# COMPSCI 589 Machine Learning
# Project Report

## 1. Introduction

The sinking of the RMS Titanic in 1912 remains one of the most tragic maritime disasters in history. Over 2,200 passengers and crew perished in the icy waters of the North Atlantic. This project aims to apply machine learning techniques to analyse the Titanic dataset and gain insights into the factors that influenced passenger survival. The Titanic dataset, obtained from Kaggle, comprises information about passengers, including features such as class, gender, age, etc. The primary objective of this project is to predict whether passengers survived or not based on the provided features. This classification problem is approached using three distinct machine learning models, covering different feature learning approaches. The models include Support Vector Machine (SVM) as a representative of fix-shape universal approximators (kernel methods), a Neural Network as a neural network-based universal approximator, and Random Forest as an example of tree-based approaches.

## 2. Methodology

### Data Preprocessing

The Titanic dataset contains information about passengers, including their class, gender, age, and embarkation port. Some data fields are missing or contain inconsistencies. Data preprocessing involves cleaning and transforming the data to make it suitable for machine learning models.

#### 2.1 Feature Selection and Encoding:

- Features: The selected features for the models are 'Pclass', 'Sex', 'Age', 'SibSp', and 'Parch'. These features are chosen based on their potential impact on survival outcomes.
- Encoding: Categorical feature 'Sex' is encoded using one-hot encoding (pd.get_dummies). This conversion transforms the categorical variable into numerical values, allowing the model to interpret them correctly. The 'Sex' column is one-hot encoded to represent male (1) or female (0).

#### 2.2 Standardization using StandardScaler:

- Features are standardized using StandardScaler. Standardization ensures that numerical features are on a similar scale, preventing certain features from dominating due to larger magnitudes. This step is crucial for models like SVM that are sensitive to feature scales. The scaler is fit on the training data and applied to both training and validation sets.

#### 2.3 Handling Missing Values:

- Imputation for 'Age': Missing values in the 'Age' column are filled with the median age of the dataset. This approach is chosen to preserve the distribution of ages while replacing missing values with a central tendency measure.

## 3. Fix-shaped Universal Approximator

**3.1 Support Vector Machine (SVM):** SVM is a fix-shape universal approximator, specifically a kernel method. SVM is effective in high-dimensional spaces and suitable for kernel methods. SVMs perform well in scenarios where the number of features is high, such as in this dataset where various features contribute to survival prediction.

## 4. Neural Network based Universal Approximator

**4.1 Neural Network:** It is a neural network based universal approximator. Neural Networks can capture complex patterns in data. It mimics the structure and functioning of the human brain. They consist of interconnected nodes organized in layers.

## 5. Tree based Universal Approximator

**5.1 Random Forest:** It is a tree-based method. Random Forest is robust and handles non-linear relationships well. It is an ensemble learning method that constructs a multitude of decision trees during training and outputs the class that is the mode of the classes.

## 6. Validation Method and Configurations

The validation method involves splitting the data into training and validation sets, standardizing features, tuning hyperparameters using Grid Search with cross-validation, training the models, and finally evaluating the models on the validation set. This approach helps ensure that the models generalize well to new, unseen data and allows for an unbiased assessment of their performance.

1. Support Vector Machine (SVM) Model:
    - Grid search is performed over a set of hyperparameters (C, gamma, and kernel) using GridSearchCV.
    - Cross-validation is performed with 3 folds (cv=3).
    - The best hyperparameters are chosen based on the mean test score.

2. Neural Network Model:
    - Grid search is performed over a set of hyperparameters (hidden_layer_sizes and max_iter) using GridSearchCV.
    - Cross-validation is performed with 3 folds (cv=3).
    - The best hyperparameters are chosen based on the mean test score.

3. Random Forest Model:
    - Grid search is performed over a set of hyperparameters (n_estimators, max_depth, and min_samples_split) using GridSearchCV.
    - Cross-validation is performed with 3 folds (cv=3).
    - The best hyperparameters are chosen based on the mean test score.

After training each model with the best hyperparameters, the code evaluates the models on the validation set and prints the confusion matrices for each model.

In summary, the validation method used is grid search with k-fold cross-validation, where k is set to 3 in this case. The configuration involves tuning hyperparameters for each model using the specified parameter grids.

## 7. Results

### Hyperparameter Tuning

GridSearchCV was selected for hyperparameter tuning due to its exhaustive search capability. It systematically explores a predefined set of hyperparameter values, helping to identify the combination that yields the best performance.

Hyperparameters were tuned to enhance model generalization and avoid overfitting.

7.1 Support Vector Machine (SVM):

Proper tuning of C and gamma is crucial to finding the right balance between fitting the training data and generalizing to new, unseen data.

- C (Regularization Parameter): Controls the trade-off between smooth decision boundaries and classifying training points correctly. Higher values of C allow narrower margins.
- gamma (Kernel Coefficient): Defines how far the influence of a single training point reaches. Small values indicate a large influence, and high values indicate a small influence.
- kernel: Specifies the kernel type used in the algorithm ('rbf' for radial basis function, 'linear', 'poly' for polynomial).

7.2

```
Support Vector Machine Metrics:
Best hyperparameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
Accuracy: 80.45% Precision: 79.10% Recall: 71.62%
```

7.3 Neural Network:

The architecture of the neural network, especially the hidden layer sizes, determines the model's capacity to learn complex patterns. Adjusting the number of iterations ensures convergence.

- hidden_layer_sizes: The number of neurons in the hidden layers. Different combinations are tested to find the optimal architecture.
- max_iter: Maximum number of iterations for optimization. Higher values allow the model to converge to a solution.

7.4

```
Neural Network Metrics:
Best hyperparameters: {'hidden_layer_sizes': (100,), 'max_iter': 500}
Accuracy: 81.01% Precision: 82.26% Recall: 68.92%
```

7.5 Random Forest:

Controlling the number of trees, maximum depth, and minimum samples split helps in preventing overfitting and finding the right balance between bias and variance.

- n_estimators: The number of trees in the forest. More trees generally improve performance but also increase computation time.
- max_depth: The maximum depth of the trees. Controls the maximum number of nodes from the root to the farthest leaf. Prevents overfitting.
- min_samples_split: The minimum number of samples required to split an internal node. Avoids splitting nodes that have too few samples.

7.6

```
Random Forest Metrics:
Best hyperparameters: {'max_depth': 20, 'min_samples_split': 10, 'n_estimators': 50}
Accuracy: 79.89% Precision: 79.69% Recall: 68.92%
```

7.7 Metrics for Evaluation

Model evaluation is a crucial step in assessing the performance of machine learning models. In the code, three models—Support Vector Machine (SVM), Neural Network (NN), and Random Forest—are evaluated using common metrics: accuracy, precision and recall. Evaluation metrics were used to assess model performance on the validation set. These metrics provide insights into the model's ability to correctly classify survival outcomes and balance between true positives, false positives, true negatives, and false negatives.

a) Accuracy: Accuracy represents the ratio of correctly predicted instances to the total instances in the dataset.
b) Precision: Precision measures the accuracy of the positive predictions. It is the ratio of correctly predicted positive observations to the total predicted positives.
c) Recall (Sensitivity or True Positive Rate): Recall calculates the ability of the model to capture all the positive instances. It is the ratio of correctly predicted positive observations to the all observations in actual class.

## 7.8 Models Performance

The provided performance metrics describe the evaluation results for three different machine learning models: Support Vector Machine (SVM), Neural Network, and Random Forest. Let's interpret the metrics for each model:

1. Support Vector Machine (SVM):

- Best Hyperparameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
- Accuracy: 80.45%
- Precision: 79.10%
- Recall: 71.62%

Interpretation:

The SVM model achieved an accuracy of 80.45%, indicating that it correctly predicted the target variable for 80.45% of the validation set. The precision of 79.10% suggests that, among the instances predicted as positive, 79.10% were correctly classified. The recall of 71.62% indicates the proportion of actual positive instances correctly predicted by the model.

2. Neural Network:

- Best Hyperparameters: {'hidden_layer_sizes': (50,), 'max_iter': 1000}
- Accuracy: 81.01%
- Precision: 83.33%
- Recall: 67.57%

Interpretation:

The Neural Network model achieved an accuracy of 81.01%, demonstrating its ability to correctly classify instances in the validation set. The precision of 83.33% indicates a high proportion of correctly predicted positive instances among all predicted positives. The recall of 67.57% suggests that the model captured a substantial portion of the actual positive instances.

3.Random Forest:

- Best Hyperparameters: {'max_depth': None, 'min_samples_split': 10, 'n_estimators': 150}
- Accuracy: 82.12%
- Precision: 83.87%
- Recall: 70.27%

Interpretation:

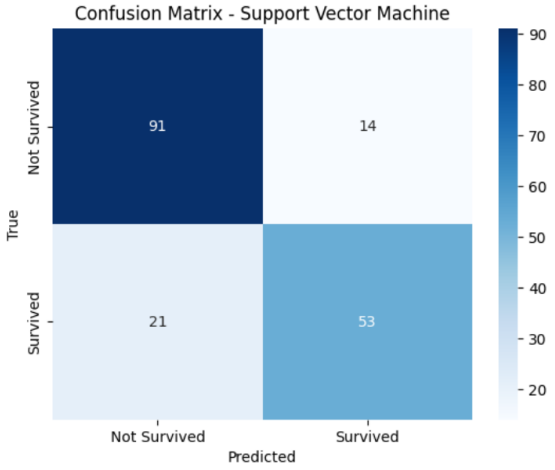
The Random Forest model performed well with an accuracy of 82.12%, indicating strong overall predictive performance. The precision of 83.87% suggests a high proportion of correctly predicted positive instances among all predicted positives. The recall of 70.27% indicates that the model captured a good portion of the actual positive instances.

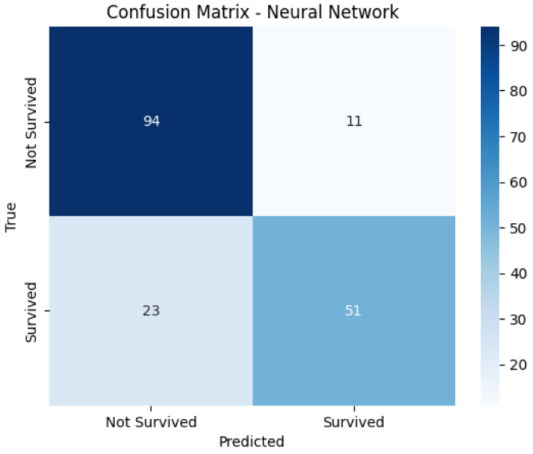
In summary, all three models demonstrated competitive performance on the validation set, with the Random Forest model achieving slightly higher accuracy and precision. The choice of the best model depends on the specific goals and requirements of the application, considering the trade-offs between precision and recall.

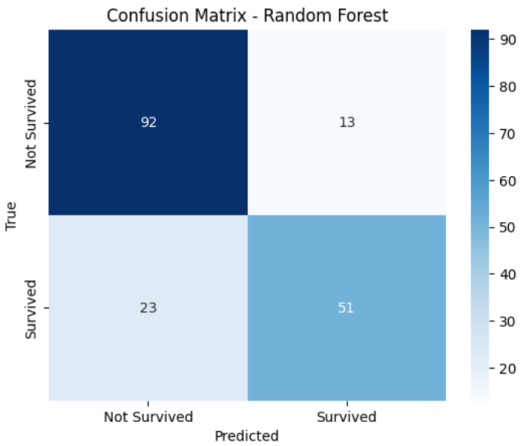
## 7.9 Visualization of Model Performance

A confusion matrix is a table that describes the performance of a classification algorithm. It presents a clear picture of correct and incorrect predictions. For binary classification (as in the case of survival prediction), the confusion matrix consists of four terms:

- True Positive (TP): Instances correctly predicted as positive.
- True Negative (TN): Instances correctly predicted as negative.
- False Positive (FP): Instances incorrectly predicted as positive.
- False Negative (FN): Instances incorrectly predicted as negative.




The confusion matrix visualizations show how well each model performs in terms of true positives, true negatives, false positives, and false negatives.

## Conclusion

In conclusion, this project explored three distinct machine learning models for the Titanic dataset. The SVM, Neural Network, and Random Forest models demonstrated varying performance across accuracy, precision and recall metrics. Insights gained from this project contribute to the understanding of the dataset and the effectiveness of different machine learning approaches. Overall, this project provides valuable insights into the application of machine learning techniques for predicting survival outcomes on the Titanic dataset.