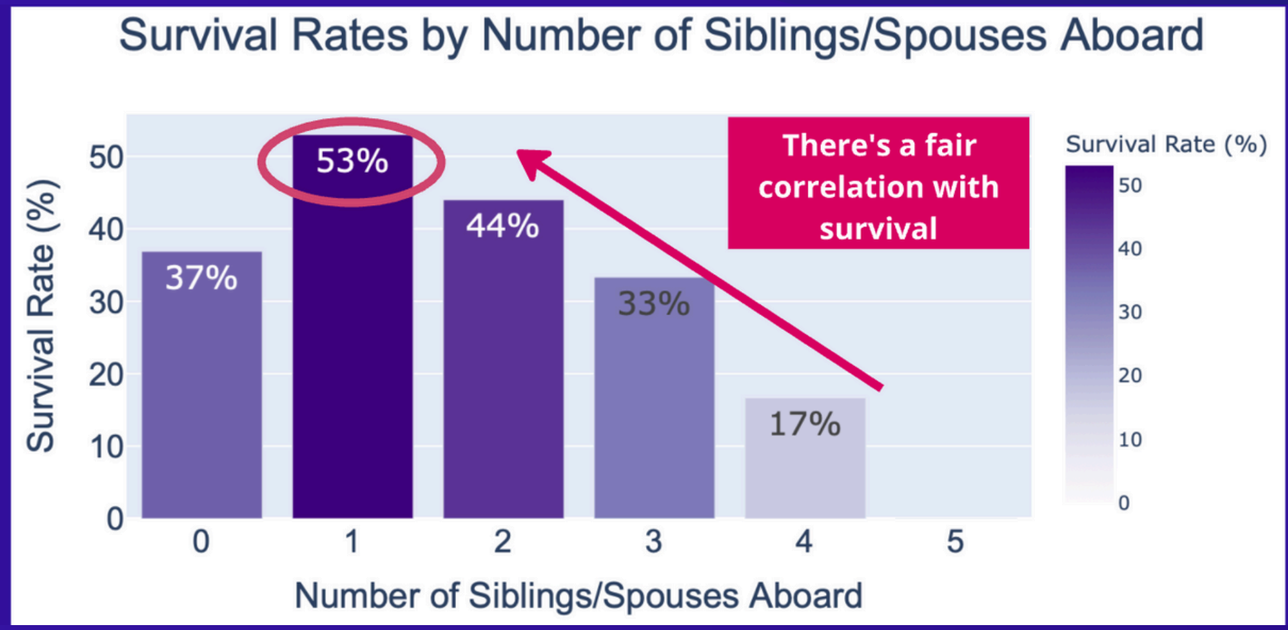




**It's correlated but just "superficially" , since in Cherbourg due to a high number of people the survival rate is harder to increase/decrease.**

***It's not the same: 50% out of 4, than 50% out of 1000***

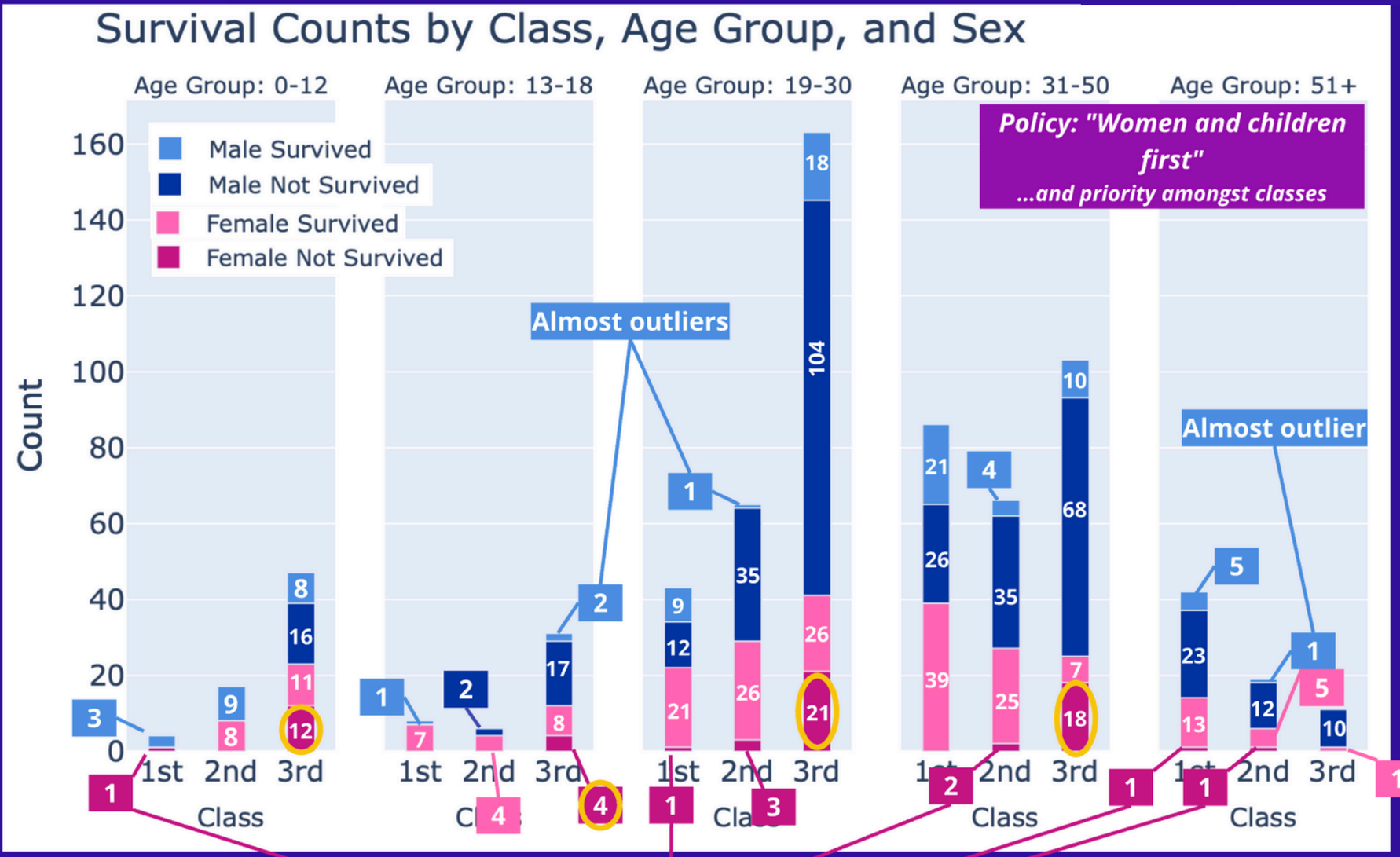
# Parch & SibSp features check



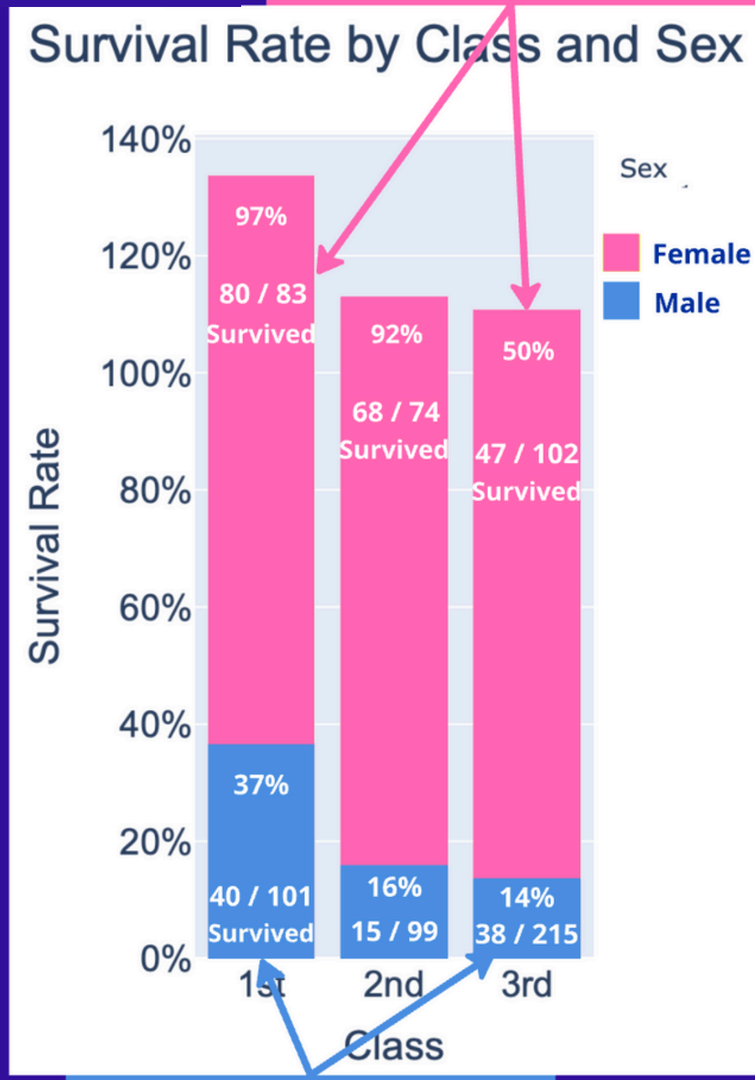




**Both variables were essential to add value to the model**

# Exploratory data analysis




Numerically not much difference between the number of women in 1st and 3rd class



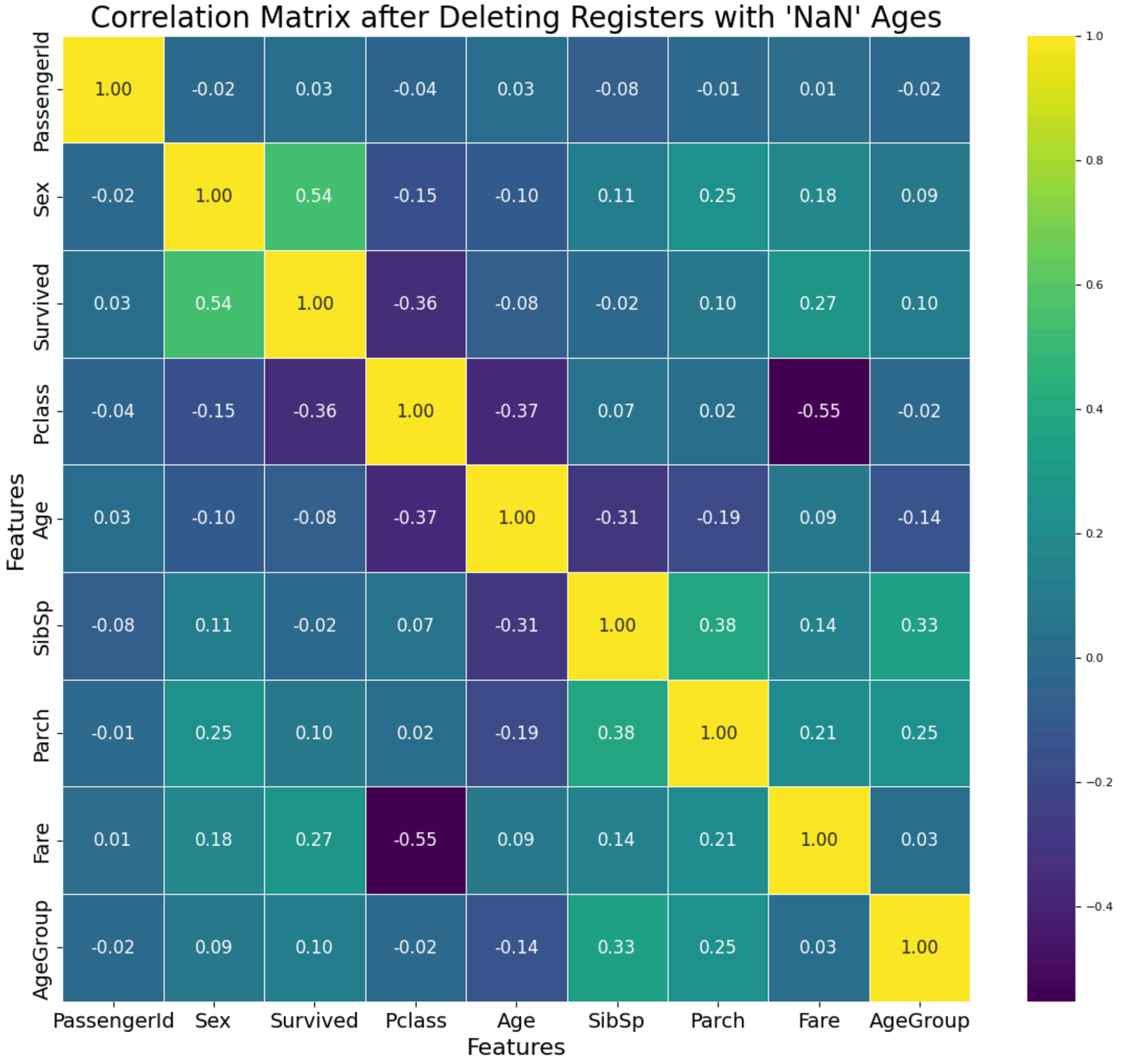

Almost the same number of men survived

# Feature Engineering

**Relevant variables:**

1. **Sex**
2. **Pclass**
3. **Fare**
4. **Age**
5. **SibSp**
6. **Parch**
7. **+ New variable**

**‘AgeGroup’ created in order to group Ages for classification improvement**



# Model(s)

# Initial pool of Model selection



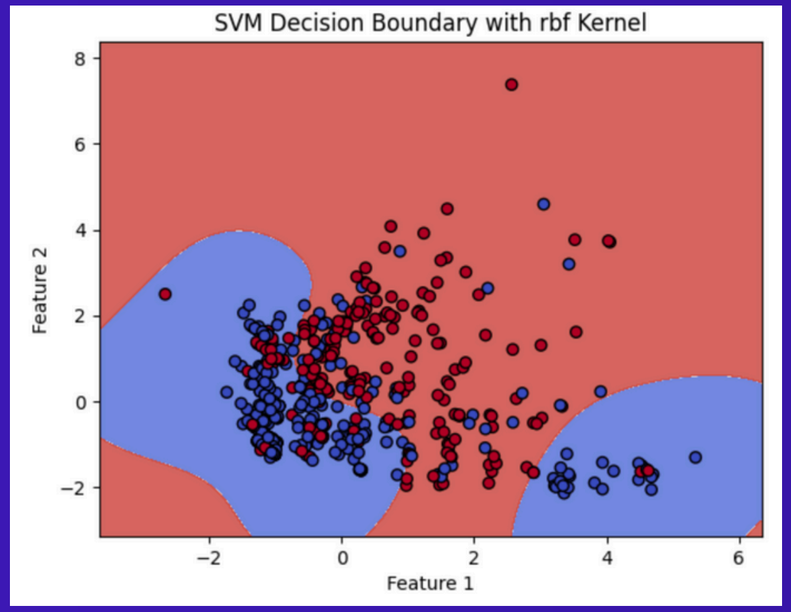
| Aspect | SVM | Logistic Regression | Neural Network |
|---|---|---|---|
| **Complexity** | High (depends on kernel) | Low | High (depends on model) |
| **Interpretability** | Moderate | High | Low |
| **Handling Non-Linearity** | Good (with kernels) | Limited unless feature engineering | Excellent (with deep architectures) |
| **Training Time** | Slow with large datasets | Fast | Long (depends on architecture) |
| **Computational Resources** | High | Low | Very High |
| **Supervised Learning** | Yes | Yes | Yes |
| **Performance with Low-Dimensional Data** | Good | Good | Depends on how low |
| **Suitability for Small Datasets** | Good | Good | Often requires larger datasets |
| **Classification** | Yes | Yes | Yes |
| **Binary problems** | Yes | Yes | Yes |
| **Recommendation** | * | ** | *** |

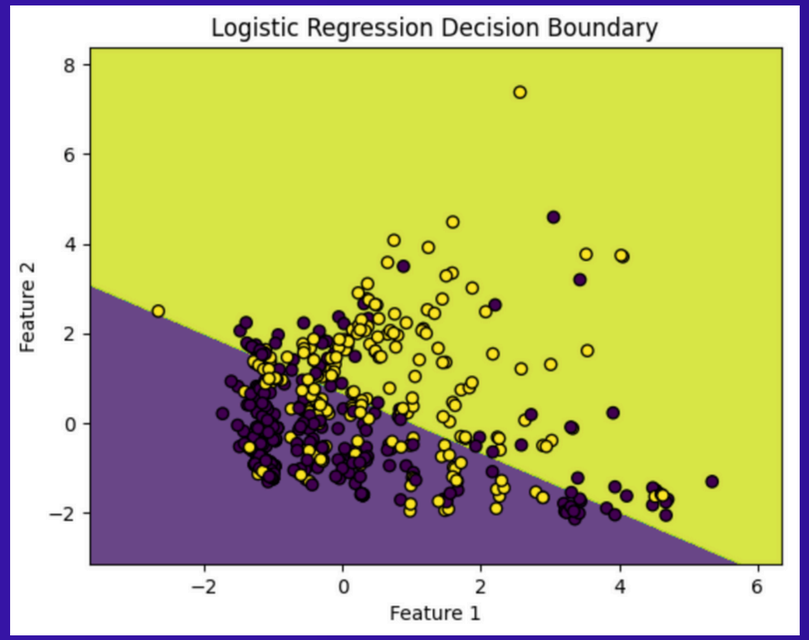
*** Tensorflow https://www.tensorflow.org/datasets/catalog/titanic?hl=es-419

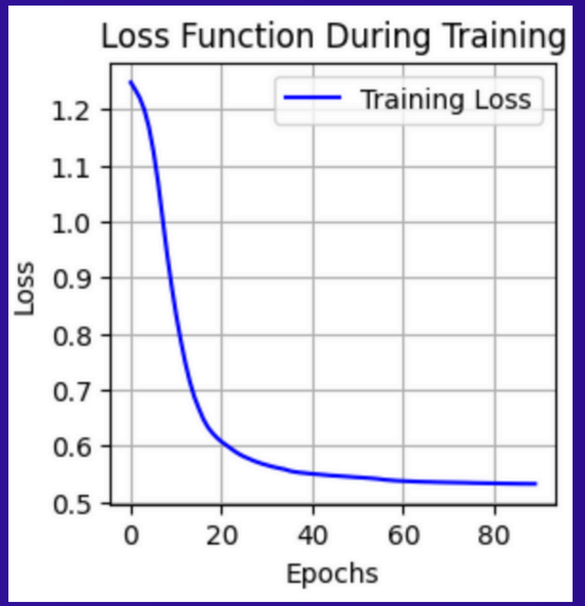
# Model Training



| Aspect | SVM | Logistic Regression | SVM + Autoencoder Tensorflow |
|---|---|---|---|
| Hyper Parameter Tuning | GridSearch and RandomizedSearch took a lot of time testing , only able to test RBF | Tested with GridSearch and RandomizedSearch no gain or loss registered to final Model | Epochs: 90 & Batch size: 128Optimizer: Adam performed better |
| k-Cross-Validation | Tested from 5 to 20 Folds, no gain or loss registered to final Model | Tested from 5 to 20 Folds, no gain or loss registered to final Model | It skewed very hard to false positives, it was discarded. |
| Loss Function | Hinge Loss | Cross entropy loss | Mean Squared Error |



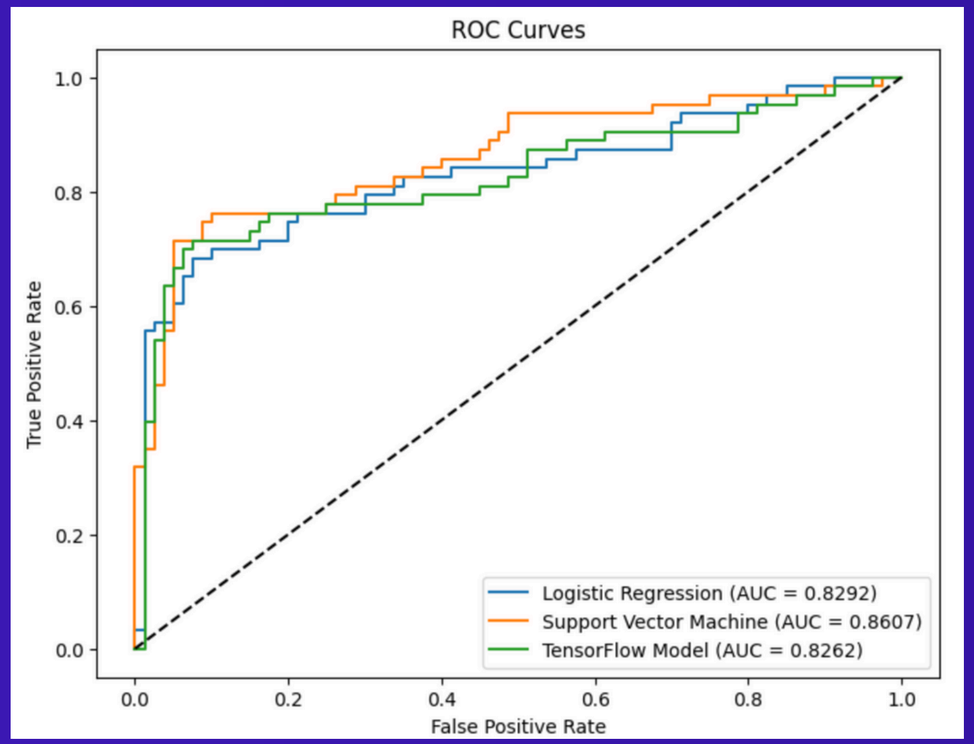



# Model Selection

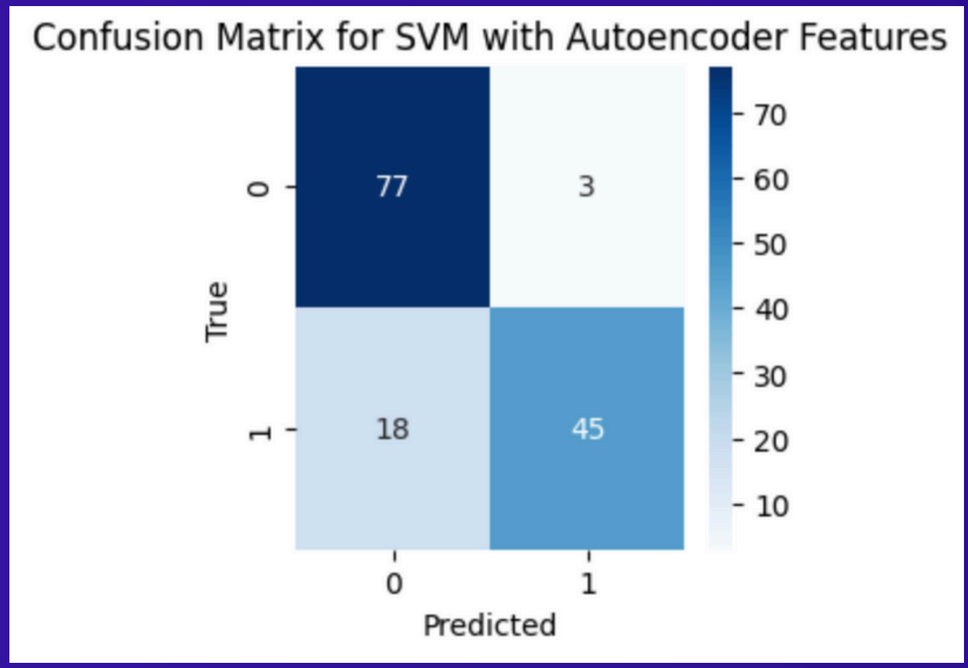


## *Methodology: Best Accuracy and f1-score*

| Aspect | SVM | Logistic Regression | SVM with Autoencoder Features (Tensorflow) |
|---|---|---|---|
| **Accuracy obtained** | 0.8322 | 0.8042 | **0.8531** |
| **f1-score obtained** | 0.7895 | 0.7544 | **0.8108** |




## *New Feature: 'AgeGroup' added 1% Accuracy!*




*Actual ROC AUC for Tensor differs a little bit due to re-run training to get graph*

# Conclusion

- Objective
- Hypothesis
- Future steps

# Results & Conclusions

**Deletion of 177 'Age' rows proved to backfired, since now, our top models can't predict with NaN values for Kaggle.**

**However, independently from Kaggle Accuracy and f1-score are still great.**

**Parch & Sibsp proved to be useful, and are still varibles that can be explored, like the creation of a new feature that establishes number of family. This can be a great development in strengthening relation from the person to survival or not.**

**Feature engineering proved to give more significant improvement to Accuracy with less work.**

**Hyperparameters tunign proved to requiere more time and effort to improve Accuracy, actually it tends to lower it easier than raise it, except in Neural Networks, which can lead to significant variance while adjusting Epochs and BatchQty**

**Objectives achieved?**
Yes.
Target: 80%
Achieved: 85% (Maximum)

**Hypothesis achieved?**
Yes.
Sex and Age were the most important factors

**Improvements to the future**
Creation of a new column:
One that points exactly what is their family is something very interesting to explore, since there was important correlation to survival with Parch and Sbsp variables

# Thank You!