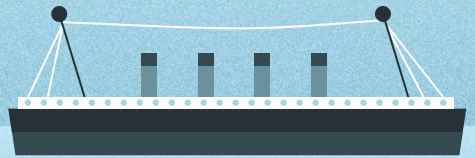
— Artificial Intelligence—

TITANIC

Assignment No. 2
Supervised Learning



Gonçalo Matias, up202108703
Fernando Afonso, up202108686
Tiago Simões up202108857

— Work Specification —

The task at hand is to analyze and understand the factors that influenced survival rates among passengers during the tragic sinking of the RMS Titanic on April 15, 1912.

Dataset attributes:

- **Pclass:** Ticket class indicating the socio-economic status of the passenger.
- **Survived:** If the passenger survived (1) or not (0).
- **Name:** Full name including title (e.g., Mr., Mrs., etc.).
- **Sex:** Gender of each passenger.
- **Age:** Age of each passenger in years.
- **SibSp:** Number of siblings or spouses aboard the Titanic for the respective passenger.
- **Parch:** Number of parents or children aboard the Titanic for the respective passenger.
- **Ticket:** The ticket number assigned to the passenger.
- **Fare:** Paid by the passenger for the ticket.
- **Cabin:** Cabin number assigned to the passenger, if available.
- **Embarked:** The port of embarkation for the passenger
- **Boat:** If the passenger survived, this column contains the identifier of the lifeboat they were rescued in.
- **Body:** If the passenger did not survive, this column contains the identification number of their recovered body, if applicable.
- **Home.dest:** The destination or place of residence of the passenger.

— Related Work/References —

1. Scikit Library - https://scikit-learn.org
2. https://www.kaggle.com/datasets/sakshisatre/titanic-dataset
3. Geeksforgeeks (K-NN) - https://www.geeksforgeeks.org/k-nearest-neighbours/
4. https://www.ibm.com/topics/supervised-learning

kaggle

— Tools and Algorithms —

- **Programming Language:** Python
- **Python Libraries:** Pandas, NumPy, Matplotlib, Scikit-learn, Seaborn
- **Development Environment:** Jupyter Notebook | VSCode | PyCharm

- **Machine Learning Algorithms:**
  - ✓ Decision Trees
  - ✓ Neural Networks
  - ✓ K-Nearest Neighbors
  - ✓ Support Vector Machine

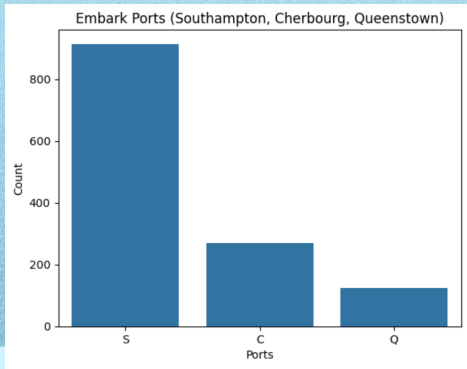
jupyter

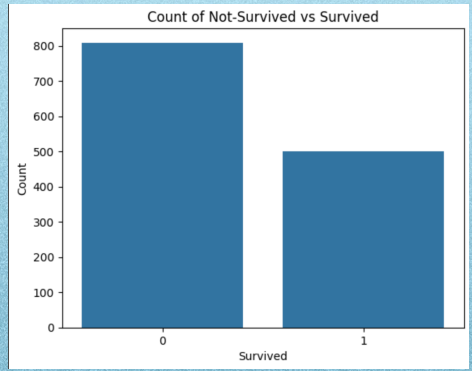
python™

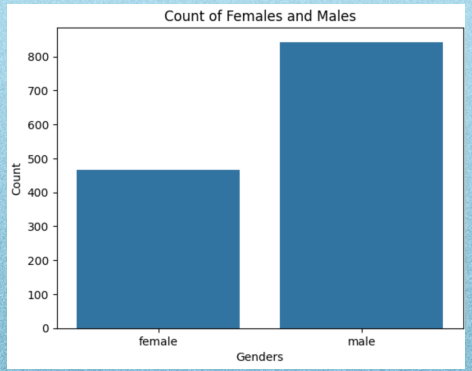
— Data Analysis —

- Create DataFrame to read data
- Data Description
- Number of duplicates, missing values, data types

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.00	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.92	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1	0	Allison, Miss. Helen Loraine	female	2.00	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.00	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.00	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON







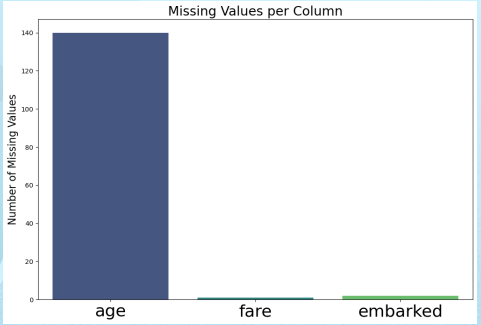

```
Data types of each column:
 pclass         int64
survived        int64
name           object
sex            object
age           float64
sibsp           int64
parch           int64
ticket         object
fare          float64
cabin          object
embarked       object
boat           object
body          float64
home.dest      object
dtype: object
```

- **Number of Instances**: 1309
- **Number of Attributes**: 14

— Data Preparation (½)—

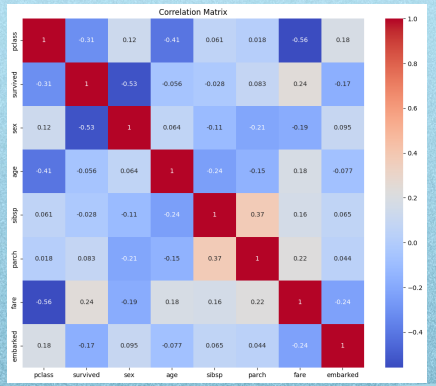
Removing Columns:

Name, Ticket, Home.dest, Body, Cabin and **Boat** were discarded as they were not interesting to bring into our analysis. We also cleaned any duplicate values.



Handling Missing Values and Outliers:

Age was the column with most missing values. This was solved by filling them with the median age. The outliers detected in the **Age, Sibsp, Parch,** and **Fare** columns were capped with the IQR method.



Correlation Analysis:

We could infer that social-economic status, represented by **pclass** and **fare**, played a significant role in survival. Additionally, gender had a notable impact on survival rates, with females having a higher chance of survival.

— Data Preparation (2/2) —

Feature Engineering:

Combining features such as **sibsp** and **parch** to create a family size feature can be particularly useful. Additionally, binning the **age** variable into age groups can help in capturing non-linear relationships more effectively.

	pclass	survived	sex	age	fare	embarked	family_size	age_group
0	1	1	0	29.0	66.3438	2	1.0	0
1	1	1	1	2.5	66.3438	2	2.0	1
2	1	0	0	2.5	66.3438	2	2.0	1
3	1	0	1	30.0	66.3438	2	2.0	0
4	1	0	0	25.0	66.3438	2	2.0	0

Data Splitting:

This ensures that we can train the models on one subset of the data and evaluate them on another, which helps in assessing their generalization performance.

The training dataset **X_train** has 1047 samples and 7 features, and the test dataset **X_test** has 262 samples and 7 features. This translates into a **80-20%** division between the two.

— Algorithms Implemented —

Decision Trees:

- Model built from the **DecisionTreeClassifier** from **sklearn.tree**
- **Max_depth** = 5 and **random_state** = 42
- Easy to understand and interpret, require relatively little data preprocessing and are able to handle multi-output problems.

Neural Network:

- Used tensorflow **Sequential** model and **StandardScaler** from sklearn.
- Can model **complex**, non-linear relationships; can be **fine-tuned** through various parameters such as the number of layers, neurons, etc.

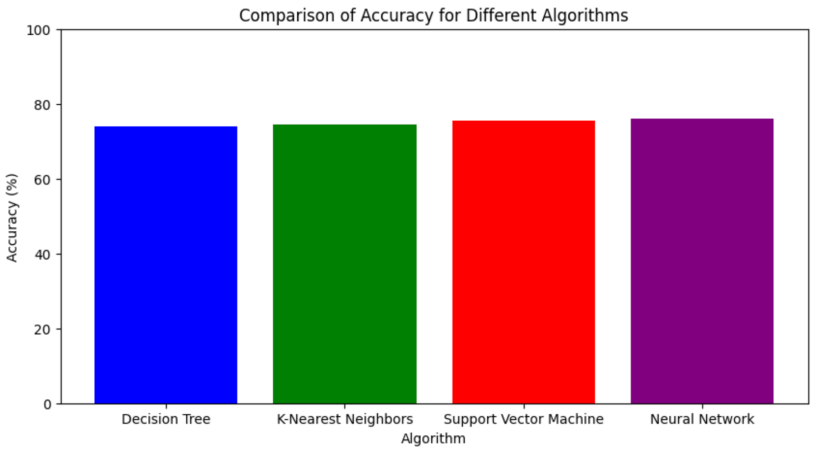
K-Nearest Neighbours:

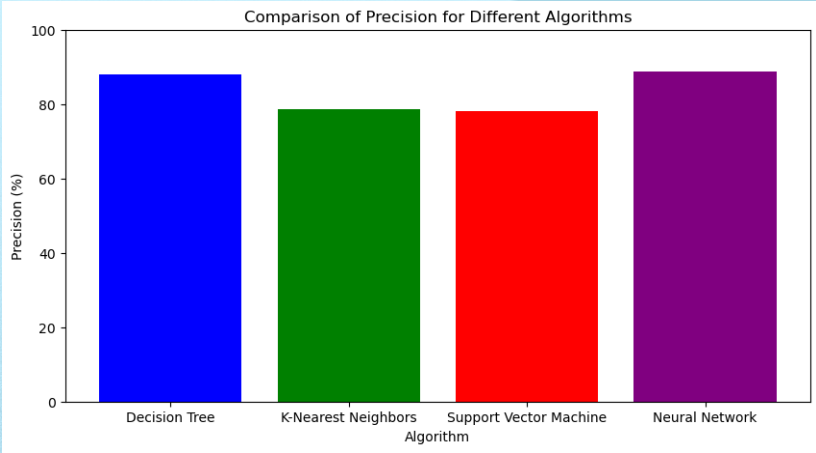
- Built from **KNeighborsClassifier** of **sklearn.neighbors**
- **n_neighbors** = 5
- **Less likely** to **overfit** than individual decision trees; provides feature importance scores.

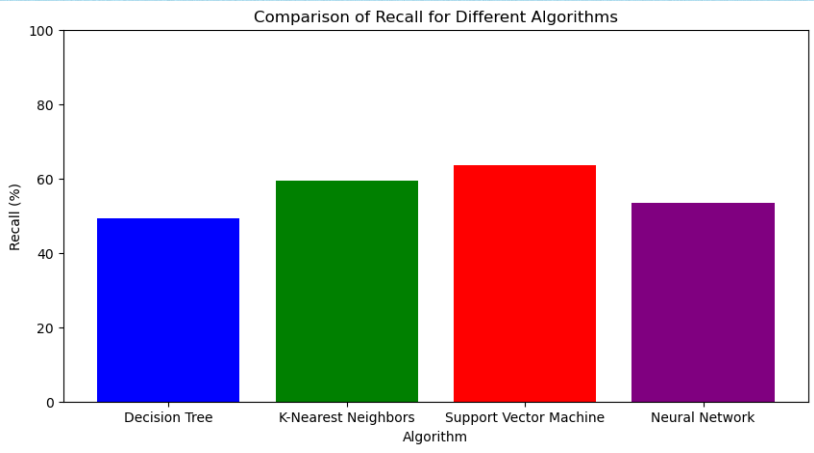
Support Vector Machine:

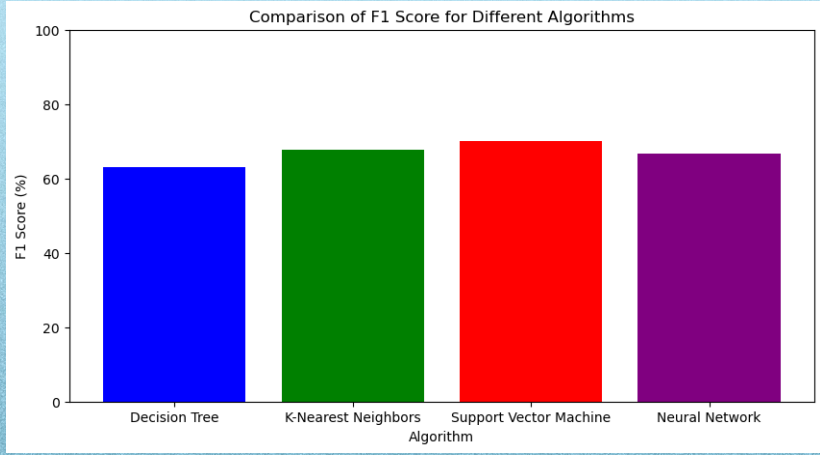
- From **SVC** of the library **sklearn.svm**
- **Kernel** = 'linear', **random_state**=42
- Uses a subset of training points (support vectors), so it's also **memory efficient**; **versatile**, as it uses different Kernel functions

— Results Obtained —









— Conclusions —

- The **Decision Tree** algorithm demonstrated the fastest testing time, making it efficient for rapid predictions.
- The **Support Vector Machine** excelled in recall, indicating its effectiveness in identifying true positive cases.
- **Neural Network** and **Decision Tree** showed higher precision, beneficial for tasks requiring high accuracy in positive predictions.

The project's findings highlight the importance of selecting the appropriate algorithm based on specific performance requirements.

Decision Tree is suitable for scenarios demanding quick predictions, while Support Vector Machine is advantageous for tasks where identifying positive cases is critical. Neural Networks offer a balanced approach with relatively high precision.