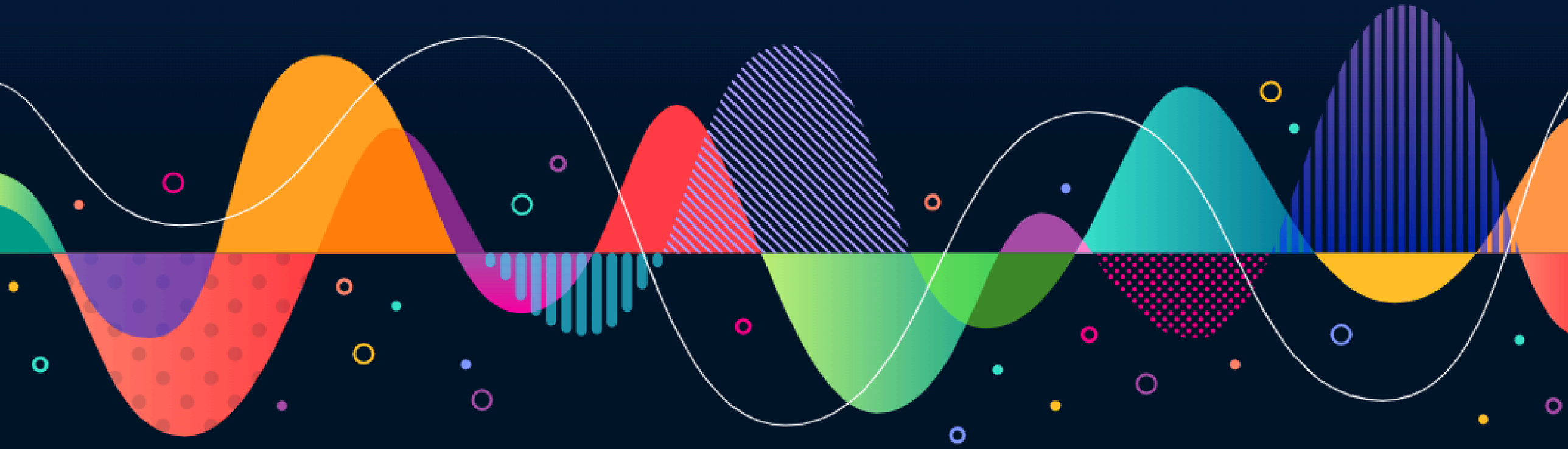
Team:

- Maricarmen Daniela Barillas Duarte a01369993@tec.mx
- Daniel Alejandro Olivares Ángeles a01754838@tec.mx
- Gael Eduardo Pérez Gómez a01753336@tec.mx
- Santiago Martínez Vallejo a00571878@tec.mx

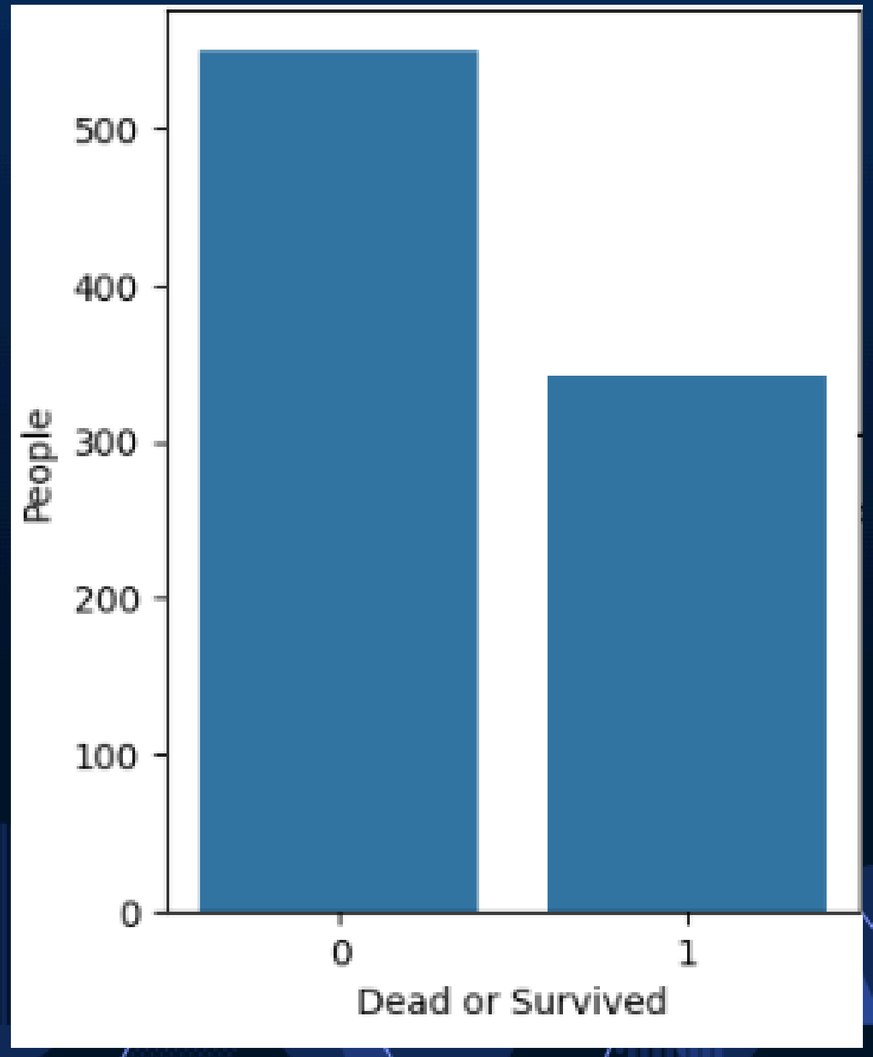
Kaggle – Titanic classification



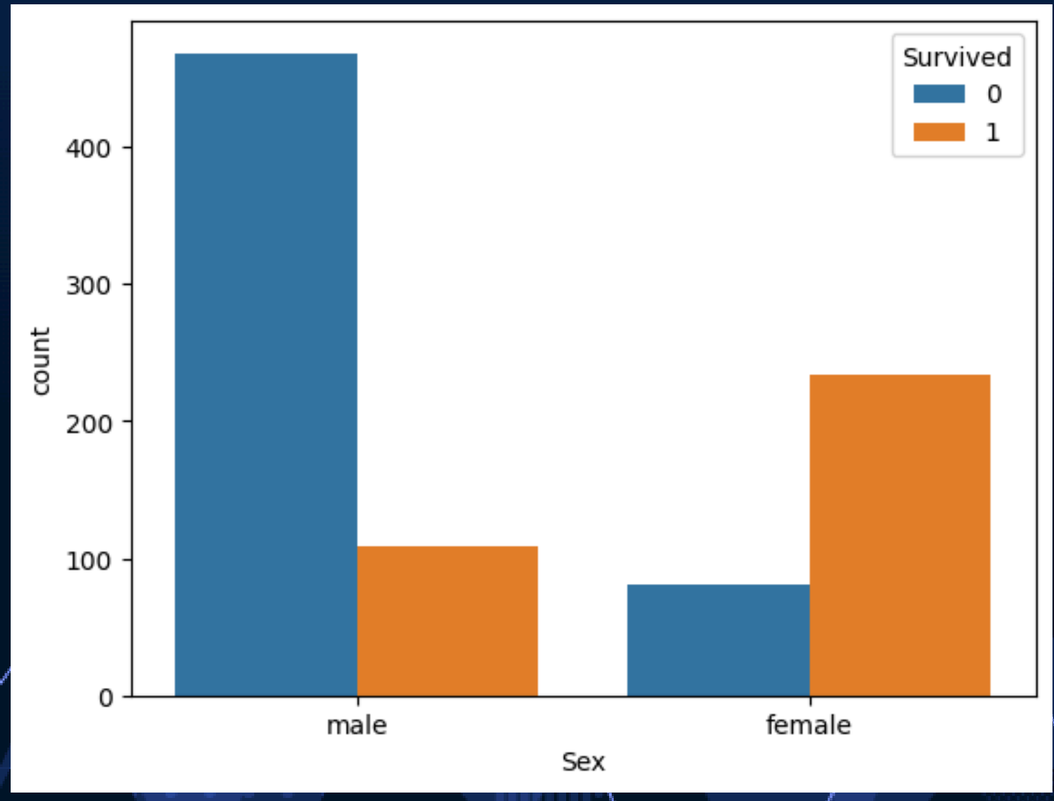
Problem & Goals

- Identify the most relevant features that determine a passengers survival probability.
- Select the best model for predicting the survival probability.
  - Identify the model with an accuracy of 85.
  - Consistent K-cross fold validation avarage.
- Hypothesis: Random Forest will be the best model.
  - Randomness in sampling of data points and selection of features. Less overfitting and improved generalization.

Data Distribution

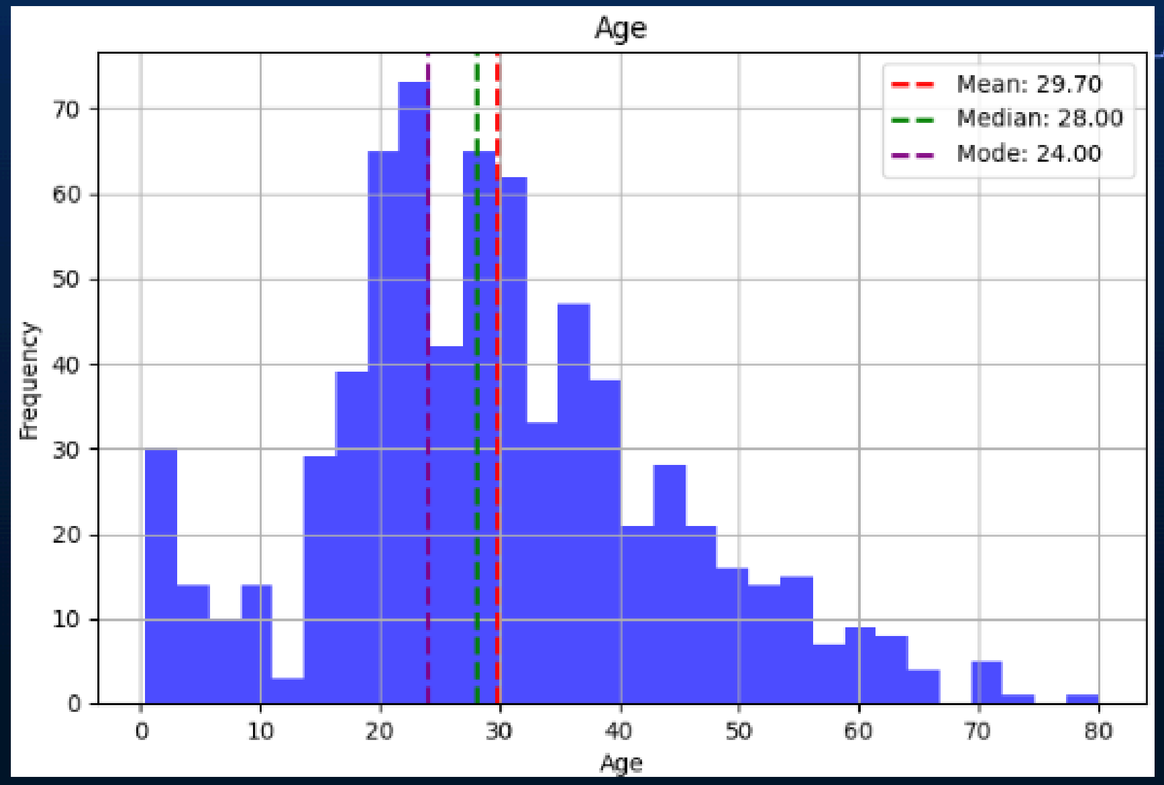


- The data leans on dead rather than not survied.
- Sex distribution with survived or dead.



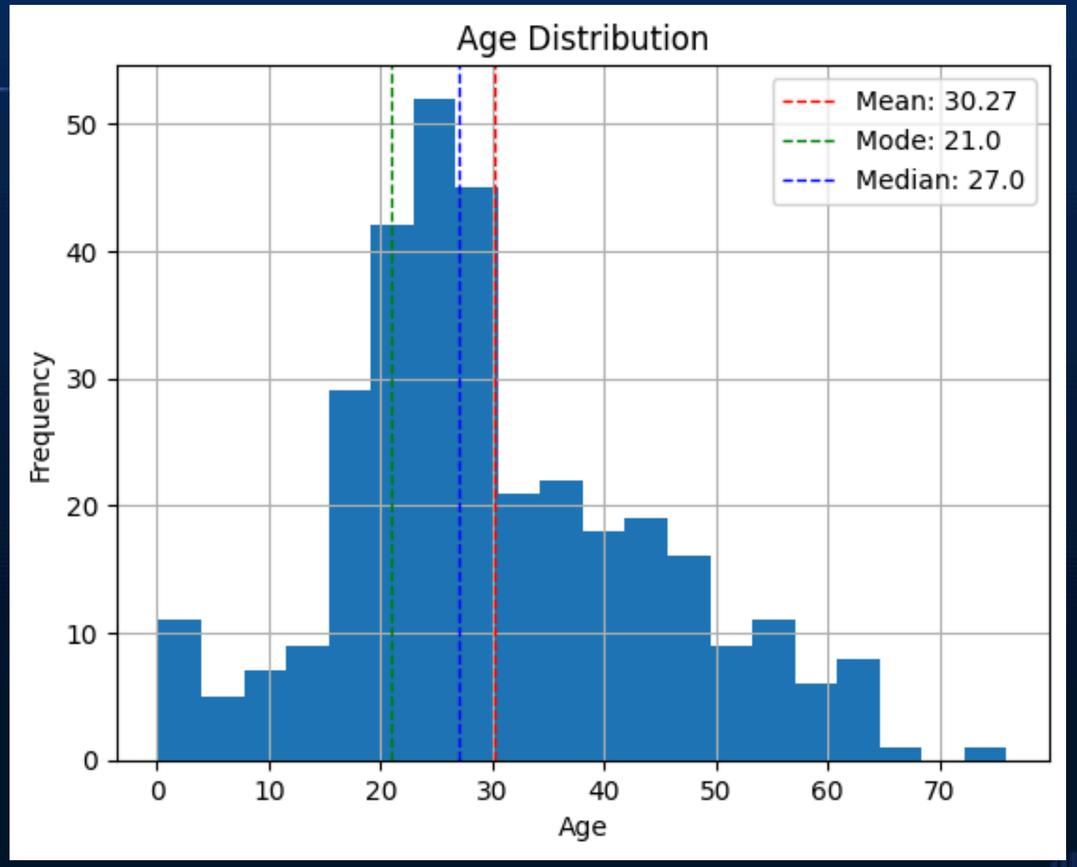
Missing Data

Training Set



- Age: 19.9% missing
- Median for missing values

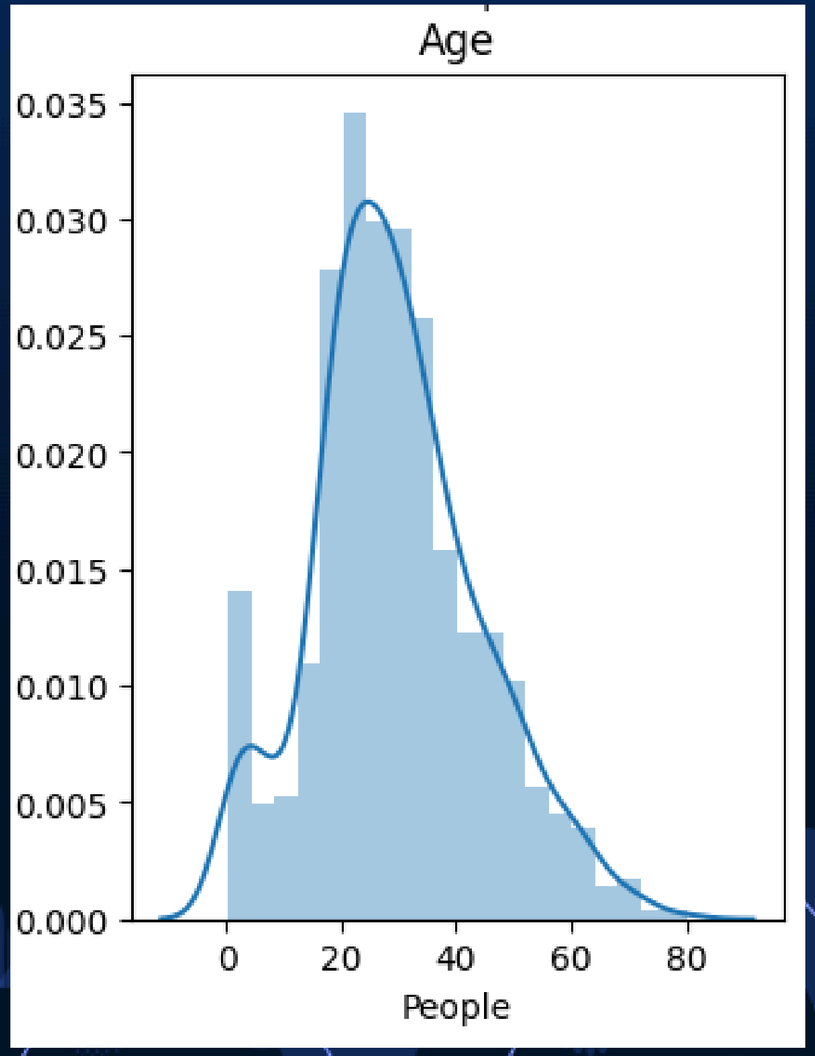
Test Set



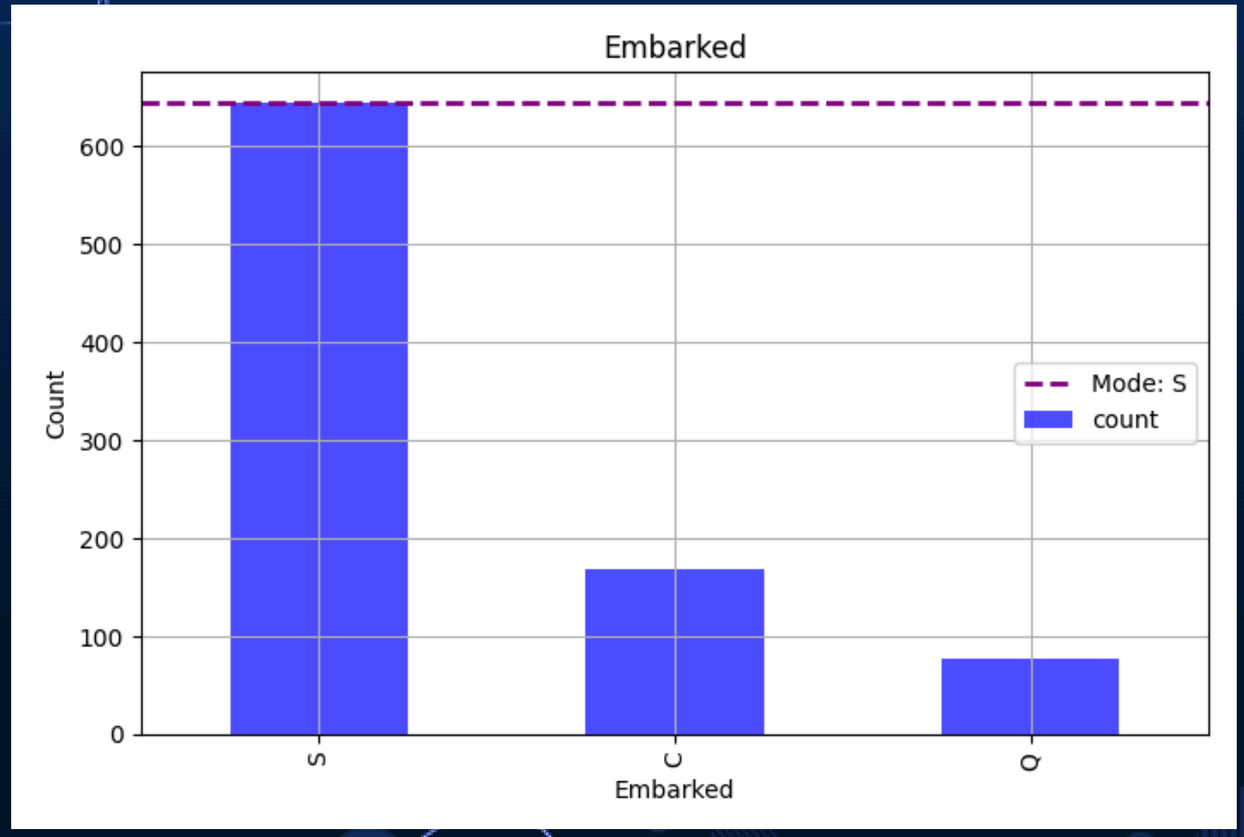
- Age: 20.6% missing
- Median for missing values

Missing Data

Age distribution

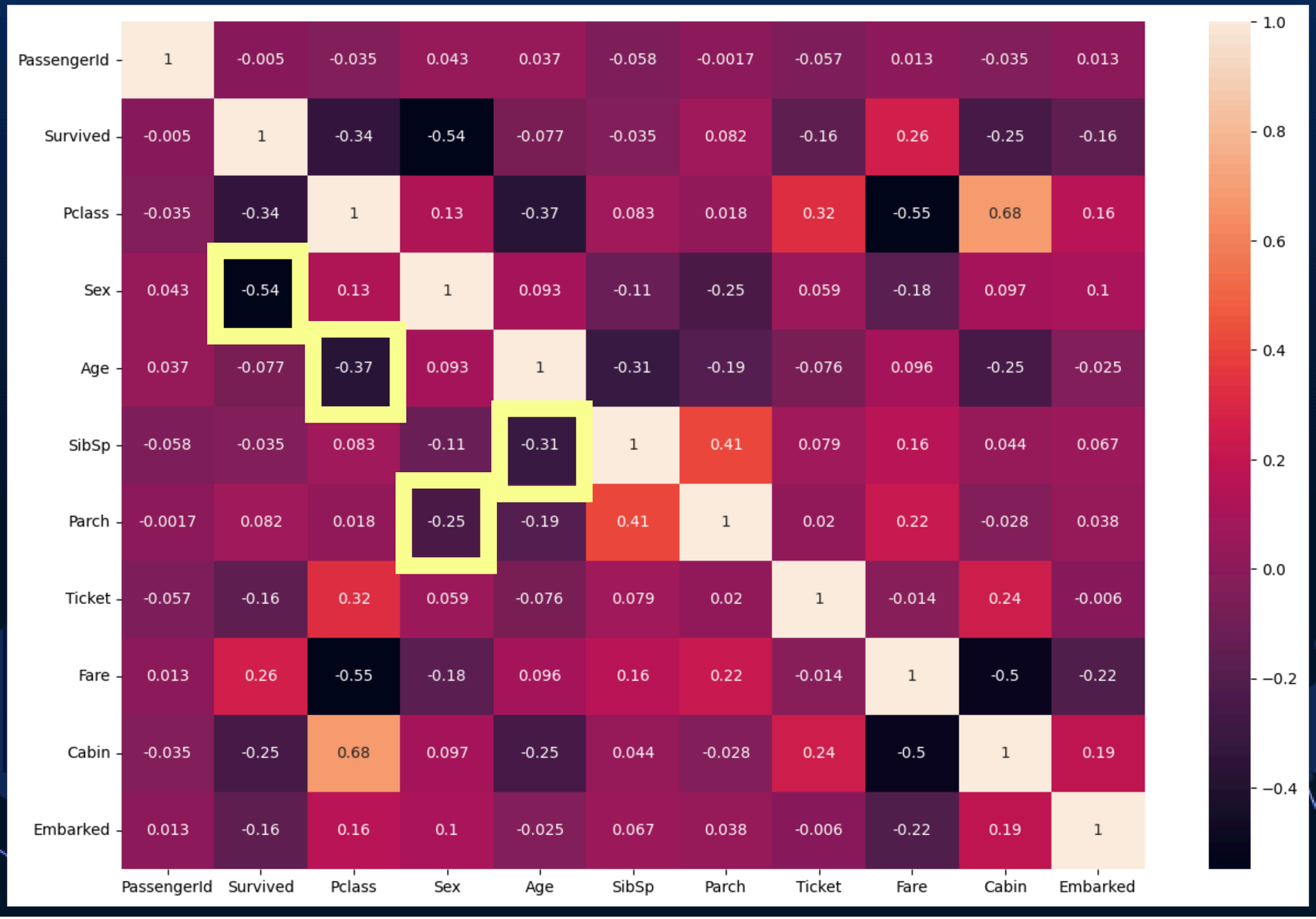


Embarked



- Embarked: 2 values missing
- Mode for missing values

Correlation Analysis

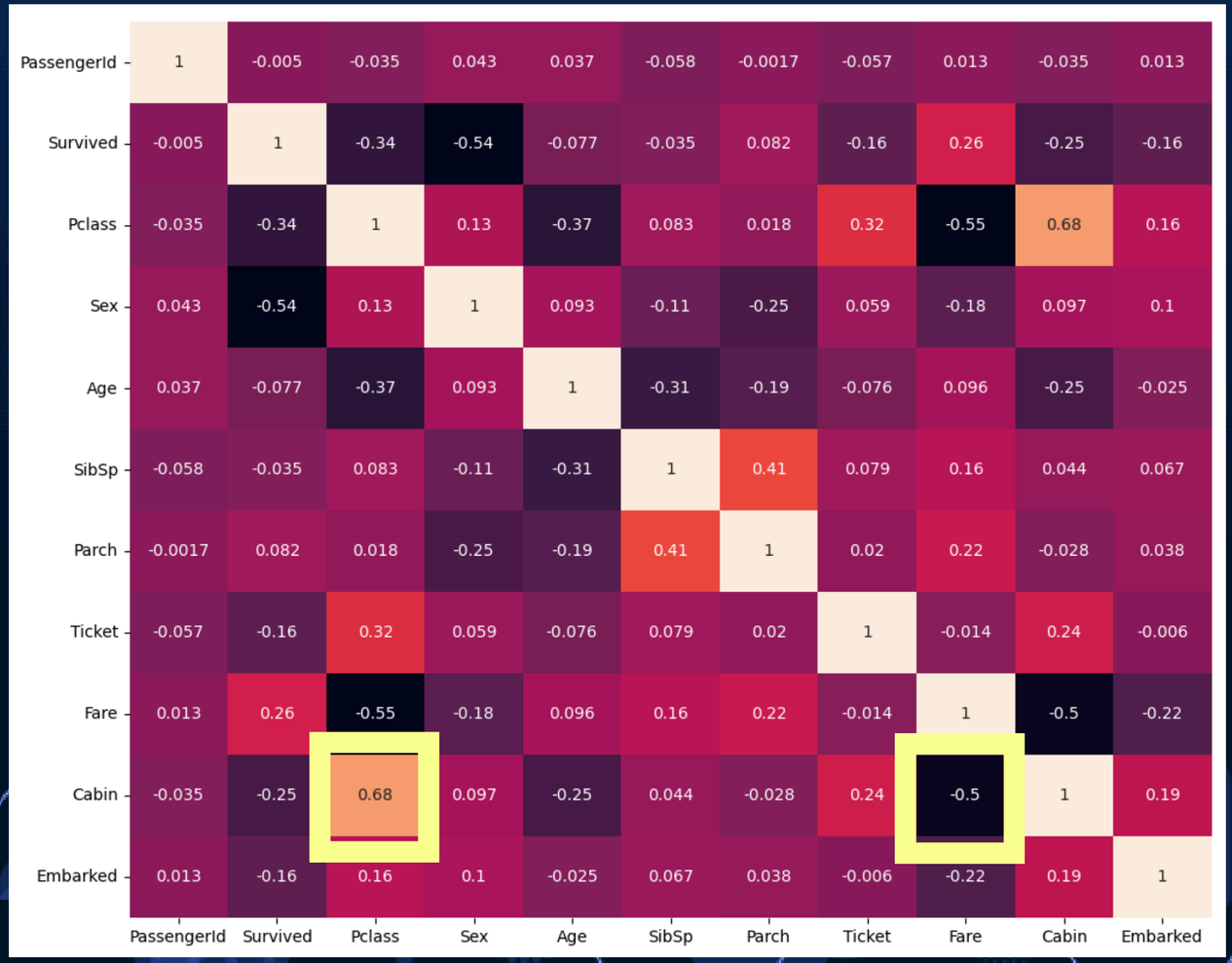


- Sex
- Age
- Pclass
- Embarked
- Parch + Sibsp *

* go to Data transformation

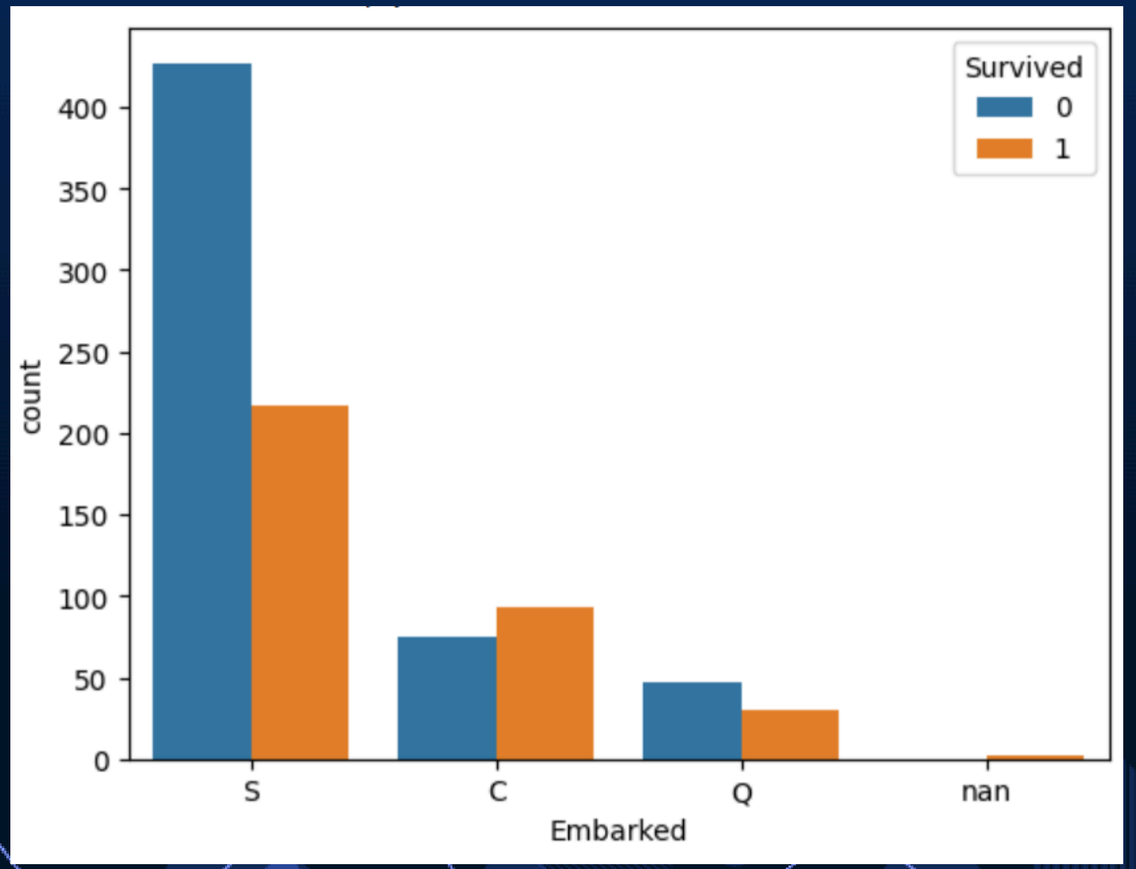
Dropped Features

- Passenger Id: Irrelevant feature, just an assignment to the tickets.
- Ticket and Cabin: Both are related in the correlation matrix with Pclass, so we can prescend with both, also, there are a lot of missing values in cabin.
- Name: As we saw before, there are a lot of names unique, dont make sense to generalize this side, irrelevant.
- Fare: Too high related with pclass.



Data Transformation

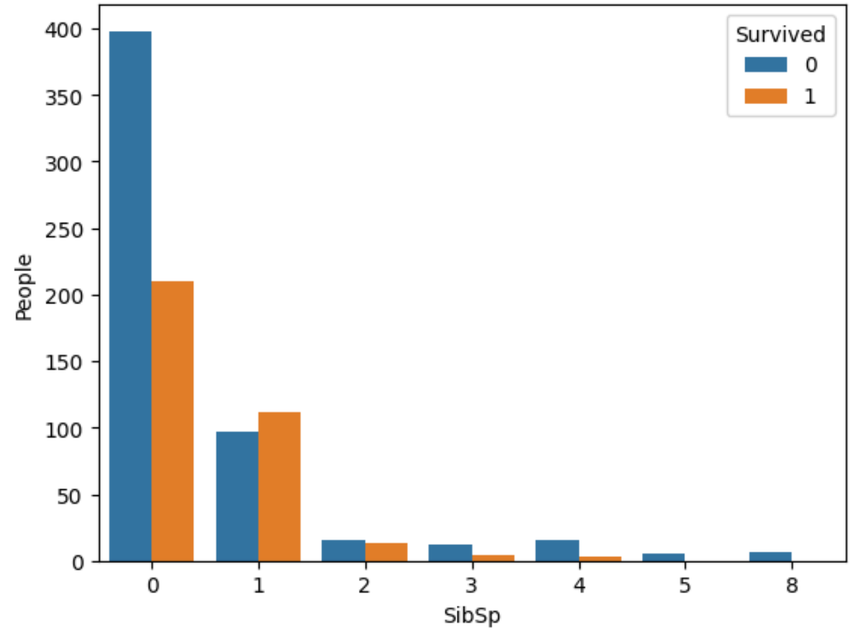
- **Sex:**
  - male -> 1
  - female -> 0
- **Embarked:**
  - C -> 1
  - Q -> 1
  - S -> 0
- fit_transform from scikit learn to standardize the features.



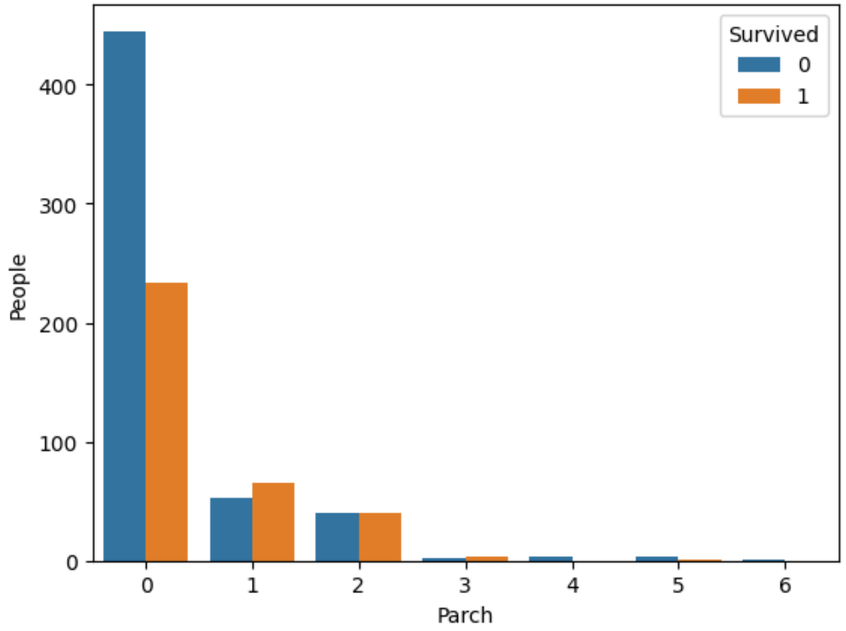
Data Transformation

- We created “companians” Parch + Sibsp

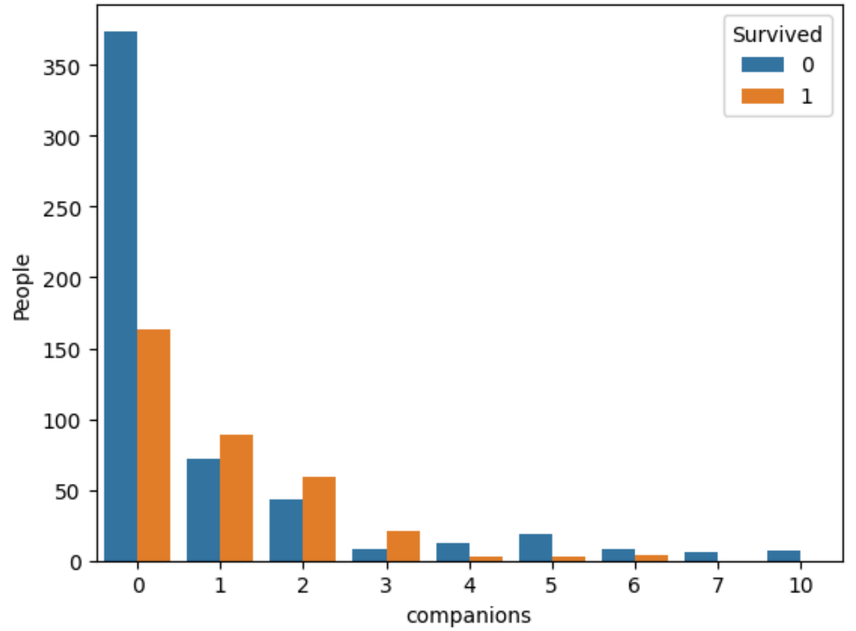
SibSp

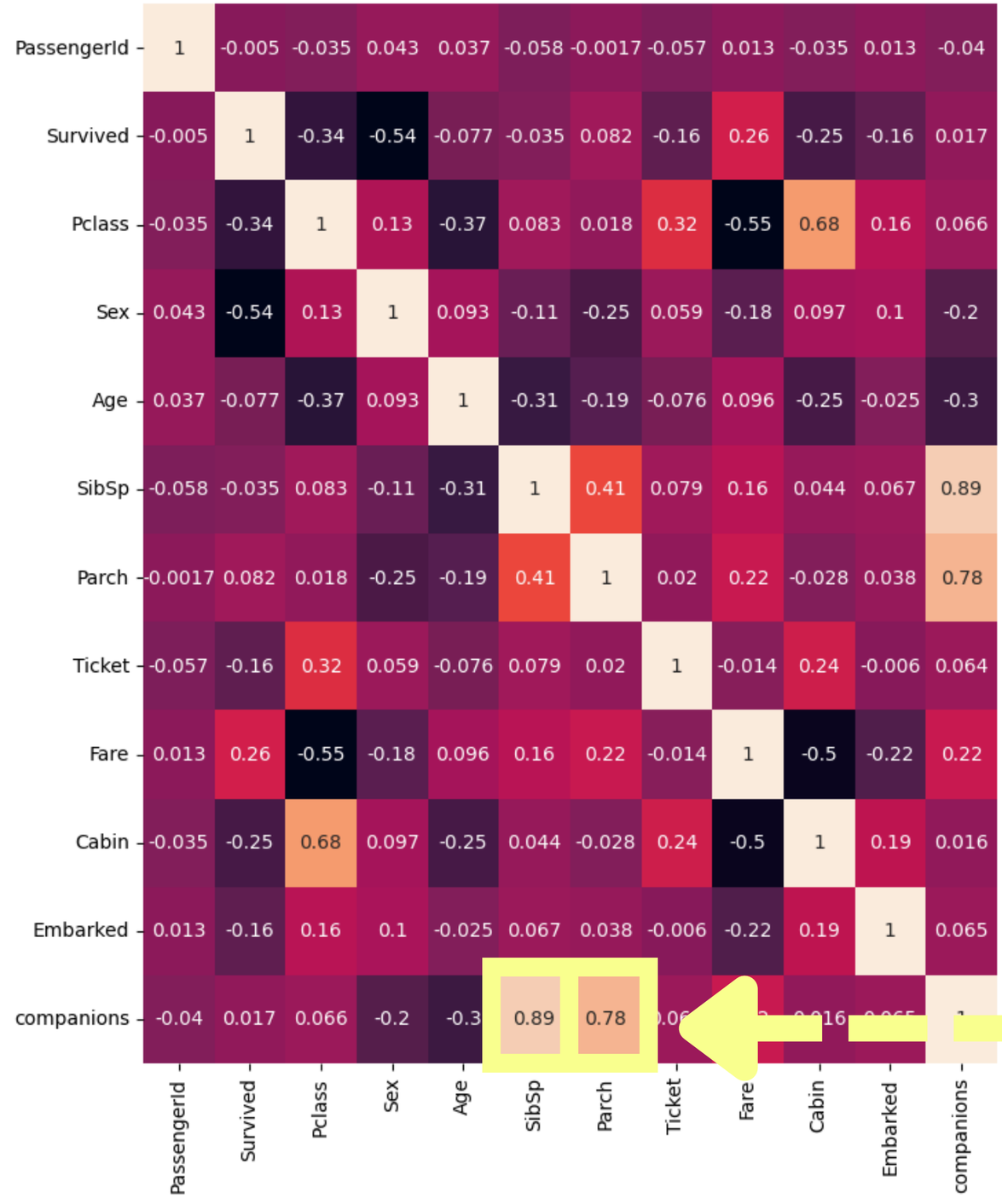


Parch



Companions

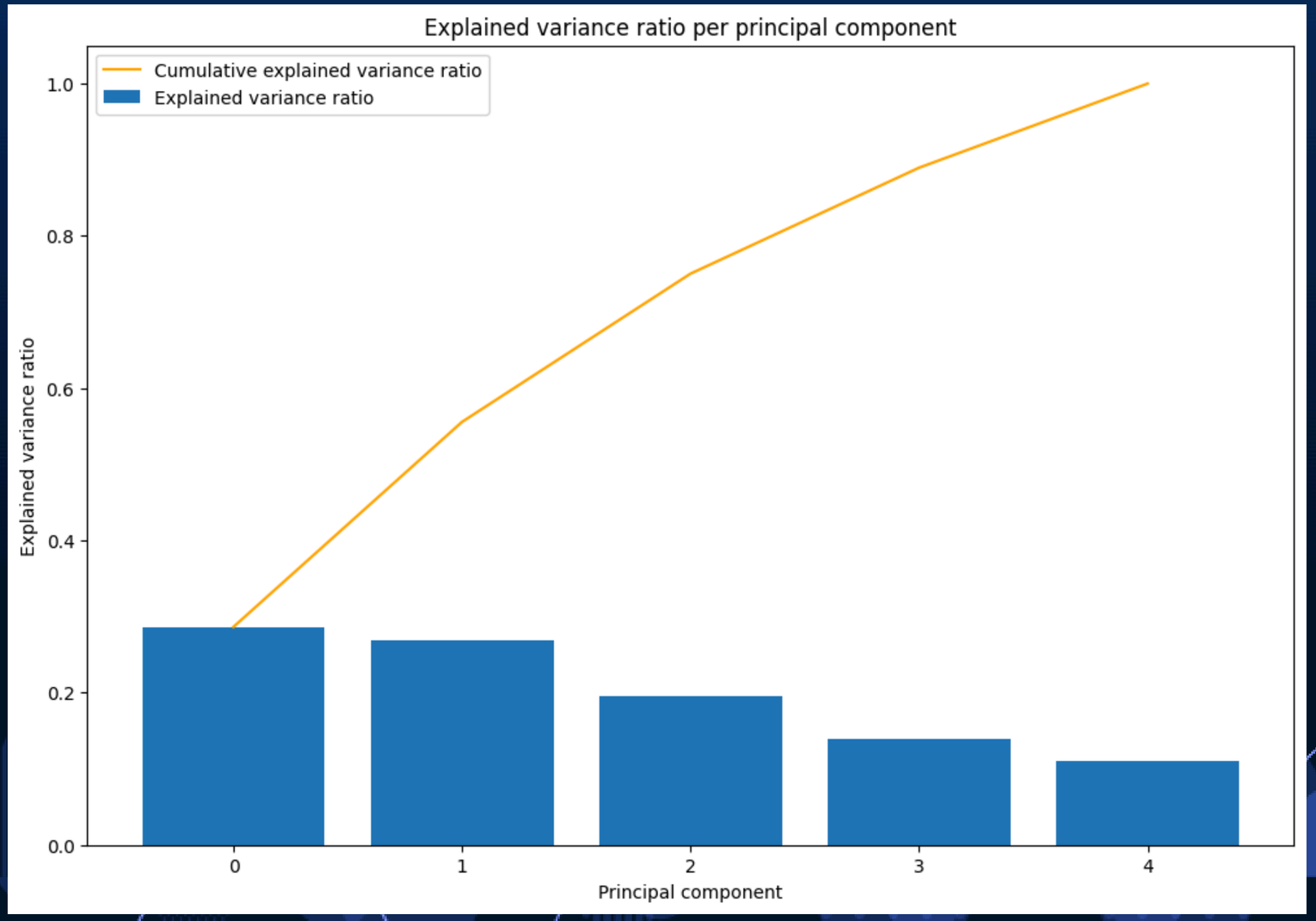




Data Transformation

- **Correlation with companions**

Principal Component Analysis



85% of variance

PC1
PC2
PC3
PC4

Random Forest

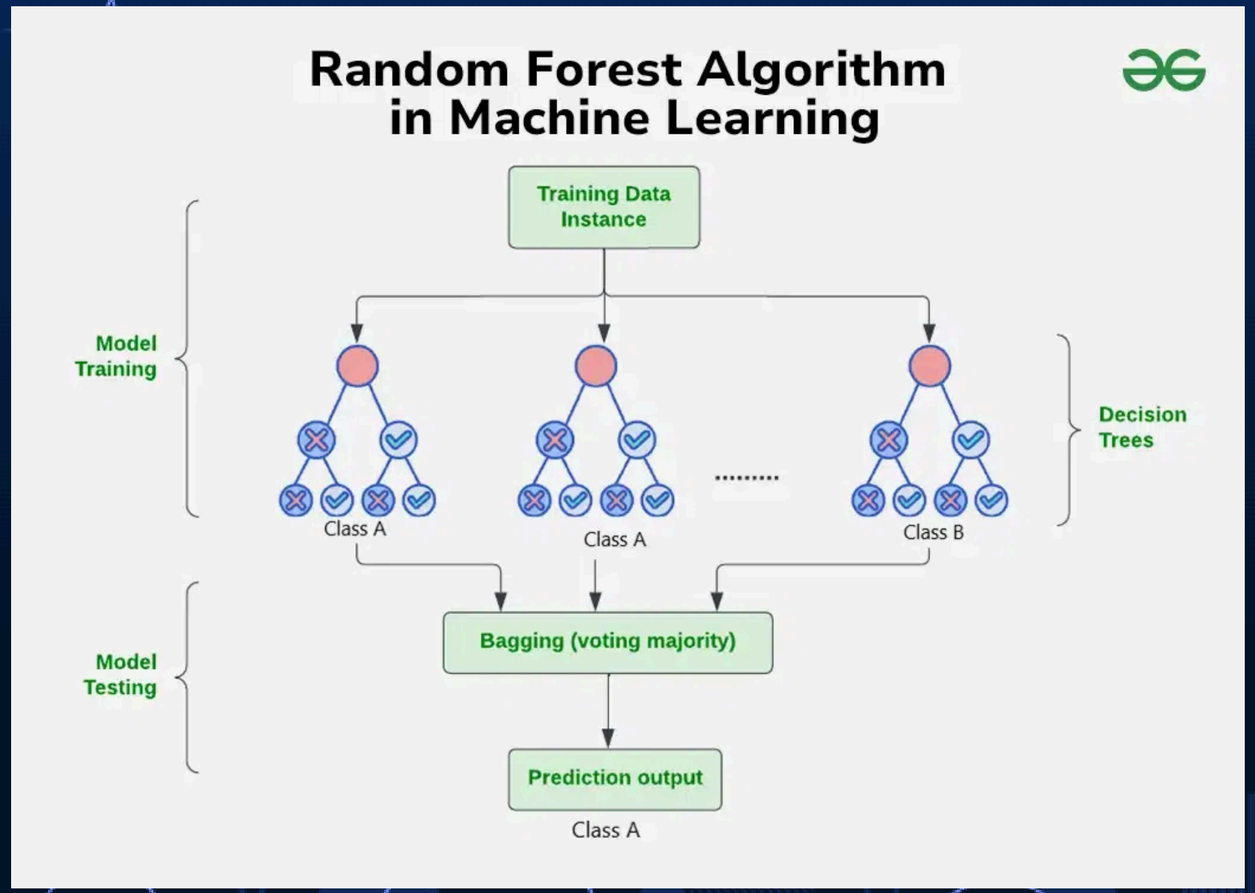
- Handles both categorical and numerical features well.
- Provides feature importance, which can help identify influential factors.
- Often achieves high accuracy

- Strengths:
  - Handles missing data and non-linear relationships well.
  - Robust to overfitting with enough trees.
- Weaknesses:
  - Harder to interpret.
  - May require tuning (e.g., number of trees, depth).

Random Forest

Hyperparameters used

- max_depth: 5 <
- max_features :log2 -
- min_samples_leaf: 2 >
- min_samples_split: 10 >
- n_estimators: 100 >



https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/

Logistic regression

- Nature of the Problem
- Inherent binary classification algorithm
- Interpretable.
- Linear Relationship
- Provides probability estimates.

- Weaknesses:
  - May underperform with non-linear relationships.
  - Can be sensitive to outliers and multicollinearity.

- Regularization: Elasticnet (L1 and L2)

Support Vector Machine (SVM)

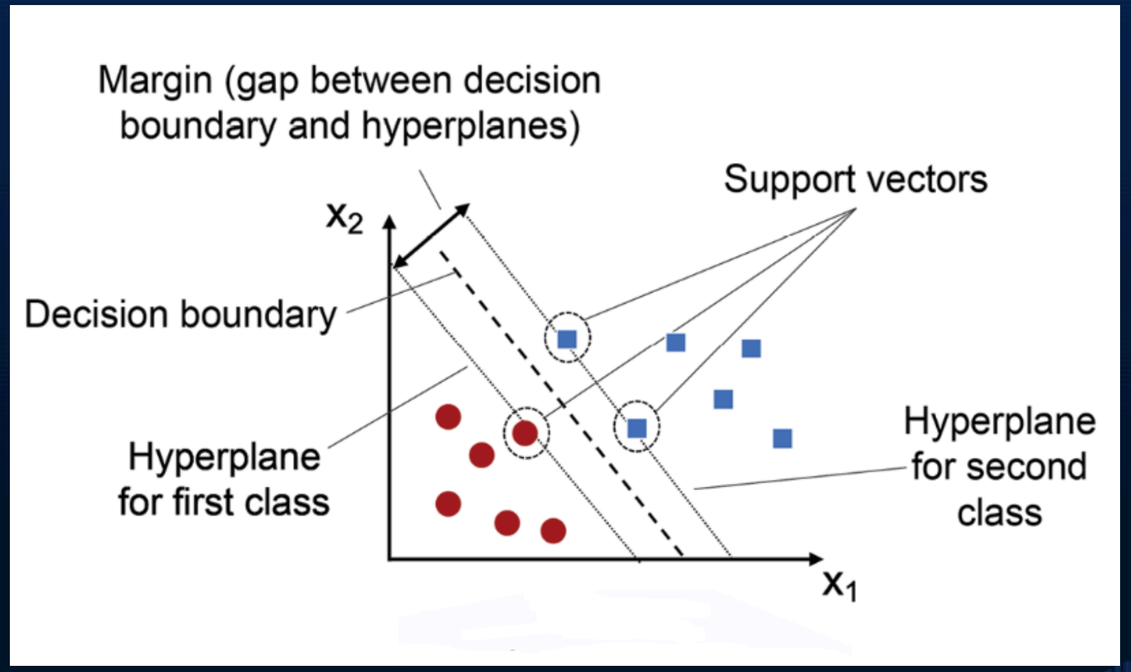
- Effective for many dimensions.
- Can handle non-linear relationships.
- Requires hyperparameter tuning.

Hyperparameters:

- Regularization (C): controls the trade-off between maximizing the margin between classes and minimizing the classification error on the training data.
- Kernel: defines the mathematical function used to transform the input data into a higher-dimensional space.
- Gamma: defines how far the influence of a single training example reaches.

Support Vector Machine (SVM)

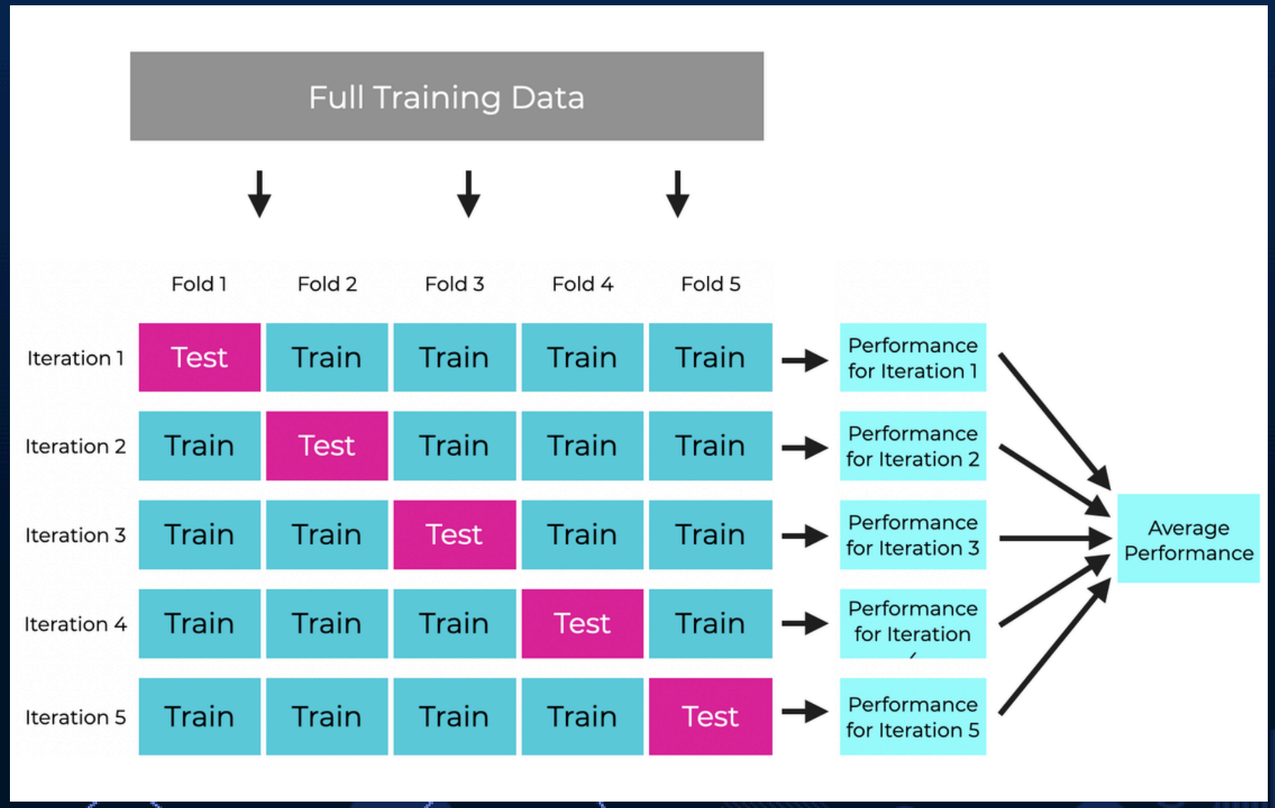
- 'C': 50: relatively high, because we prefer the correct classification of all points.
- 'gamma': 0.1: the influence of each training point on the decision boundary is moderate.
- 'kernel': 'rbf': the data was best separated using the Radial Basis Function (RBF) kernel.



https://vitalflux.com/classification-model-svm-classifier-python-example/

K-cross fold validation

- Iteration 100 times
  - Time
- With k-cross
  - Avg Confusion Matrix
  - Avg Auc



https://www.sharpsightlabs.com/blog/cross-validation-explained/

Comparison

Random Forest

Accuracy: 0.8170
Precision: 0.8199
Recall: 0.8170
F1-Score: 0.8118

Logistic Regresion

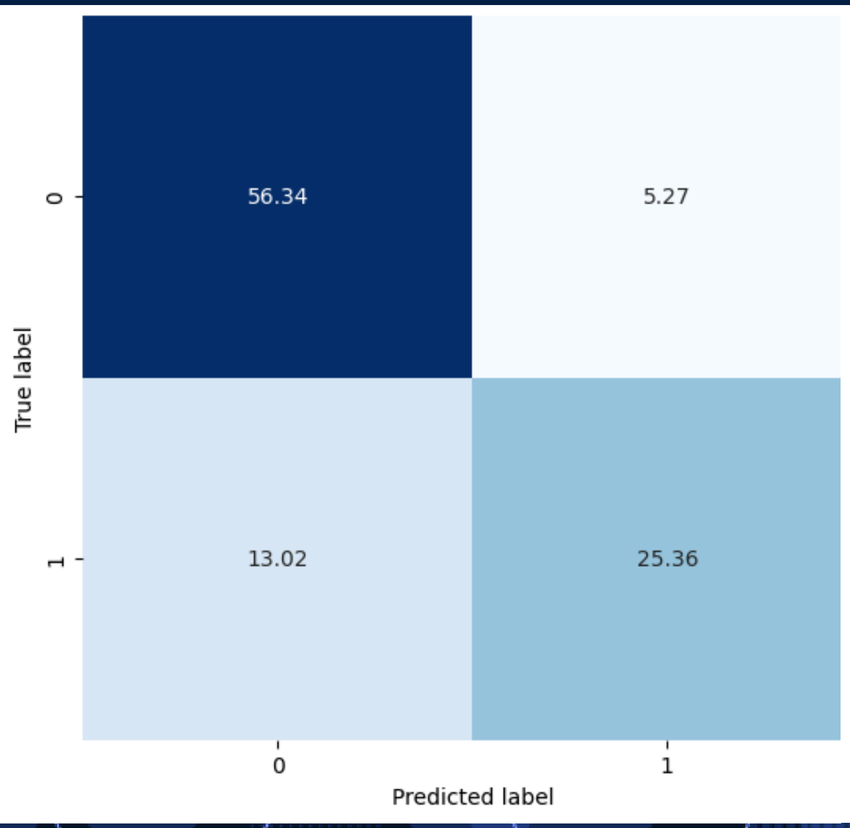
Accuracy: 0.7789
Precision: 0.7772
Recall: 0.7789
F1-Score: 0.7776

Support Vector Machine

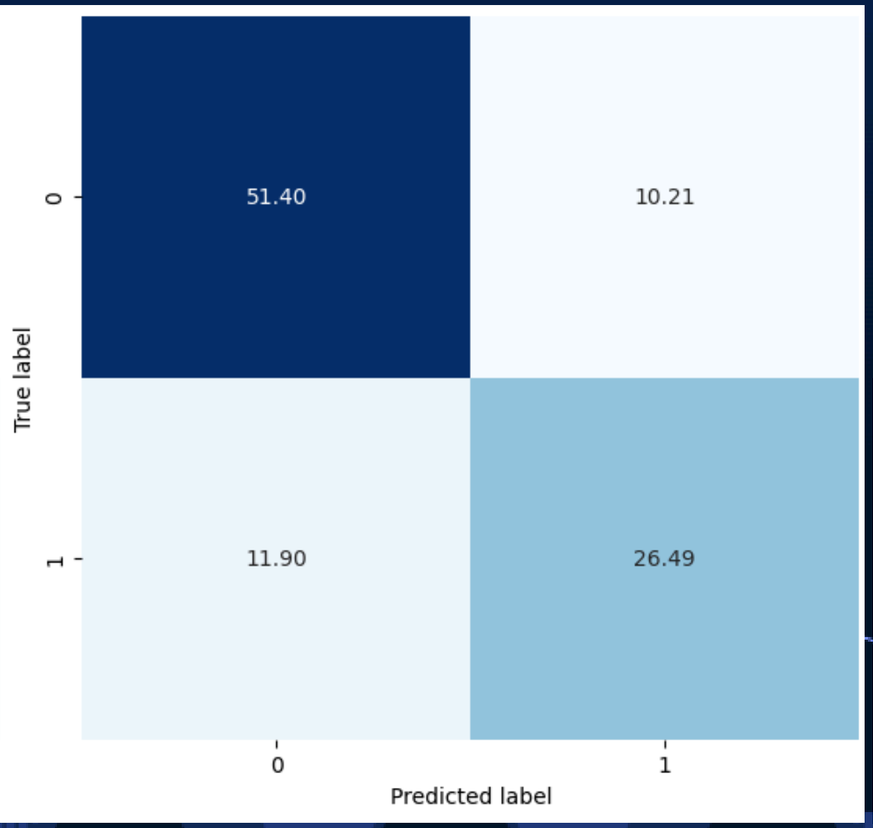
Accuracy: 0.8193
Precision: 0.8013
Recall: 0.7991
F1-Score: 0.7947

Confusion matrix comparison

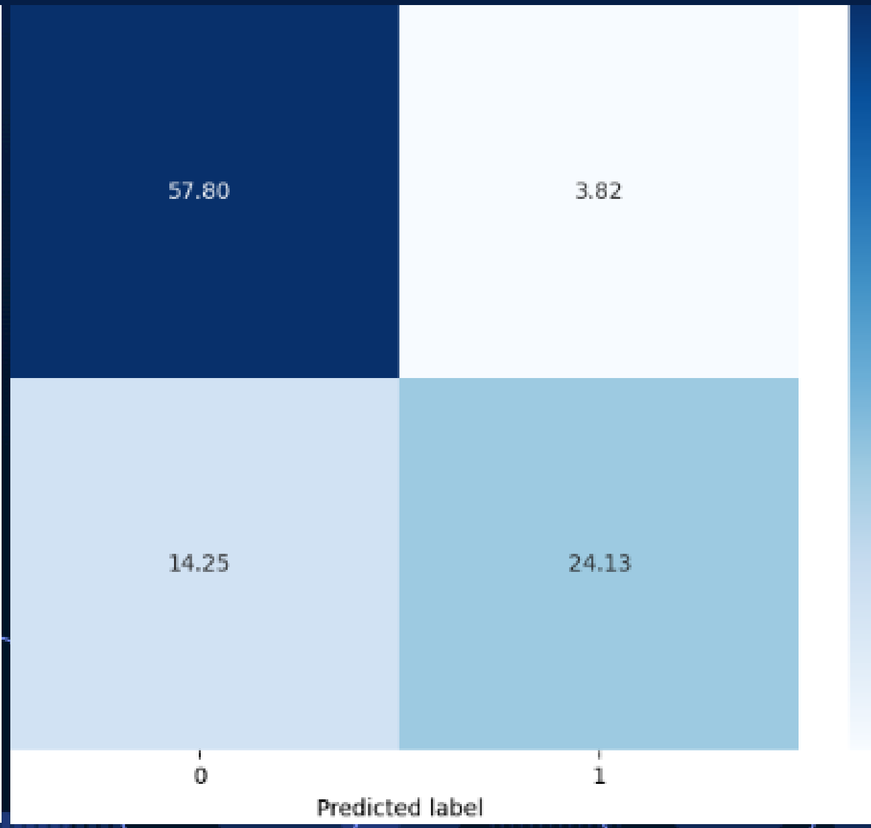
Random Forest



Logistic Regresion



Support Vector Machine



Discussion

Through our analysis, we identified that the most significant variables contributing to the model's performance are five key variables.
We applied PCA using four components to reduce dimensionality and improve model efficiency.
Despite the improvements, the model's accuracy did not reach the 85% threshold.

Best Model

SVM performed better because various hyperparameters were adjusted and different combinations of these parameters were evaluated to determine which configuration yielded the best performance.

Future Improvement

- One-Hot Encoding: apply one-hot encoding to categorical features to ensure they are correctly represented in the model.
- Model Diagnostics: conduct residual analysis and analyze ROC curves to understand model performance better.
- Apply transformation to variables to linearilize not linear relations.
- Hyperparameter Tuning: perform grid search or randomized search for hyperparameter optimization to enhance model performance.
- Polynomial Kernels: experiment with polynomial kernels to capture interactions between features.
- Compare models with others (Naive-Bayes, knn, etc).

References

- "A Practical Guide to Support Vector Classification" por Chih-Wei Hsu, Chih-Chung Chang y Chih-Jen Lin: Este artículo ampliamente citado proporciona una guía práctica para la clasificación con SVM, incluyendo una discusión detallada sobre la selección de gamma.
- "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods" por Nello Cristianini y John Shawe-Taylor: Este libro ofrece una introducción completa a los SVM y otros métodos de aprendizaje basados en kernels, incluyendo una explicación detallada de diferentes tipos de kernels y cómo elegir el kernel adecuado para un problema específico.
- "Support Vector Networks" por Corinna Cortes y Vladimir Vapnik: Este artículo fundamental presenta el concepto de SVM y discute el papel del parámetro C en el control de la complejidad del modelo y la prevención del sobreajuste.
- Breiman, L. (2001). Random Forests. Machine Learning, 45, 5–32. © 2001 Kluwer Academic Publishers.
- Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
- Bishop, C. M. (2006). Pattern recognition and machine learning. Springer

References

- Ekinci, E., İlhan Omurca, S., & Acun, N. (n.d.). A comparative study on machine learning techniques using Titanic dataset. Kocaeli University, Faculty of Engineering, Computer Engineering Department. Retrieved from https://www.researchgate.net/profile/Neytullah-Acun/publication/324909545_A_Comparative_Study_on_Machine_Learning_Techniques_Using_Titanic_Dataset/links/607533bc299bf1f56d51db20/A-Comparative-Study-on-Machine-Learning-Techniques-Using-Titanic-Dataset.pdf
- Titanic Machine Learning Study from Disaster. (n.d.). UDSpace. University of Delaware. Retrieved September 11, 2024, from https://udspace.udel.edu/items/61cb7b16-2590-4cd9-9d5b-fa23284bf2a8
- Khan, M., & Qamar, U. (2018). Predicting survival on Titanic by applying exploratory data analytics and machine learning techniques. International Journal of Computer Applications, 179(44), 18-25. https://doi.org/10.5120/ijca2018916972
- Rigatti, S. J. (2017). Random forest. Journal of Insurance Medicine, 47(1), 31–39. https://doi.org/10.17849/insm-47-01-31-39.1

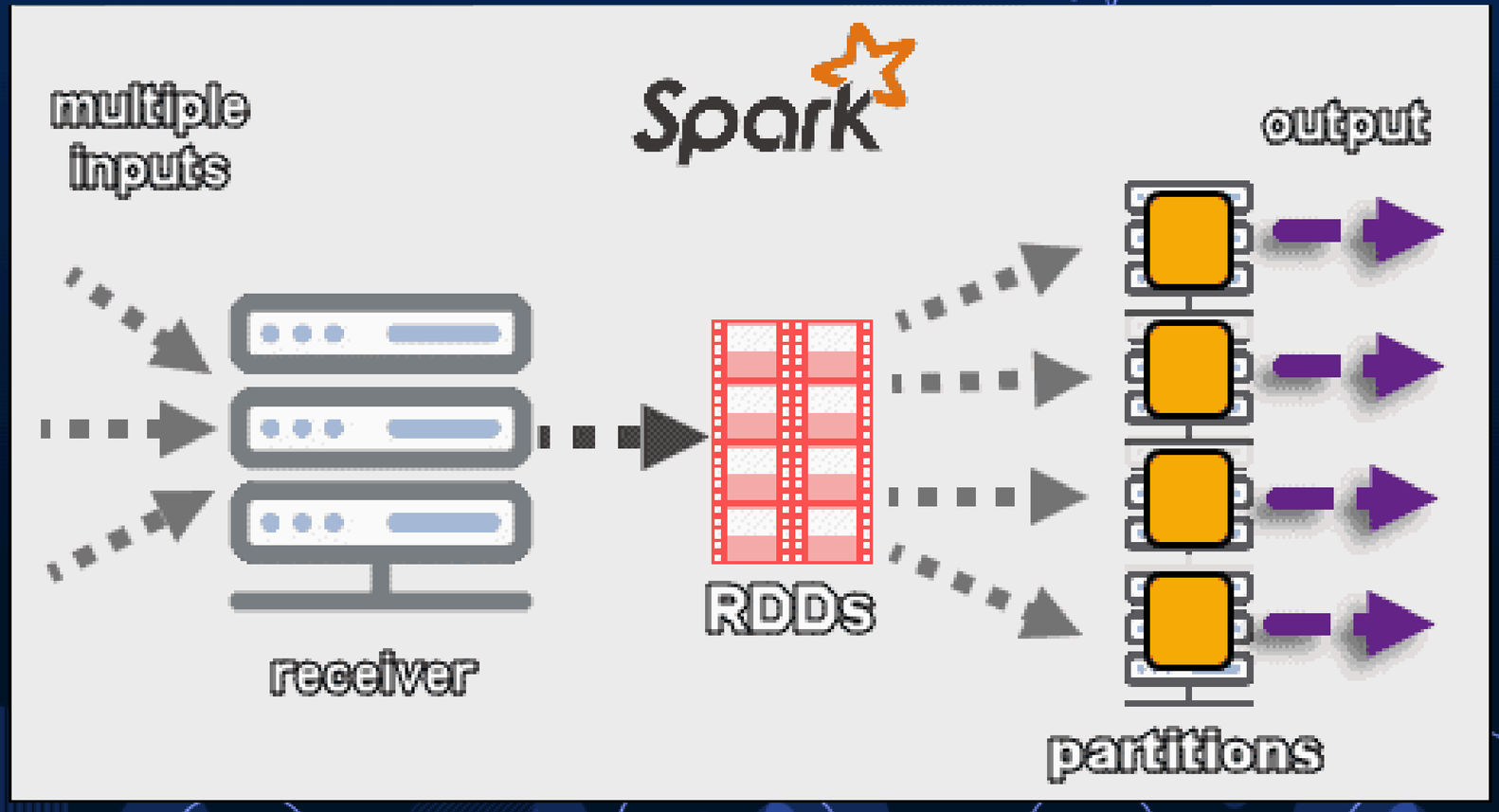
Thank you

Video: https://youtu.be/aYYxbzGhE_Q

Code:

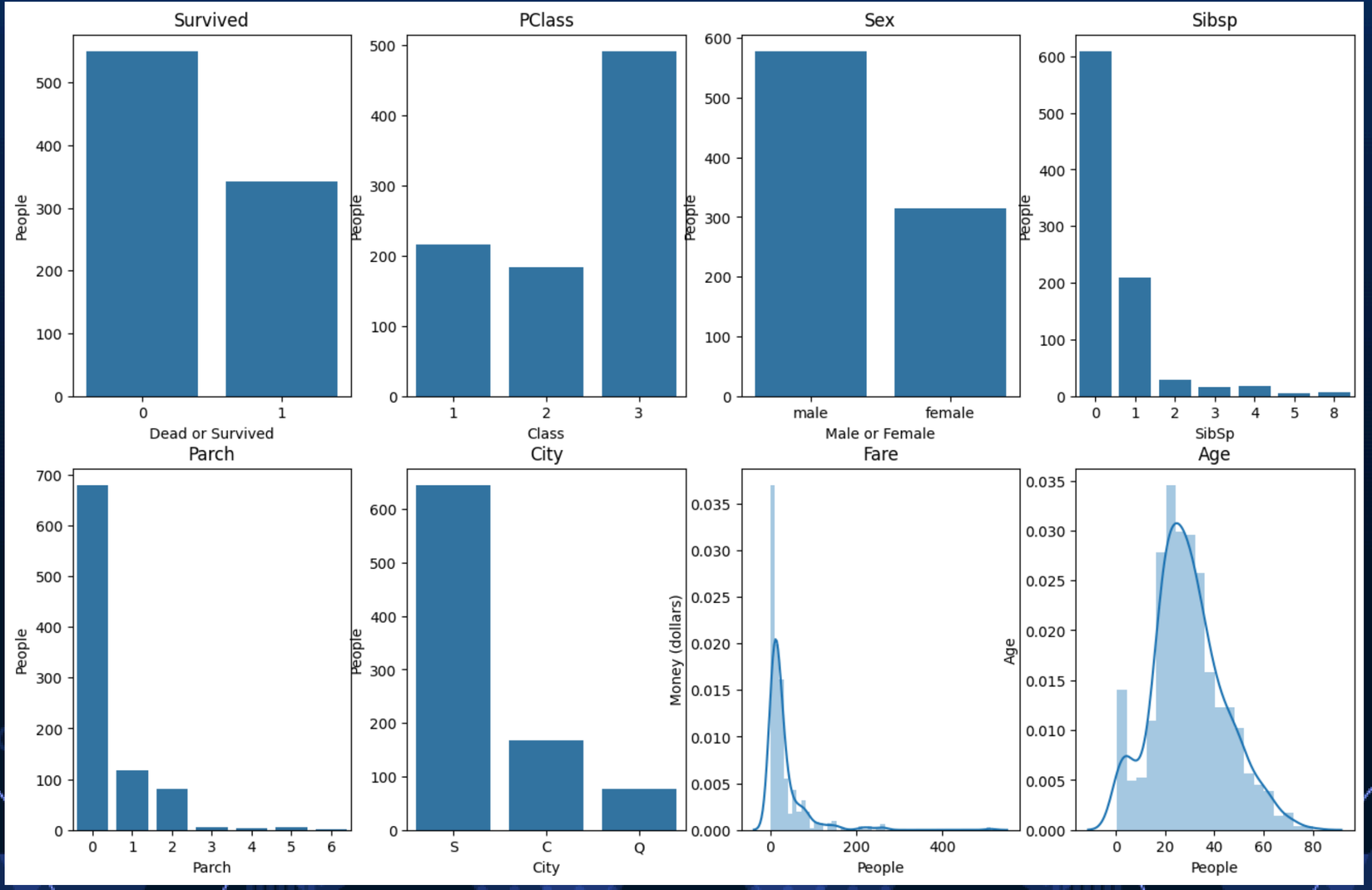
Big Data Aproach:

Big Data Aproach

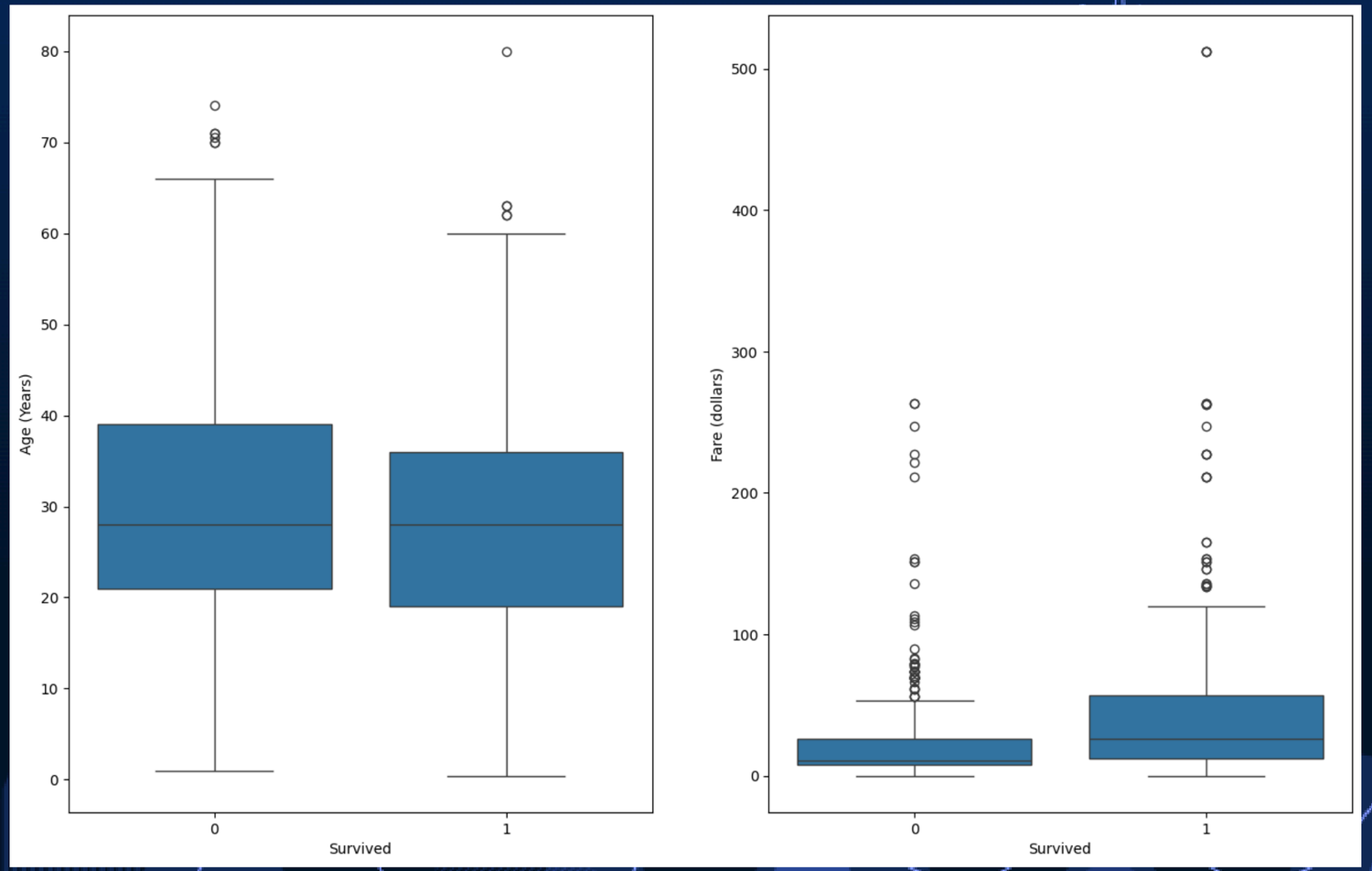


PySpark

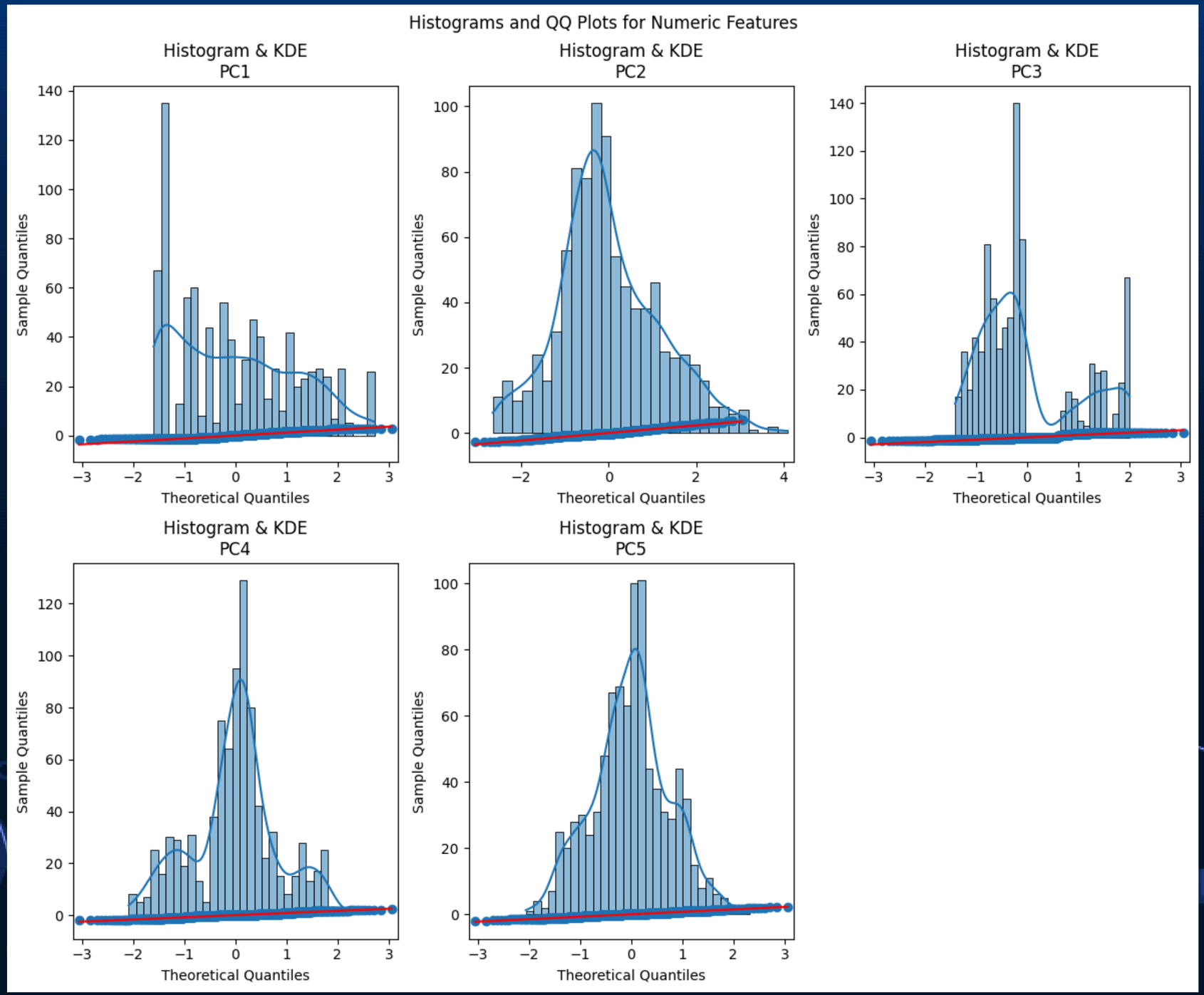
Data Distribution



Data Distribution Age & Fare



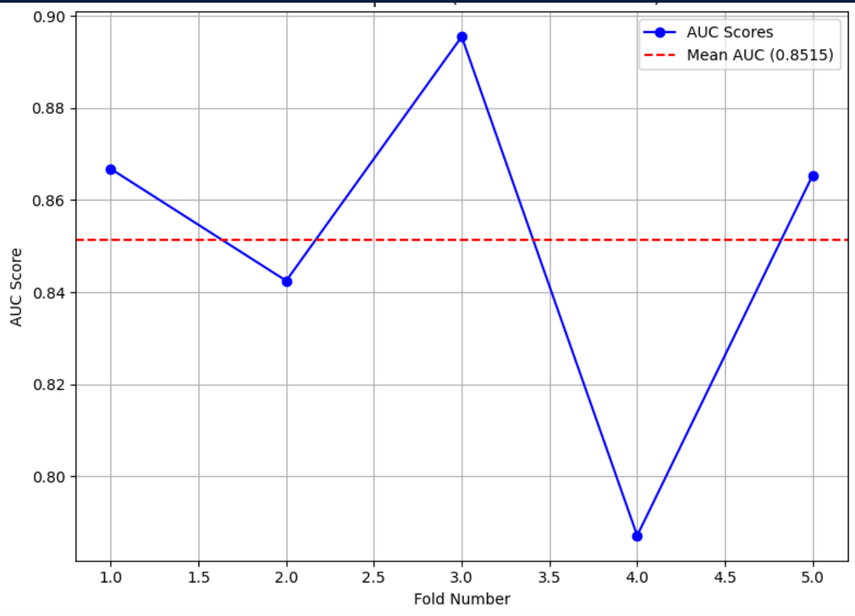
We can conclude from here:
Just a few outliers from older people.
Cause of the IQR is similar, that told us that the age is not a determinant data for the survivors.
Passengers for first class, had more survivance probability
Younger people survived more.



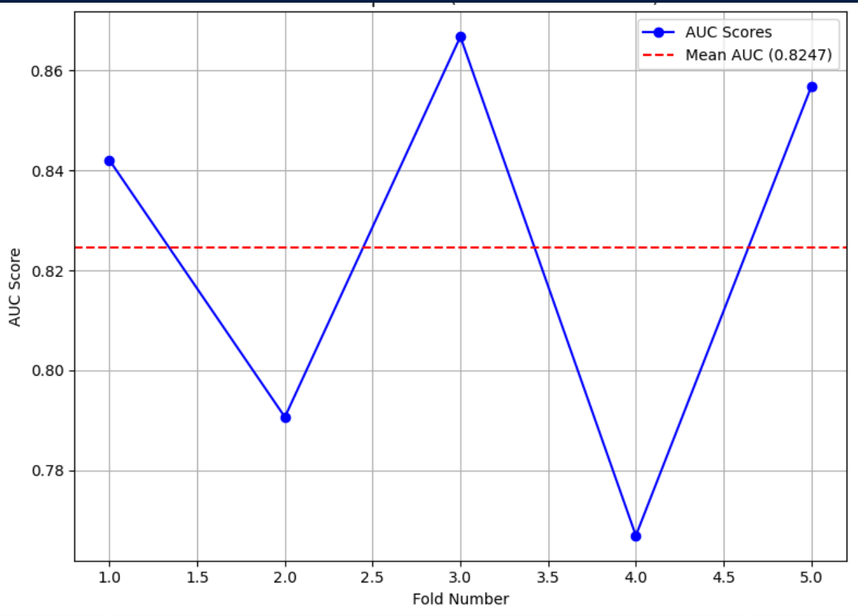
Histograms and QQ Plots for Numeric Features

Comparison AUC Score Per Fold

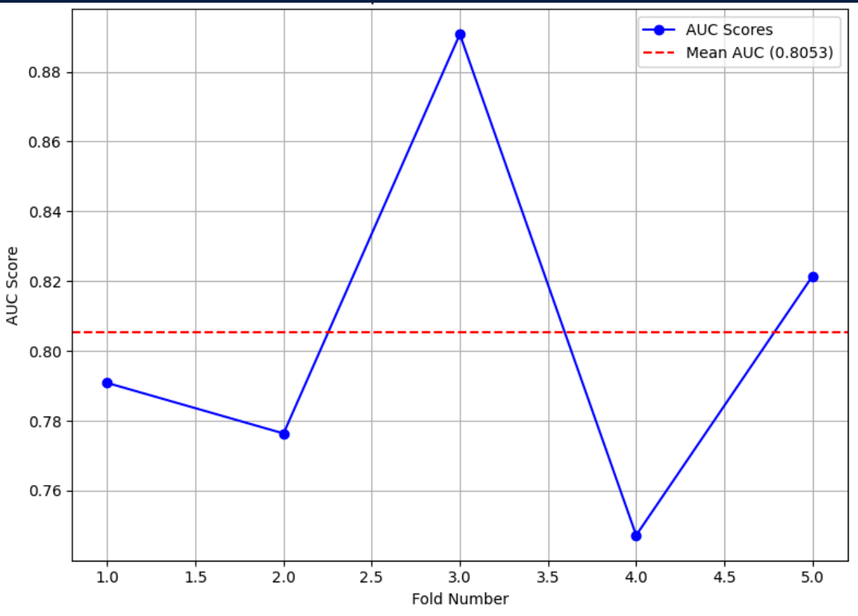
Random Forest



Logistic Regresion

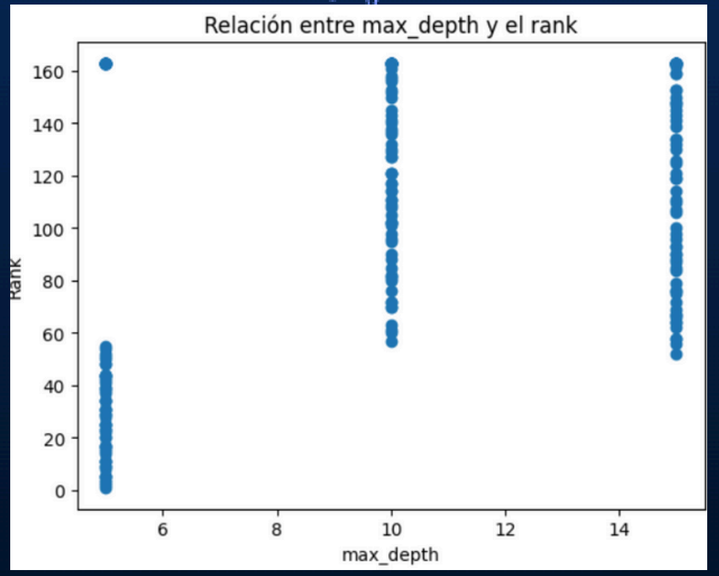


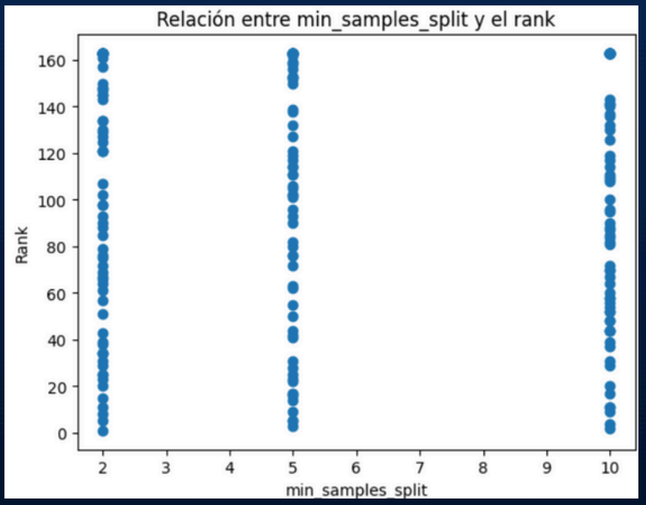
Support Vector Machine

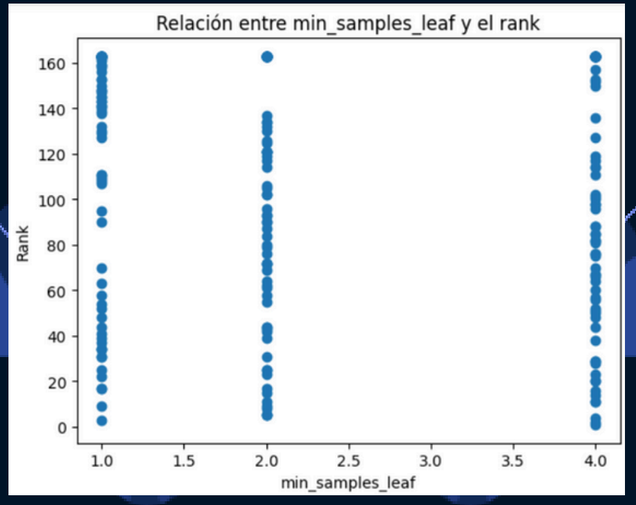


Anexos


Relación entre n_estimators y el rank
Rank
n_estimators


Relación entre max_depth y el rank
Rank
max_depth

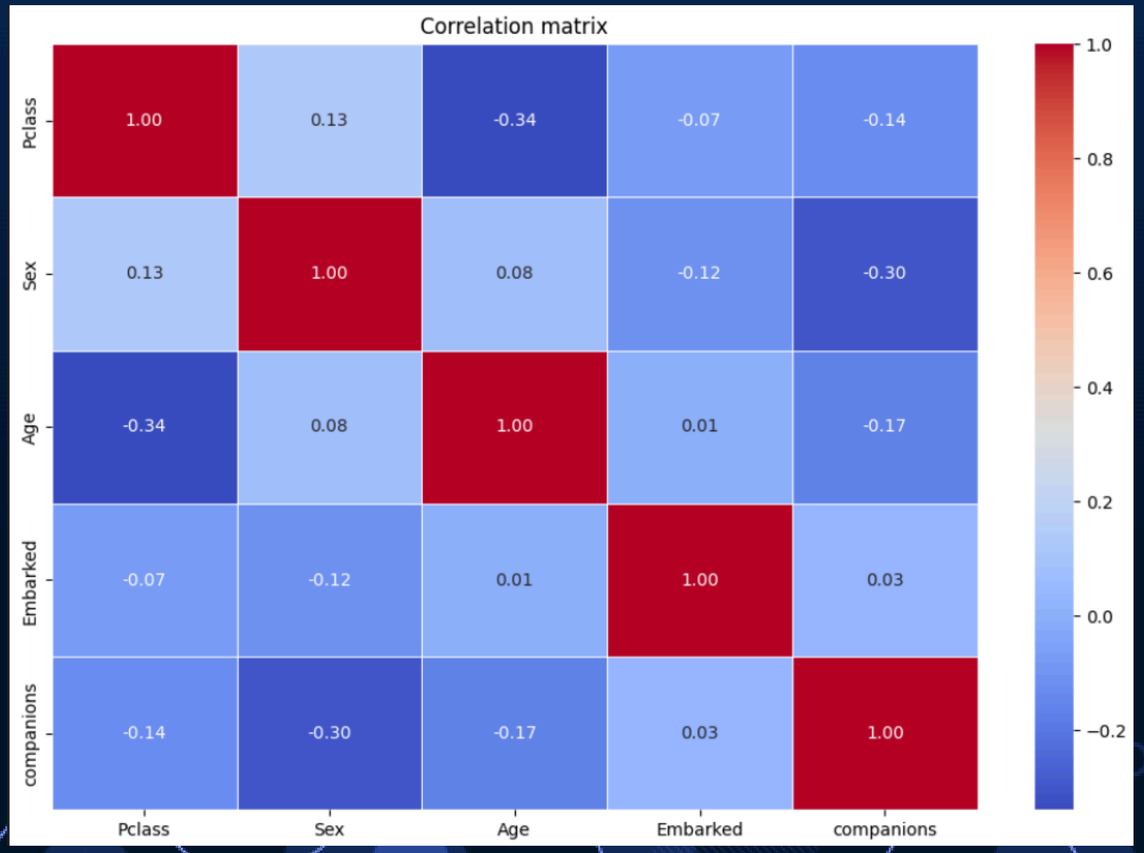

Relación entre min_samples_split y el rank
Rank
min_samples_split


Relación entre min_samples_leaf y el rank
Rank
min_samples_leaf

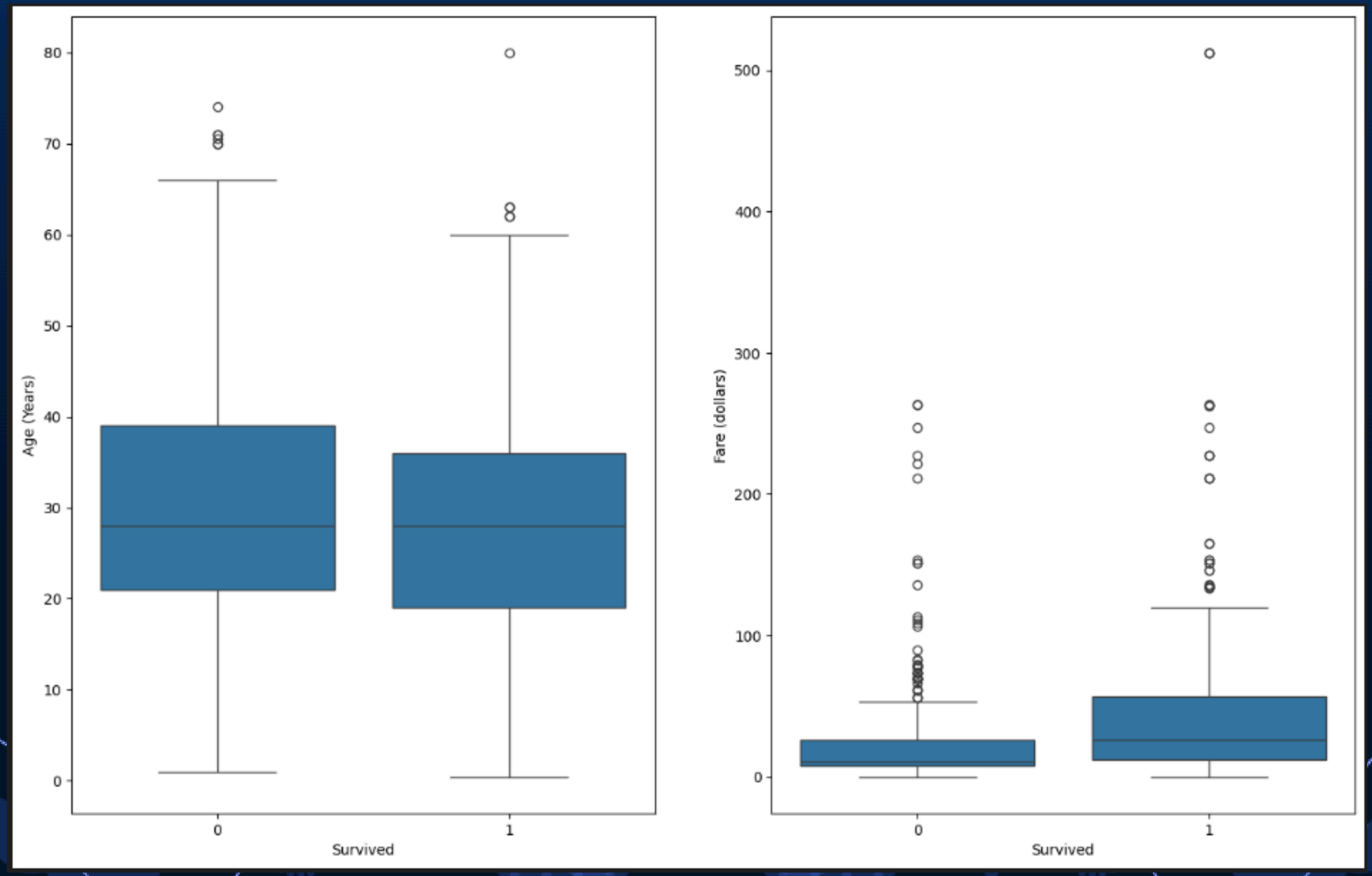
Anexos

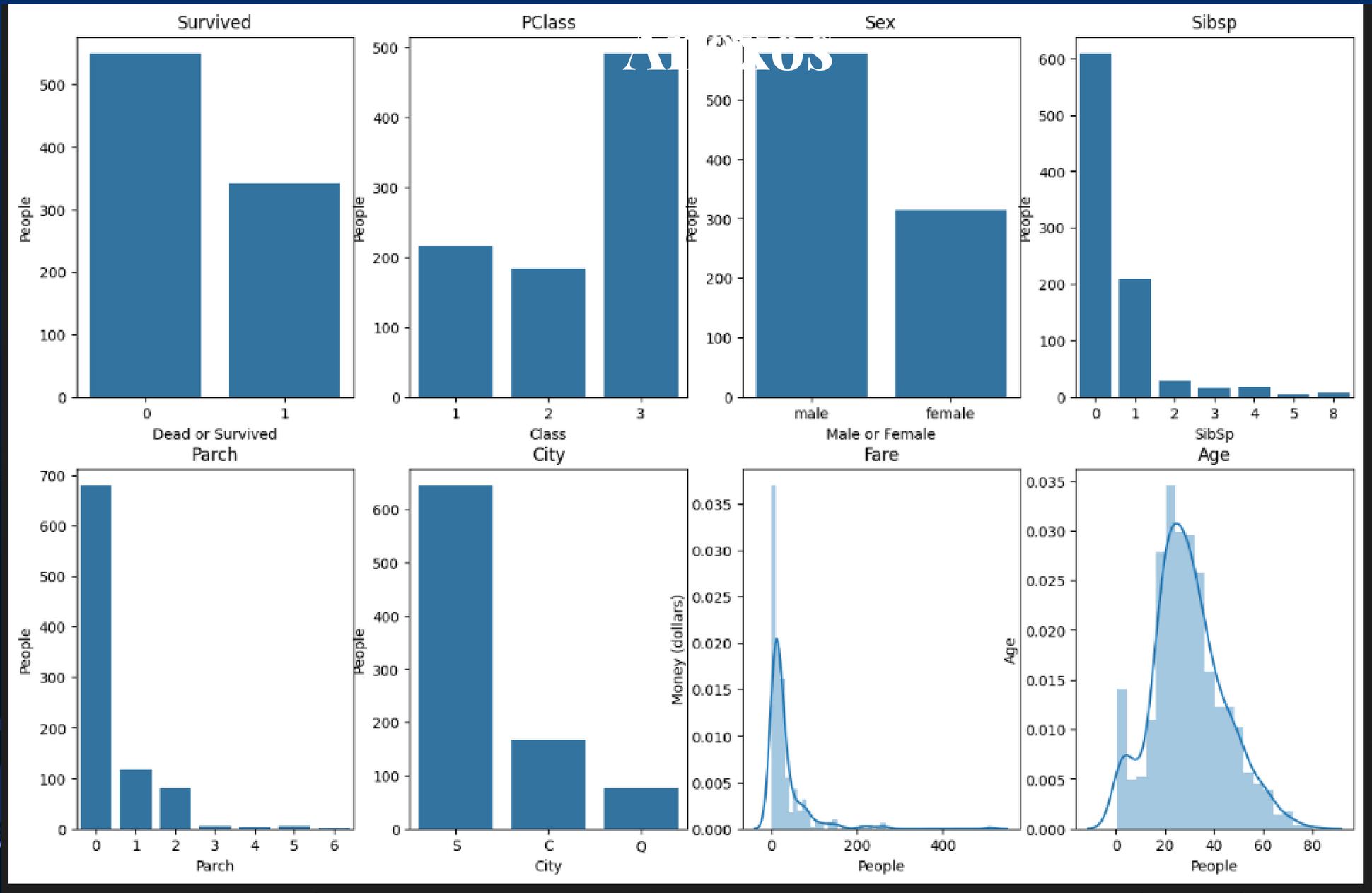
```
data_frame
✓ 0.0s
```

	Survived	Pclass	Sex	Age	Embarked	companions
0	0	3	1	22.0	0	1
1	1	1	0	38.0	1	1
2	1	3	0	26.0	0	0
3	1	1	0	35.0	0	1
4	0	3	1	35.0	0	0
...	...	...	...	...	...	...
886	0	2	1	27.0	0	0
887	1	1	0	19.0	0	0
888	0	3	0	28.0	0	1
889	1	1	1	26.0	1	0
890	0	3	1	32.0	1	0

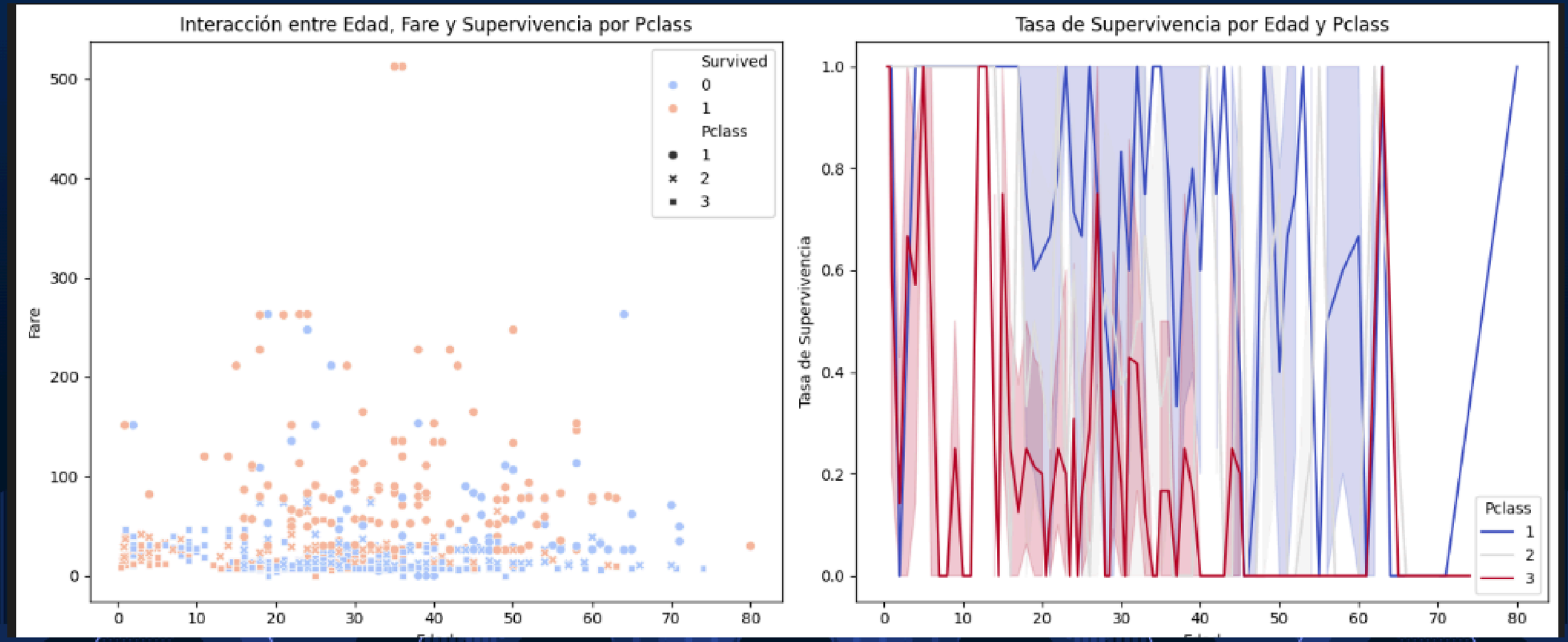


Anexos


Age (Years)
80
70
60
50
40
30
20
10
0
0
1
Survived
Fare (dollars)
500
400
300
200
100
0
0
1
Survived

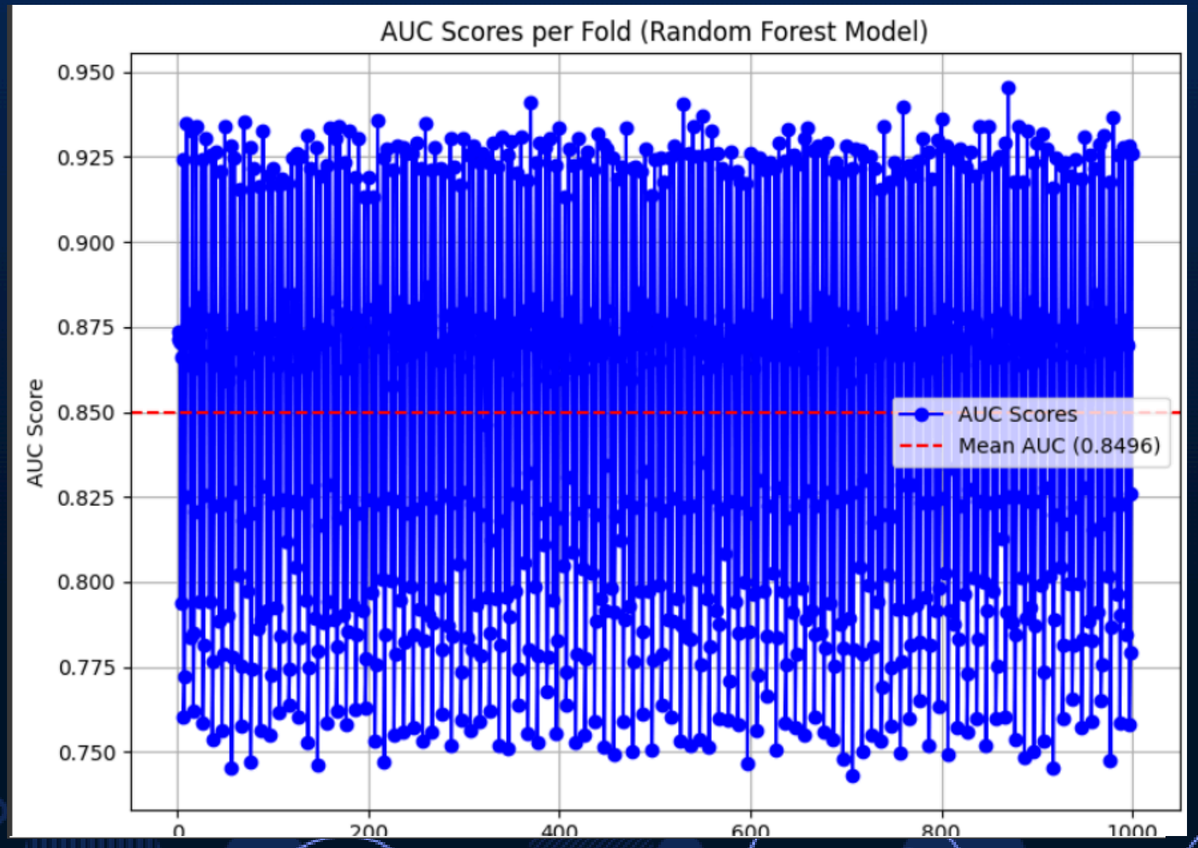

Survived
People
Dead or Survived
PClass
People
Class
Sex
People
Male or Female
male
female
Sibsp
People
SibSp
Parch
People
Parch
City
People
City
S
C
Q
Fare
Money (dollars)
People
Age
Age
People

Anexos

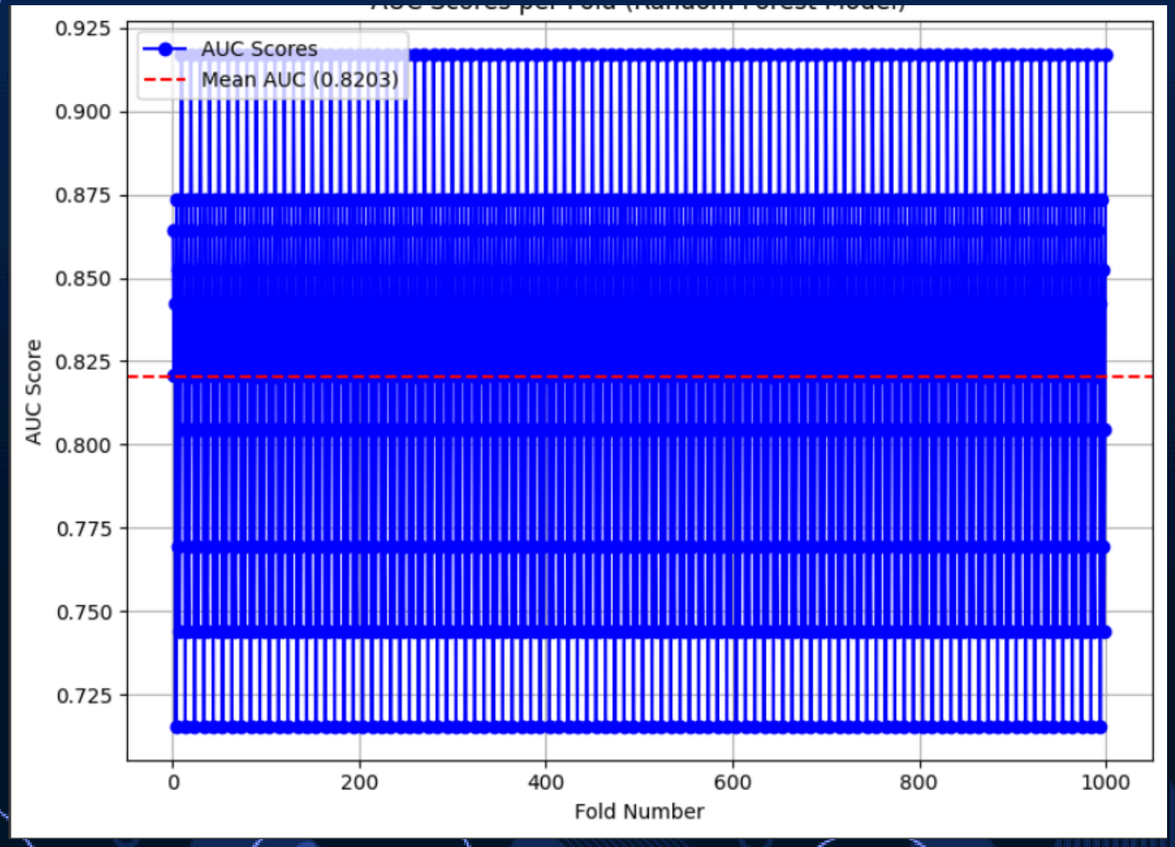

Interacción entre Edad, Fare y Supervivencia por Pclass
Survived
0
1
Pclass
1
2
3
Fare
0
100
200
300
400
500
0
10
20
30
40
50
60
70
80
Tasa de Supervivencia por Edad y Pclass
Tasa de Supervivencia
0.0
0.2
0.4
0.6
0.8
1.0
Pclass
1
2
3

Anexos

RF

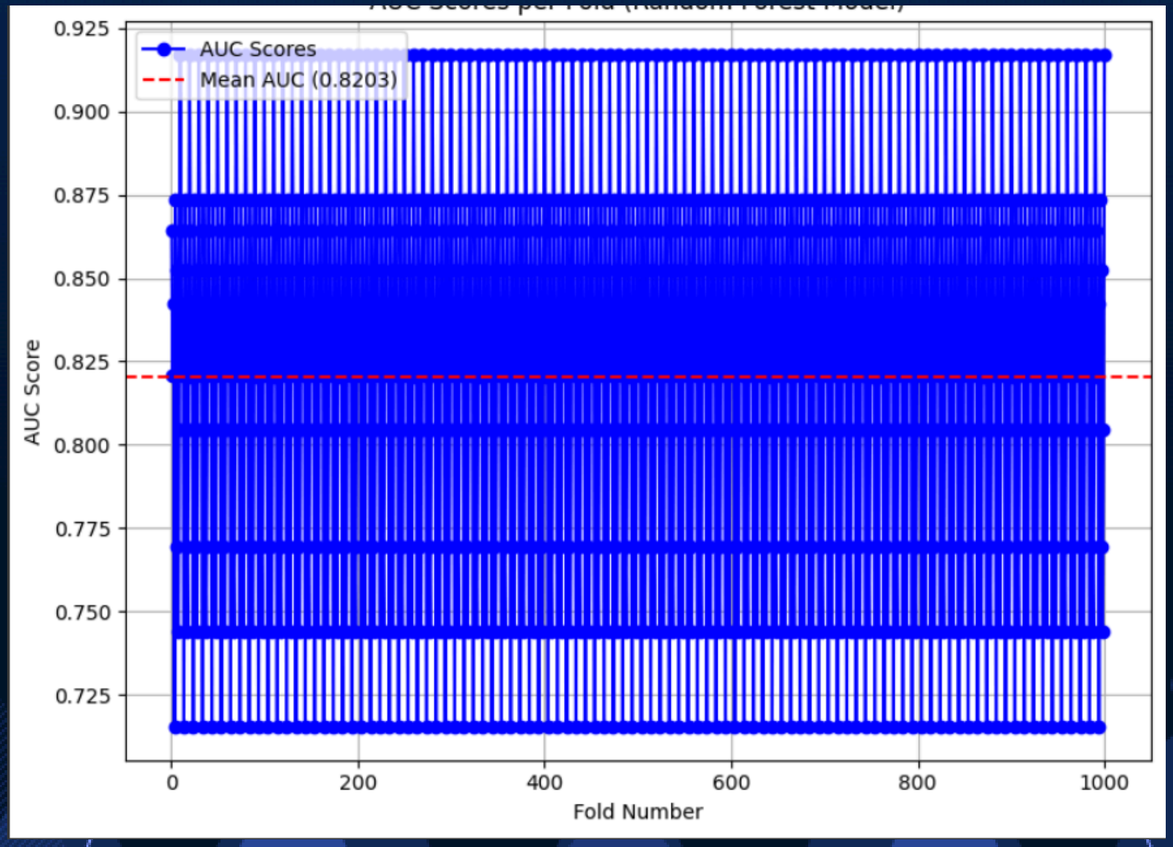

AUC Scores per Fold (Random Forest Model)
AUC Score
0.950
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0
200
400
600
800
1000
AUC Scores
Mean AUC (0.8496)

LR


AUC Score
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0.725
0
200
400
600
800
1000
Fold Number
AUC Scores
Mean AUC (0.8203)

Anexos

SVM


AUC Scores
Mean AUC (0.8203)
AUC Score
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0.725
0
200
400
600
800
1000
Fold Number