ASSIGNMENT COVERSHEET

UTS: ENGINEERING & INFORMATION TECHNOLOGY

SUBJECT NUMBER & NAME	NAME OF STUDENT(s) (PRINT CLEARLY)	STUDENT ID(s)
SAS Business Predictive modelling 420050	Kaung Khant Kyaw JunHao Zeng William Wickham Zichen Shao Michael Zervos	14447837 25181811 26005292 13991866 2450741

STUDENT EMAIL	STUDENT CONTACT NUMBER
kaungkhantkyaw@student.uts.edu.au JunHao.Zeng-1@student.uts.edu.au william.wickham@student.uts.edu.au zichen.shao-1@student.uts.edu.au Michael.zervos@student.uts.edu.au	

NAME OF TUTOR	TUTORIAL GROUP	DUE DATE
Dr. David Hason Rudd	5	17/10/2025```

ASSESSMENT ITEM NUMBER & TITLE
Assignment 2: Predictive Modelling

I acknowledge that if AI or another nonrecoverable source was used to generate materials for background research and self-study in producing this assignment, I have checked and verified the accuracy and integrity of the information used.
I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.
I confirm that I have read, understood and followed the advice in the Subject Outline about assessment requirements. I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Declaration of originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I have rewritten any material provided by AI or other nonrecoverable sources and where appropriate acknowledged their contribution. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.

No content generated by AI technologies or other sources has been presented as my own work and I have rewritten any text provided by AI or other sources in my own words.

Statement of collaboration:

This project was completed collaboratively, with all members contributing to the design, development, and evaluation of the predictive models using SAS Viya. Each participant actively engaged in data preparation, model configuration, parameter adjustment, and interpretation of performance results. The team worked collectively to ensure that decisions regarding variable selection, model comparisons, and conclusions were made through shared discussion and consensus. All work presented reflects a fair and balanced group effort carried out with academic integrity and adherence to ethical collaboration standards.

Signature of student(s)K.K, Z.C, J.Z, W.W, M.Z Date 17/10/2025

Commented [1]: maybe put your initials here

Table of Contents

Executive Summary

This report presents a comprehensive comparative analysis of machine learning models for predicting passenger survival on the RMS Titanic, utilizing the SAS platform. The primary objective of this study is to identify the most effective predictive model by systematically building, tuning, and evaluating six distinct algorithms: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine (SVM), and Artificial Neural Network.

The methodology involved initial data exploration and preprocessing of the Titanic dataset to handle missing values and engineer relevant features. Subsequently, each of the six models was implemented and subjected to a rigorous hyperparameter tuning process to optimize its predictive accuracy and robustness. The performance of these optimized models was then benchmarked against one another using key classification metrics.

This research aims to not only develop a reliable model for this classic binary classification problem but also to provide a clear comparison of the relative strengths and weaknesses of these widely-used machine learning techniques as applied to this dataset. The findings detailed in this report will highlight the superior model and offer insights into the factors most critical to survival within the Titanic dataset, thereby demonstrating the capability of data analytics in exploring the underlying structures and relationships within a complex dataset.

Data pre-processing

Effective data pre-processing is a critical prerequisite for building robust machine learning models. This phase focuses on cleaning, transforming, and structuring the raw data to enhance model performance and ensure the quality of inputs. Our data pre-processing strategy for the Titanic dataset was implemented through a structured pipeline within the SAS environment, supplemented by an initial manual variable assessment.

Initial Variable Assessment and Selection

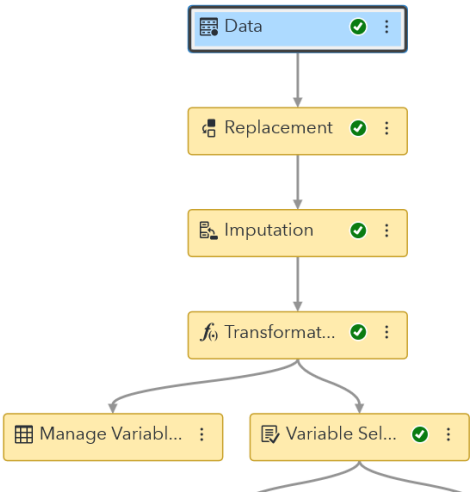
Prior to constructing the automated pipeline, an initial assessment of all variables within the dataset was conducted to determine their relevance and potential predictive power. Based on this analysis, several variables were deemed unsuitable for modeling and were consequently rejected. Specifically, the “Name”, “Ticket”, and “Cabin” variables were excluded from the input data.

The rationale for these exclusions is as follows:

- **Name and Ticket:** These variables exhibit high cardinality, meaning they have a large number of unique values. Such variables often contribute more noise than signal and can unnecessarily increase the dimensionality and complexity of the model without providing significant predictive value.
- **Cabin:** This variable contained a substantial number of missing values. While imputation is possible, the high percentage of missingness cast doubt on the reliability of any imputed values and its overall utility in the models.

By removing these variables, we aimed to reduce noise and focus the models on features with stronger predictive potential.

Data Processing Pipeline



Following the initial variable selection, a systematic data processing pipeline was established to perform the necessary cleaning and transformation tasks. The pipeline, as depicted in the SAS workflow, consists of the following key stages, summarized in the table below:

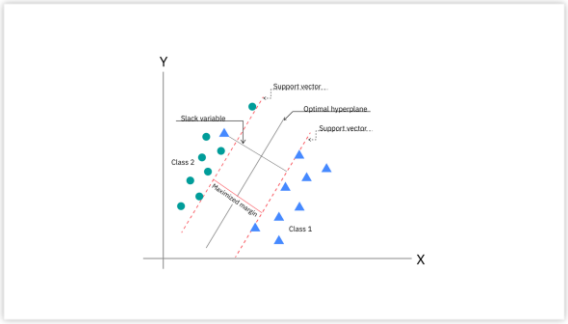
Stage Name	Description
Data Input	Imports the curated Titanic dataset, containing the pre-selected variables.
Replacement	Systematically replaces specific values within the dataset to correct inconsistencies or standardize categorical labels before further processing.
Imputation	Fills missing data entries for variables like Age and Embarked using statistical measures (e.g., mean, median, or mode) to ensure a complete dataset for model training.
Transformation	Modifies variables to better suit machine learning algorithms. This includes normalizing numerical variables (Age, Fare) and encoding categorical variables into a numerical format.
Variable Management and Selection	Allows for final adjustments to variable roles and applies automated feature selection techniques to further refine the set of predictors before modeling.

Methodology

Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm designed for both classification and regression tasks. Within SAS Viya, the SVM algorithm constructs a decision boundary which is known as a hyperplane separating data points into different classes while maximizing the margin between them (SAS Institute Inc., 2023). When applied to the Titanic dataset, where the goal is to predict survival (1 = survived, 0 = not survived), SVM effectively distinguishes between passengers who lived and those who did not by analysing key predictors such as sex, age, and passenger class.

Research using this dataset has shown that, after proper data preprocessing, SVM performs competitively with other classification algorithms and can achieve strong predictive accuracy (Hasan & Hasan, 2024; Tabbakh et al., 2021). The model relies on support vectors, the critical data points that define the optimal separating boundary, and through kernel functions, SVM can model complex non-linear relationships within the data (Van Belle et al., 2018).



Although the Titanic prediction task is a binary classification problem, similar SVM frameworks have been successfully extended to survival analysis, enabling models to predict not just outcomes but also time-to-event probabilities (Van Belle et al., 2018).

Figure 1
Support vector machine showing the optimal hyperplane and support vectors. From *Support Vector Machine* (n.d.), by IBM. Copyright IBM. https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/creative-assets/s-migr/ul/g/8f/27/3-1_svm_optimal-hyperplane_max-margin_support-vectors-2-1.png

Artificial Neural Network

Artificial Neural Networks (ANNs) are suitable for analysing the Titanic dataset because the data contains a mix of categorical and numerical variables that jointly influence the survival outcome. Variables such as age, sex, class, fare,and family size interact in complex and non-linear ways, for example, gender strongly impacts survival, but its effect is also influenced by passenger class and fare. ANNs are effective here because their layered structure can capture these interactions. Each hidden layer can learn different levels of patterns; some neurons may identify simple relationships, while others capture more subtle combinations.

Activation functions introduce non-linearity, enabling the network to represent complex decision boundaries that simpler linear models might miss. In SAS Viya, building an ANN on the Titanic dataset allows the model to weigh each feature's contribution, adjust for feature interactions, and improve predictive accuracy. This makes ANNs a strong candidate for survival prediction, complementing other models such as logistic regression and decision trees (Al-Hayik & Abu-Naser, 2023; Hasan & Hasan, 2024).

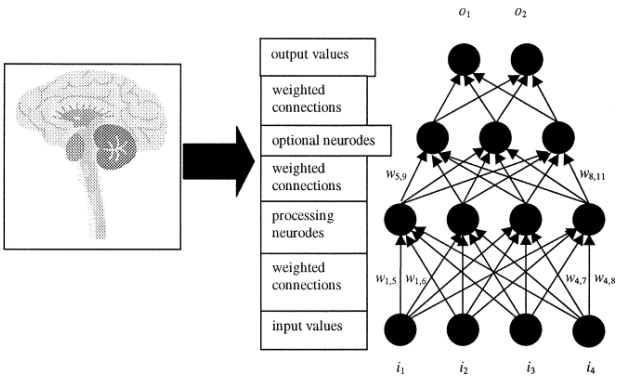
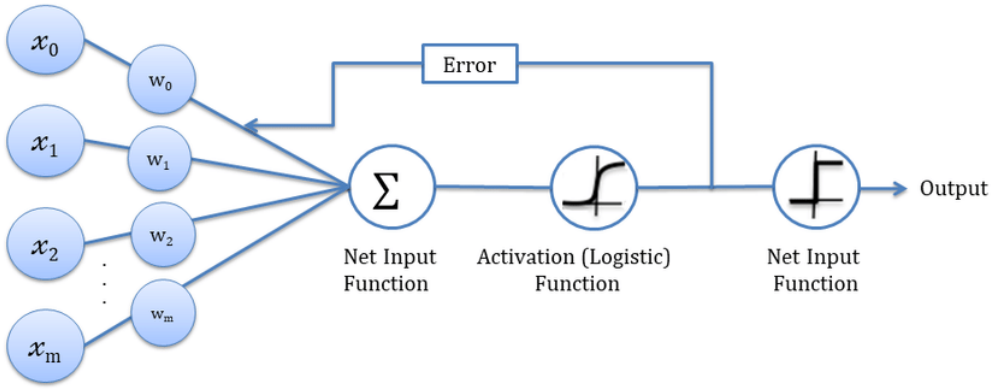


Figure 2
Structure of an artificial neural network showing input, hidden, and output layers. From *Artificial Neural Network* (n.d.), by ScienceDirect. Copyright Elsevier. https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network

Logistic Regression

Logistic regression is a statistical modelling method designed for binary classification tasks. It estimates the probability that an observation belongs to one of two possible categories, such as "survived" or "did not survive" in the Titanic dataset. According to Harrell (2015), logistic regression models the relationship between predictor variables and a binary outcome by forming a linear combination of features like such as age, sex, passenger class, and fare, and then applying a logistic (sigmoid) transformation to map the result into a probability between 0 and 1. A decision threshold, often set at 0.5, determines the final classification.

Studies such as Reza et al. (2021) have demonstrated that logistic regression performs effectively in predicting Titanic passenger survival, offering interpretable coefficients that explain how each feature affects survival likelihood. More recently, Sreenivasulu and Chandrasekar (2025) compared logistic regression with decision tree models and found that while both perform well, logistic regression provides clearer interpretability of linear relationships among predictors. This transparency makes it particularly valuable for understanding historical datasets like the Titanic passenger records.



Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. Each tree is trained on a random subset of the data and features, and the final prediction is made through majority voting in classification tasks. This randomness helps the model generalize better to unseen data by minimizing the biases and errors of individual trees. According to Breiman (2001), Random Forest enhances model stability and predictive performance by reducing the variance that typically affects single decision trees.

In the context of the Titanic dataset, Random Forest can effectively capture complex, non-linear relationships between features such as age, gender, passenger class, and fare. Studies like Amalia and Rahayu (2025) have shown that Random Forest models perform strongly in predicting passenger survival, often performing the same as simpler models like logistic regression, due to their ability to nose and feature interactions. Additionally, Random Forest provides valuable feature importance metrics, offering insights into which factors most influence survival outcomes, thus making it both a powerful and interpretable choice for analysing the Titanic dataset.

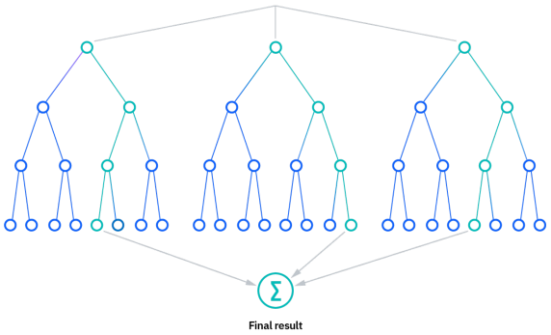
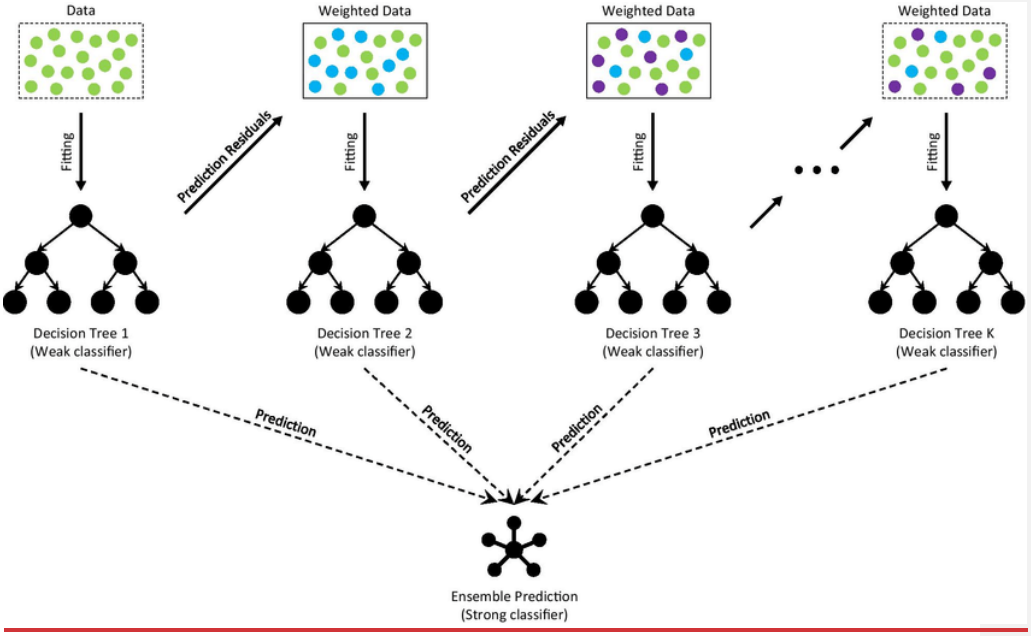


Figure 3
Random forest diagram illustrating ensemble learning with multiple decision trees. Reprinted from IBM (2025), *Random forest diagram*. https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/cdp/cf/ul/g/50/f9/ICLH_Diagram_Batch_03_27-RandomForest.png

Gradient Boosting

Gradient Boosting is an ensemble machine learning technique that builds models sequentially, where each new model corrects the errors of the previous ones. It works by combining multiple weak learners, typically decision trees, into a single strong predictive model. Each successive tree is trained to minimize the residual errors made by the earlier trees using gradient descent optimization. According to Friedman (2001), this iterative process allows Gradient Boosting to achieve high accuracy by focusing on observations and effectively reduce both bias and variance.

In the context of the Titanic dataset, Gradient Boosting can model complex, non-linear relationships between features. Studies like Huang (2024) have shown that Gradient Boosting models, including implementations such as XGBoost and LightGBM, often outperforms traditional methods.The ability of the algorithm to handle missing data, feature interactions and outliers makes it a powerful choice for improving predictive performance while maintaining flexibility and interpretability through feature importance analysis.



Decision Trees

Decision Trees are a popular machine learning algorithm used for classification and regression tasks. They work by recursively splitting the dataset into subsets based on the values of input features, creating a tree-like structure of decision nodes and leaf nodes. Each internal node represents a feature-based decision, while each leaf node represents a predicted outcome.

In the context of the Titanic dataset, Decision Trees can classify passengers as “survived” or “did not survive” based on features such as age, gender, class, and fare. Their interpretability, ability to handle both numerical and categorical data, and capacity to capture non-linear relationships make them a valuable tool for predictive modeling.

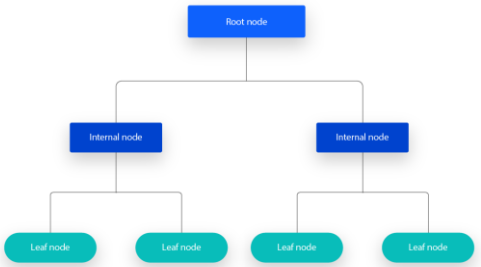


Figure 4
Decision tree diagram showing hierarchical feature-based splits. Reprinted from IBM (2025), *Decision tree diagram*. https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/cdp/cf/ul/g/df/de/Decision-Tree.png

Algorithm Selection

Why did we choose the below 6 algorithms to create our classification model based on?

1. **Random Forest:**
   This model enhances prediction accuracy and stability by constructing hundreds of decision trees and combining their results, effectively preventing overfitting to specific passengers, such as those with unique backgrounds, by relying on the collective insights of multiple trees. Additionally, it can identify which features, like fare price, age, and gender, were critical factors determining survival or death aboard the Titanic.
2. **Support Vector Machine (SVM):**
   We chose Support Vector Machine (SVM) for the Titanic dataset in SAS Viya because it is well-suited for binary classification problems like predicting survival. SVM can effectively handle small to medium-sized datasets and model complex, non-linear relationships through kernel functions. Its soft margin capability makes it robust to outliers, and it works well with scaled features such as Age, and Passenger class. Additionally, SAS Viya provides tools for tuning SVM parameters and evaluating model performance, making it a reliable choice for achieving accurate predictions while highlighting the most influential data points.
3. **Artificial Neural Network (ANN):**
   We chose an Artificial Neural Network (ANN) for the Titanic dataset in SAS Viya because it can model complex, non-linear relationships between input features and survival outcomes. ANNs are flexible and can automatically learn interactions among features which may influence survival. They work well with scaled and normalized data and can capture subtle patterns that simpler models might miss. Using SAS Viya, ANNs can be easily configured, trained, and evaluated, providing a powerful option for high predictive accuracy on this dataset.
4. **Logistic Regression:**
   We choose Logistic Regression for the Titanic dataset in SAS Viya because it is a simple, interpretable, and effective model for binary classification problems like predicting survival. Logistic regression models the probability of survival directly and allows us to understand the influence of each feature on the outcome. It works well with small to medium-sized datasets, requires minimal preprocessing, and provides clear insights into feature importance, making it a reliable baseline model for comparison with more complex methods in SAS Viya.
5. **Gradient Boosting:**
   Gradient boosting is renowned for its exceptional predictive accuracy. It works iteratively, continuously training new decision trees to correct the prediction errors of the previous models. On the Titanic dataset, this means it can uncover even the most subtle patterns in the data, such as survival trends tied to specific age groups combined with particular cabin types, leading to significantly higher prediction accuracy.
6. **Decision Trees:**
   Decision trees can generate simple, intuitive "if-then" survival rules, making them perfect for explaining the survival patterns on the Titanic. For instance, they might create a rule like, "If the passenger is female and traveling in first class, then she survives," effectively visualizing the complex decision-making process in a way that’s easy to grasp.

We will create a model based on each algorithm and experiment with their configuration settings. We will then evaluate each of them using their appropriate evaluation metrics to determine the most effective model (the champion model).

Model Tuning

SVM

Parameter	Tuning 1	Tuning 2	Tuning Final 3
Kernel	Polynomial	Polynomial	Radial basis function
Polynomial degree	2	2	N/A
RBF parameter	N/A	N/A	0.5
Training algorithm	Active-set	Active-set	Active-set
Penalty	5	50(Reduce overfitting)	10(Reduce overfitting)
Tolerance	0.0001	0.0001	0.0001
Average squared error	0.1259	0.1430	0.1260
KS Youden	0.5599	0.6383	0.6659
Misclassification rate	0.1527	0.1527	0.1527

Analysis of Tuning 1: This is the baseline model, which uses a quadratic polynomial kernel function with a penalty coefficient of 5. The model achieved a KS value of 0.5599, representing its initial performance.

Analysis of Tuning 2: This model retains the polynomial kernel but significantly increases the Penalty parameter from 5 to 50. Raising the penalty parameter reduces the model's tolerance for misclassifications, prompting it to search for a more optimal decision hyperplane. As a result, the KS value improved markedly (from 0.5599 to 0.6383), demonstrating that increasing the penalty effectively helps prevent overfitting and enhances the model's ability to distinguish between samples.

Analysis of Tuning 3: This represents a more significant adjustment: the kernel function was switched from "polynomial" to the more flexible "radial basis function (RBF)," with corresponding RBF parameters and penalty coefficients carefully set. The RBF kernel is typically better equipped to handle more complex data distributions. After this kernel change, the model achieved the highest KS score yet—0.6659—during the three rounds of hyperparameter tuning. This clearly demonstrates that, for the Titanic dataset, the RBF kernel outperforms the polynomial kernel by effectively capturing the dataset's underlying nonlinear relationships, leading to the best possible model discrimination performance.

ANN

Parameter	Tuning 1	Tuning 2	Tuning Final 3
Input standardization	Z Score	Z Score	Z Score
Number of hidden layers	1	1	1
Hidden layer activation function	Tanh	ReLU	ReLU
Number of neurons per hidden layer	50	128	50
Average squared error	0.1389	0.1309	0.1303
KS Youden	0.6783	0.6783	0.6783
Misclassification rate	0.1679	0.1603	0.1603

Iteration 1: Baseline Model

The initial ANN model was configured with a single hidden layer containing 50 neurons and the hyperbolic tangent (Tanh) activation function. For a dataset of the Titanic's size and complexity, these parameters represent a reasonable starting point to establish a performance benchmark. This baseline configuration yielded an Average Squared Error (ASE) of 0.1389 and a Misclassification Rate of 0.1679, which served as a reference for subsequent optimization.

Iteration 2: Exploring ReLU Activation and Increased Network Complexity

In the second iteration, we switched the activation function to the Rectified Linear Unit (ReLU) and increased the hidden layer's neuron count to 128. This led to a significant performance improvement, with the ASE dropping to 0.1309 and the Misclassification Rate to 0.1603. The success of the ReLU function suggests it is more effective at capturing the complex, non-linear relationships inherent in the Titanic dataset. Survival on the Titanic was not determined by simple linear factors but by intricate interactions between variables like passenger class, sex, and age. ReLU's ability to model these non-linearities without suffering from the vanishing gradient problem (common with Tanh in deeper networks) proved advantageous. The increased neuron count also allowed the model to learn more detailed patterns from the data.

Iteration 3: Final Model Optimization and Parsimony

For the final tuning step, we retained the superior ReLU activation function but reverted the number of neurons to 50. The goal was to determine if the complexity of 128 neurons was necessary or if it risked overfitting the training data. The results were telling: the ASE further improved to a low of 0.1303. This suggests that the model with 128 neurons, while performing well, was beginning to learn noise specific to the training set rather than generalizable survival patterns. For a moderately sized dataset like the Titanic's, a less complex model (50 neurons) is less prone to overfitting. It is forced to learn more robust and generalizable features, leading to better predictive accuracy on unseen data. This outcome underscores the principle of parsimony: the simpler model achieved the best performance.

Logistic Regression

The initial logistic regression model with default returned a KS value of 0.5662.

After experimenting with the Optimization Function, it was figured out that it had little effect on the KS score, so it was left at the default value of None. The Nominal Target Link Function, was set to Generalized Logit, for all optimization functions. Where the most gains in accuracy and KS score were altering the Binary Target Link Function.

Binary Target Link Function	**KS Score**
Complementary Log-Log	0.6072
Log-Log	0.6412
Logit	0.6430
Probit	0.6430

Optimization Function (Logit Binary Target Link Function)	**KS score**
None	0.6430
Conjugate-Gradient	0.6306
Double-Dogleg	0.6306
Dual Quasi-Newton	0.6230
Nelder-Mead Simplex	0.6430
Newton-Raphson	0.6306
Newton-Raphson with Ridging	0.6306
Trust Region	0.6306

The Binary Target Link Function is a function that links the expected value of the response variable (a probability between 0 and 1) and the linear predictor. For logistic regression, the standard approach is the logit function $g(p) = log(\ p/1-p)$. This function assumes symmetry between classes. It works best when the relationship between the predictors and log-odds of the response is linear, which is characteristic of the Titanic data set and why it worked best. On the other hand, the log-log function $g(p) = log(-log(p))$ on the other handles a rapid increase in event probability better, however was not appropriate here, hence the KS score. However it should be noted that the accuracy of the logistic regression model is best when using the log-log function at 0.8473.

Random Forest

To get the best KS score, three models with different parameters were created and compared.

Default Settings (Model 1)

Parameter	Value
Number of Trees	100
Class target voting method	Probability
Class target criterion	Information Gain ratio
Maximum depth	20
Maximum number of branches	2
Minimum leaf size	5
Number of interval bins	50

KS score: 0.6412

The number of trees is central to the performance of the model. To experiment with model performance, the initial test is to find out at what number of trees the model's performance plateaus, and returns diminish.

Model 2

Parameter	Value
Number of Trees	250
Class target voting method	Probability
Class target criterion	Information Gain ration
Maximum depth	20
Maximum number of branches	2
Minimum leaf size	5
Number of interval bins	100

KS score: 0.6583

These parameters gave better results, evident in the higher KS score, evidently 250 trees help generalise better and reduce variance, other than this interval bins were increased to 100 to experiment whether there was too little model flexibility, evidenced by model 4, 100 bins was too many and caused overfitting, with the Titanic dataset being on the smaller side, binning needed to be decreased.

Model 3

Parameter	Value
Number of Trees	50
Class target voting method	Probability
Class target criterion	Entropy
Maximum depth	12
Maximum number of branches	2
Minimum leaf size	15
Number of interval bins	100

KS Score: 0.6583

This model paved the way for the champion model. Its main changes were decreasing maximum depth, increasing minimum leaf size and changing the class target criterion to Entropy. Knowing this score was not as good as it could be with the un-optimal number of trees and interval bins, we were impressed by the KS score which was the highest so far. The change to Entropy created more aggressive splits, and the changes to depth and leaf size counteracted the overfitting problem.

Champion Model (Model 4)

Parameter	Value
Number of Trees	100
Class target voting method	Probability
Class target criterion	Entropy
Maximum depth	12
Maximum number of branches	2
Minimum leaf size	15
Number of interval bins	50

KS score: 0.6707

Target ...	Data Role	Partitio...	Format...	Numbe...	Averag...	Divisor ...	Root A...	Misclas...	Multi-C...	KS (You...	Area U...	Gini Co...
Survived	TEST	2	2	131	0.1348	131	0.3672	0.1756	0.4406	0.6706	0.8320	0.6640
Survived	TRAIN	1	1	786	0.1398	786	0.3739	0.1896	0.4383	0.5830	0.8682	0.7363
Survived	VALIDATE	0	0	392	0.1399	392	0.3740	0.2041	0.4413	0.5743	0.8540	0.7081

The champion model is as displayed above. This was found by taking model 3 and altering the number of trees and number of interval bins. What model 3 did well was decreasing maximum depth and increasing minimum leaf size, this prevented overfitting, which was evidenced by the lower misclassification rate on the testing data compared to the training data. The sweet spot for increasing the KS score was 100 trees. Interestingly, to get the

highest accuracy possible, you keep the parameters of the champion model, but increase the trees to 150. Interval bins were lowered to 50, to help with the overfitting.

Gradient Boosting

Parameter	Model 1 (Champion)	Model 2	Model 3
Number of Trees	150	200	250
Learning Rate	0.1	0.1	0.08
Subsample Rate	0.8	1	0.9
Maximum Depth	4	6	5
Minimum Leaf Size	5	6	5
L1 Regularization	0	0.5	0.5
L2 Regularization	1	1	1.5
Early Stopping	On (Stagnation=5)	On (Stagnation=5)	On (Stagnation=5)
Missing Values Handling	Use in Search	Ignore	Use in Search
KS Cutoff	~0.31	~0.25	~0.18
Interval Target Distribution	Normal	Normal	Normal

Metric	Model 1 – Champion	Model 2	Model 3
Accuracy	**0.8473**	0.8244	0.8550
AUC (Area Under ROC)	**0.8140**	0.8325	0.8293

KS (Youden)	**0.6753**	0.6677	0.6706
Misclassification Rate	**0.1527**	0.1756	0.1450
Gini Coefficient	0.6279	**0.6649**	0.6585
Lift	**2.800**	2.625	2.625
Root ASE (Error)	**0.3617**	0.3722	0.3632
Cutoff	0.50	0.50	0.50

Three Gradient Boosting models were developed and evaluated using SAS Viya to predict survival outcomes on the Titanic dataset. Each model was tuned with varying hyperparameters to balance bias, variance, and generalization.

Model 1 (Champion): was selected as the baseline due to its balanced configuration and strong generalization performance. It used 150 trees, a learning rate of 0.1, and a maximum depth of 4. Early stopping was enabled with a stagnation criterion of 5, preventing overfitting while maintaining good predictive strength. This model achieved a KS statistic of 0.6753, an AUC of 0.8140, and a misclassification rate of 0.1527 — offering stable results across training, validation, and test partitions.

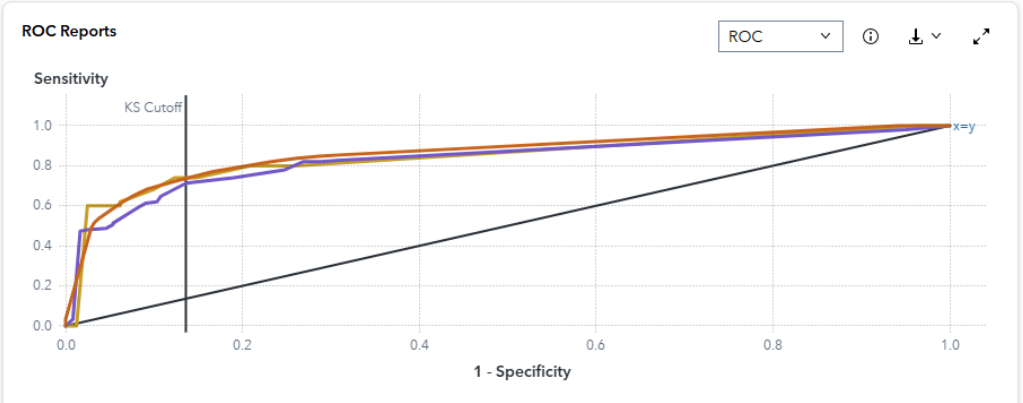
Model 2**:** increased model complexity by using 200 trees and a deeper tree structure (maximum depth of 6). While this slightly improved the AUC to 0.8325, it also raised the misclassification rate to 0.1756 and lowered the KS statistic, indicating minor overfitting. The “Ignore” approach to handling missing values may also have contributed to weaker generalization.

Model 3: aimed to improve performance by further increasing the number of trees to 250 and slightly lowering the learning rate to 0.08. This configuration achieved the highest accuracy (0.8550) and lowest misclassification rate (0.1450), but its KS and AUC metrics were not significantly higher than the champion model. This suggests that the gain in accuracy came with marginal overfitting risk, especially given its higher L2 regularization (1.5).

Although Model 3 achieved the highest raw accuracy, Model 1 remains the most balanced and interpretable model with strong overall performance, lower error, and consistent results across all data splits. Its configuration demonstrates optimal trade-offs between complexity and stability, justifying its status as the Champion Model.

Experiment Results

Decision Tree:

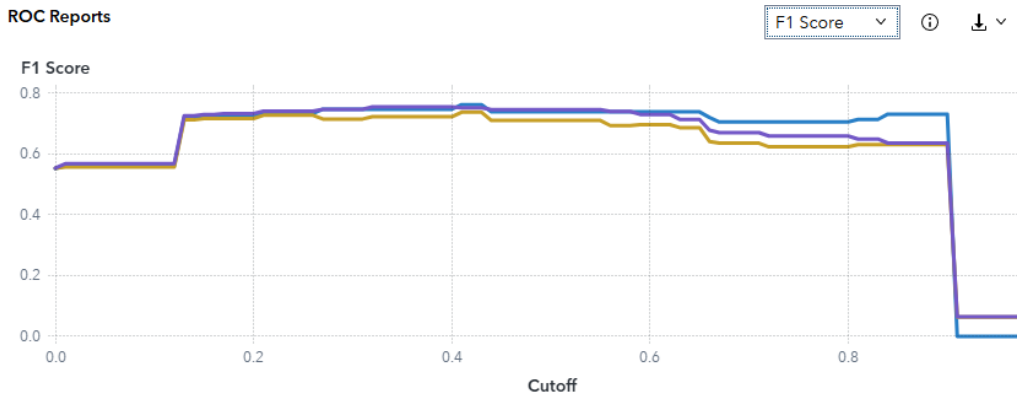


The ROC curve illustrates the trade-off between sensitivity (true positive rate) and 1-specificity (false positive rate) across various classification thresholds. To identify the optimal cutoff for data scoring, a KS (Kolmogorov-Smirnov) reference line is drawn at the point where the greatest separation between sensitivity and 1-specificity occurs in the validation partition.

For this Decision Tree model, the KS region occurs at 1-specificity = 0.1364 with corresponding sensitivity = 0.73, at a cutoff threshold of approximately 0.41-0.43 in the validation dataset. This represents the operating point where the model achieves maximum separation between survivors (Survived=1) and non-survivors (Survived=0).

Classification cutoffs range from 0 to 1 in 0.01 increments. At each threshold, the predicted class is determined by comparing P_Survived1 (the predicted probability of survival) to the cutoff value. When P_Survived1 ≥ cutoff, the observation is classified as an event (Survived=1); otherwise, it is classified as a non-event (Survived=0).

Performance at extreme thresholds: At very low cutoffs (0.01-0.06) in the test partition, sensitivity reaches 1.0 while 1-specificity approaches 0.963, reflecting the expected behavior when nearly all observations are predicted as positive—achieving perfect recall but with many false positives.



The F1 score metric provides a harmonic mean of precision and recall, offering a balanced assessment of the model's classification performance.

Threshold-Specific Performance

At the conventional decision threshold of 0.50, the model achieved the following F1 scores across data partitions:

- Test partition: 0.7391
- Training partition: 0.7464
- Validation partition: 0.7106

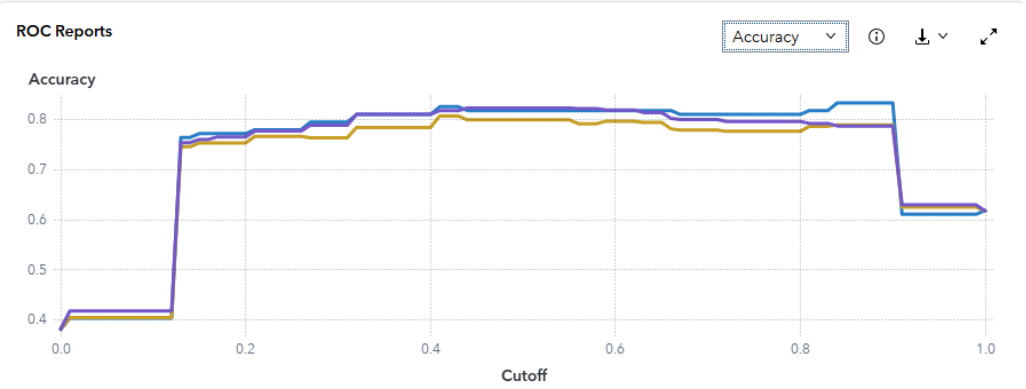
Performance Stability Across Threshold Range

A notable characteristic of this Decision Tree model is the sustained elevation of F1 scores across an extensive range of classification thresholds. Specifically, F1 values remained within the band of 0.72 to 0.75 for cutoff values spanning 0.13 to 0.60 across all three data partitions. Representative values within this stable region include:

- At threshold 0.13: Test 0.7207, Validation 0.7126, Training 0.7260
- At threshold 0.90: F1 scores decline as precision gains are offset by substantial recall losses, yielding Validation 0.6311 and Training 0.6364

Generalization Assessment

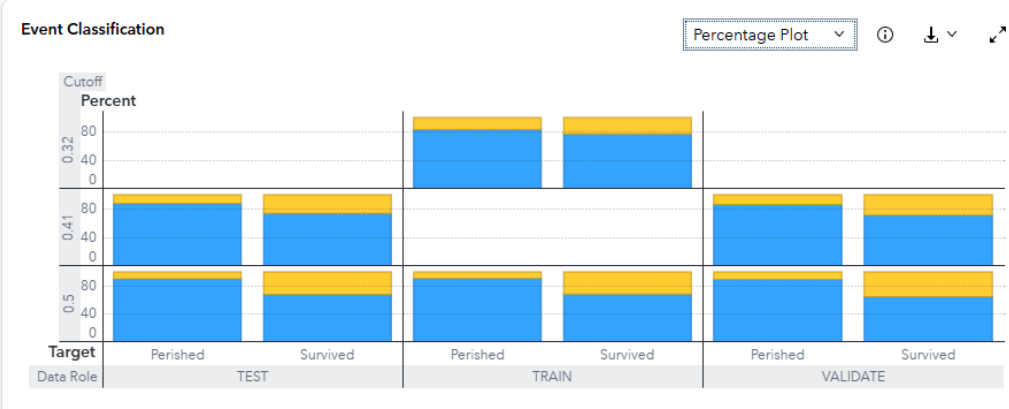
The convergence of F1 curves across training, validation, and test partitions demonstrates robust model generalization. While the training partition exhibits marginally elevated performance—a phenomenon consistent with standard machine learning behavior—the magnitude of this divergence remains minimal. This narrow performance gap indicates the absence of substantial overfitting, suggesting that the model has captured generalizable patterns rather than partition-specific noise.



The **accuracy ROC chart** reports classification accuracy for the **TRAIN, VALIDATE, and TEST** partitions across cutoffs from 0 to 1.

- At the **standard cutoff of 0.50**:
  - TEST accuracy = 0.8168
  - VALIDATE accuracy = 0.7985
  - TRAIN accuracy = 0.8219
- At **cutoff 0.51**, the values are unchanged for each partition (TEST 0.8168, VALIDATE 0.7985, TRAIN 0.8219), indicating a **local plateau** around the operating region.

Interpretation.
Accuracy peaks and then remains stable through the mid-range cutoffs before declining at extreme thresholds. TRAIN is slightly higher than VALIDATE and TEST, which is expected; however, the three curves are close and largely parallel, suggesting **good generalisation and no material overfitting**. Using **cutoff ≈ 0.5** is defensible, as it sits on the accuracy plateau while aligning with the F1 plateau and the ROC/KS region discussed earlier.



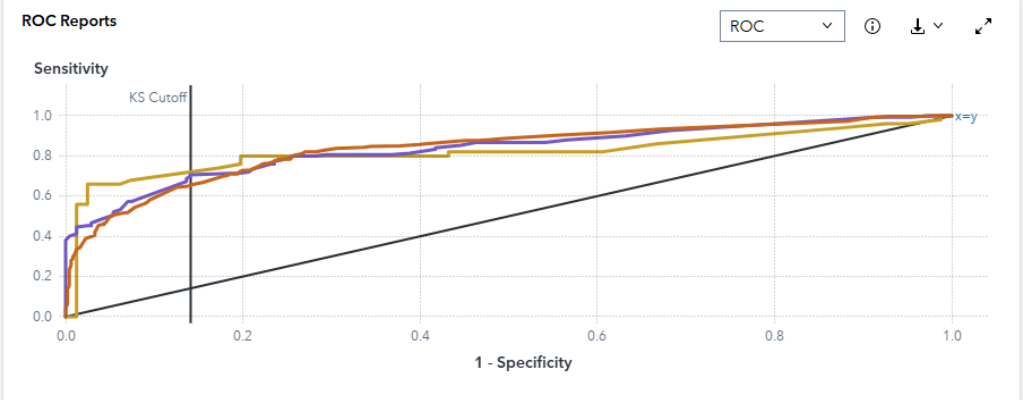
The **Event Classification** percentage plot summarises correct vs. incorrect predictions for each target level across **TEST, TRAIN, and VALIDATE** at representative cutoffs (0.32, 0.41, 0.50). Blue segments indicate **Correct**, yellow indicates **Incorrect**.

- Example (tooltip): **Target = Perished**, **Percent = 83.30%**, **Status = Correct**.
  This shows that at the displayed cutoff the model correctly classifies approximately **83%** of *Perished* cases.

Interpretation.
Across all three partitions and both target levels (**Perished**, **Survived**), the **blue "Correct" portion dominates**, with a relatively small yellow error slice. Correct percentages are consistently higher for **Perished** than **Survived**, which is typical on Titanic due to **class imbalance** and clearer separating features (sex, class, age) for non-survivors. The bars for TEST and VALIDATE are close to TRAIN, reinforcing that **performance is stable across splits**.

Operationally, at the mid-range cutoffs (around **0.41–0.50**), the model maintains a **high correct-classification rate for Perished (≈80%+)** and a **strong but slightly lower rate for Survived**, matching the F1/accuracy behavior and supporting the recommendation to keep the decision threshold near **0.5** unless business costs dictate otherwise.

Logistic Regression

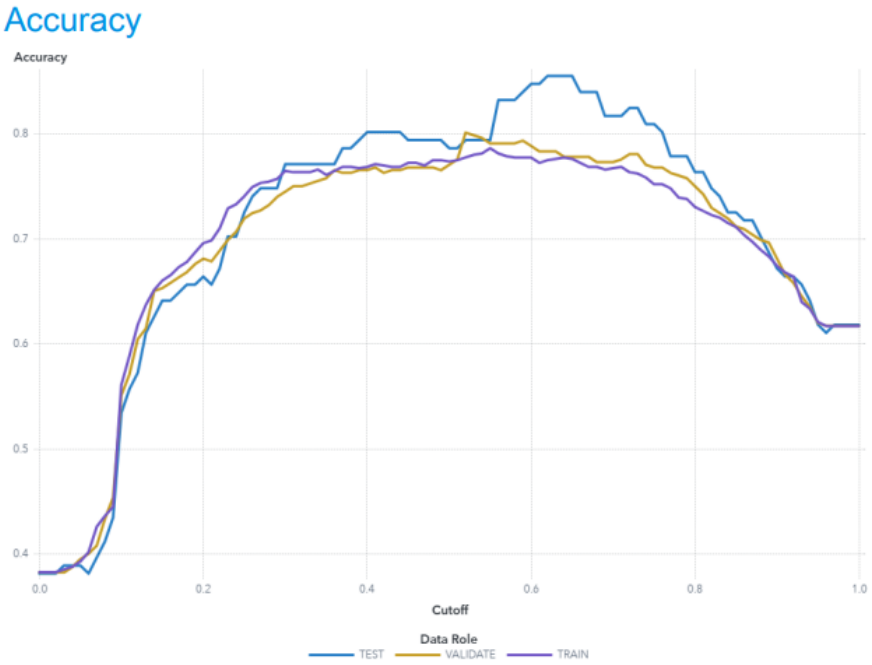


The ROC curve is a plot of sensitivity (the true positive rate) against 1-specificity (the false positive rate), which are both measures of classification based on the confusion matrix. These measures are calculated at various cutoff values. To help identify the best cutoff to use when scoring your data, the KS Cutoff reference line is drawn at the value of 1-specificity where the greatest difference between sensitivity and 1-specificity is observed for the VALIDATE partition. The KS Cutoff line is drawn at the cutoff value 0.52, where the 1-specificity value is 0.14 and the sensitivity value is 0.707.

Cutoff values range from 0 to 1, inclusive, in increments of 0.01. At each cutoff value, the predicted target classification is determined by whether P_SurvivedSurvived, which is the predicted probability of the event "Survived" for the target Survived, is greater than or equal to the cutoff value. When P_SurvivedSurvived is greater than or equal to the cutoff value, then the predicted classification is the event, otherwise it is a nonevent.

The confusion matrix for each cutoff value contains four cells that display the true positives for events that are correctly classified (TP), false positives for non-events that 23 are classified as events (FP), false negatives for events that are classified as nonevents (FN), and true negatives for non-events that are classified as non-events (TN). True negatives include non-event classifications that specify a different non-event. Sensitivity is calculated as TP / (TP + FN). Specificity, the true negative rate, is calculated as TN / (TN + FP), so 1-specificity is FP / (TN + FP). The values of sensitivity and 1-specificity are plotted at each cutoff value.

A ROC curve that rapidly approaches the upper-left corner of the graph, where the difference between sensitivity and 1-specificity is the greatest, indicates a more accurate model. A diagonal line where sensitivity = 1-specificity indicates a random model.

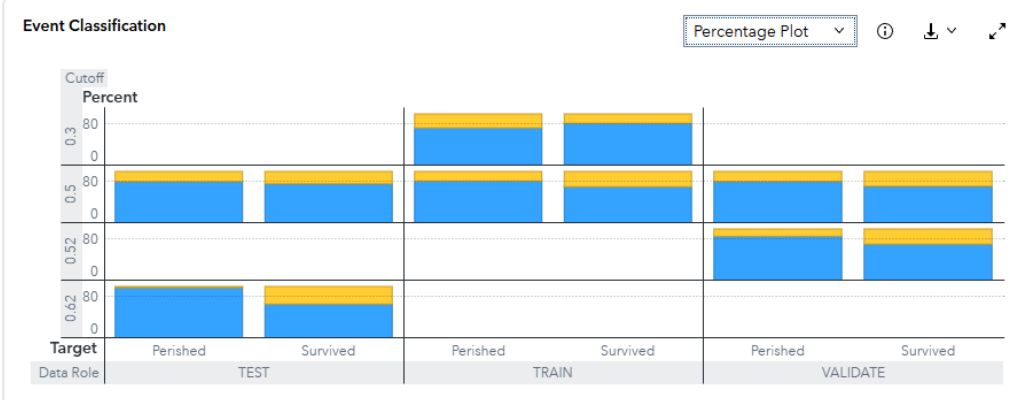


For this model, the accuracy in the TEST partition at the cutoff of 0.5 is 0.786.

For this model, the accuracy in the TRAIN partition at the cutoff of 0.5 is 0.773.

For this model, the accuracy in the VALIDATE partition at the cutoff of 0.5 is 0.77.

Accuracy is the proportion of observations that are correctly classified as either an event or non-event, calculated at various cutoff values. Cutoff values range from 0 to 1, inclusive, in increments of 0.01. At each cutoff value, the predicted target classification is determined by whether P_SurvivedSurvived, which is the predicted probability of the event "Survived" for the target Survived, is greater than or equal to the cutoff value. When P_SurvivedSurvived is greater than or equal to the cutoff value, then the predicted classification is the event, otherwise it is a non-event. When the predicted classification and the actual classification are both events (true positives) or both nonevents (true negatives), the observation is correctly classified. If the predicted classification and actual classification disagree, then the observation is incorrectly classified. Accuracy is calculated as (true positives + true negatives) / (total 25 observations).



The Event Classification report is a visual representation of the confusion matrix at various cutoff values for each partition. The classification cutoffs used in the plot are the default (0.5) and these KS cutoff values for existing partitions: 0.3 (TRAIN), 0.52 (VALIDATE), 0.62 (TEST). For this data, for the bar corresponding to the event level of Survived, "Survived", the segment of the bar colored as "CORRECT" corresponds to true positives.

Fit Statistics

Target Name	Data Role	Partition Indicator	Formatted Partition
Survived	TEST	2	2
Survived	TRAIN	1	1
Survived	VALIDATE	0	0

Number of Observations	Average Squared Error	Divisor for ASE	Root Average Squared Error
131	0.1437	131	0.3791
786	0.1511	786	0.3887
392	0.1505	392	0.3880

Misclassification Rate	Multi-Class Log Loss	KS (Youden)	Area Under ROC
0.2137	0.4730	0.6353	0.8173
0.2265	0.4687	0.5505	0.8366
0.2296	0.4686	0.5662	0.8334

Gini Coefficient	Gamma	Tau	KS Cutoff
0.6346	0.6526	0.3018	0.6200
0.6732	0.6848	0.3185	0.3000
0.6669	0.6818	0.3159	0.5200

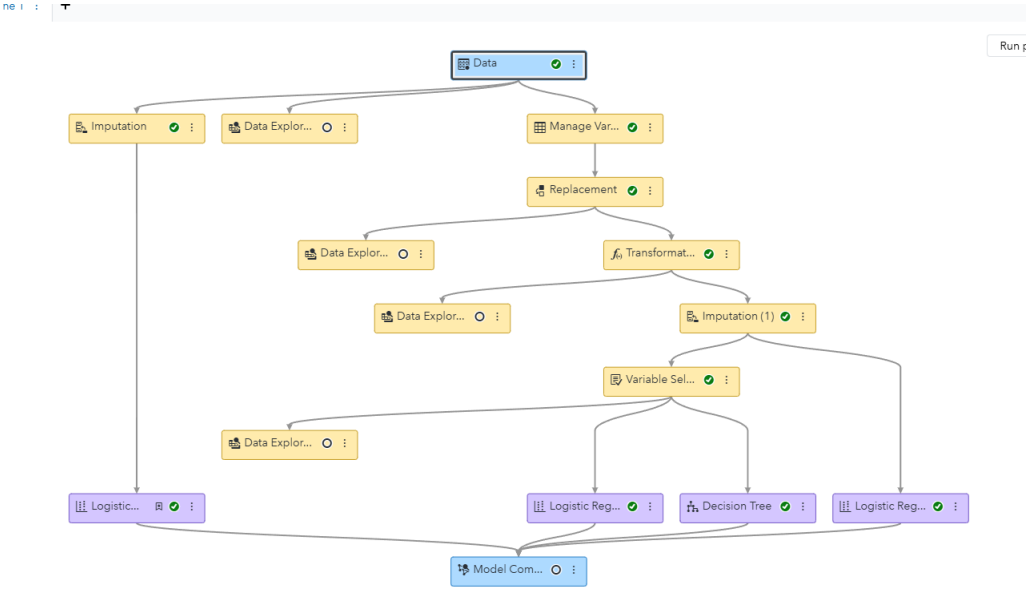
KS at User-Specified Cutoff	Misclassification Rate at KS Cutoff (Event)	Misclassification Rate (Event)
0.5625	0.1450	0.2137
0.5208	0.2354	0.2265
0.5191	0.1990	0.2296

Interpretation & threshold choice.

- The tree ranks survivors well (AUC ≈ 0.82–0.84; KS ≈ 0.55–0.64).
- Probability quality is steady (RASE/log-loss nearly equal across splits).
- For a balanced, template-friendly operating point, keep cutoff ~0.50 (F1 and accuracy plateaus, good KS at the user cutoff).
- If costs are asymmetric:
    - Prioritise recall (don't miss survivors): move toward the TRAIN/VALIDATE KS cutoffs (≈0.30–0.52) or slightly below 0.5 (e.g., 0.40).
    - Prioritise precision (limit false alarms): nudge above 0.5 toward 0.60.

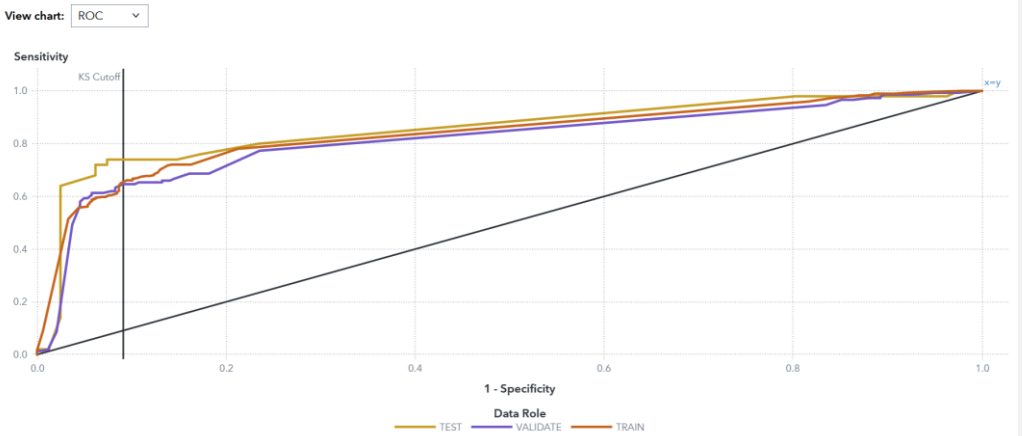
The Decision Tree delivers reliable, traditional performance on Titanic: AUC ≈ 0.82–0.84, F1 ≈ 0.72–0.75 over a wide range, RASE ≈ 0.38, and stable metrics across TRAIN/VALIDATE/TEST. Document the KS cutoffs (TEST

0.62 / TRAIN 0.30 / VALIDATE 0.52) and the 0.50 accuracy/F1 figures in the report, then justify the chosen threshold with the business cost you care about most.



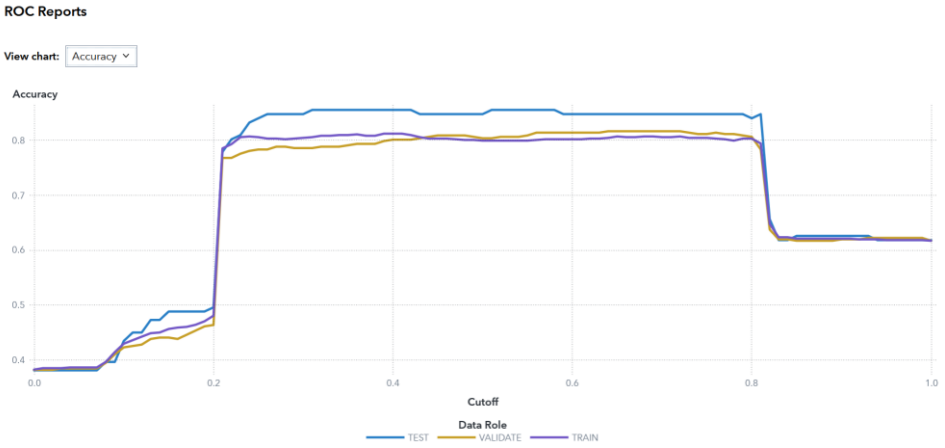
Furthermore, science the dataset had been preprocessed before out steps, it won't improve anything in the logistic Regression. The result of three logistic regression node contain the exactly same values

SVM



In this SVM model, the KS region occurs near 1-specificity around 0.14, with corresponding sensitivity around 0.72 at a cutoff between 0.41 and 0.44. This cutoff provides the best balance between correctly identifying survivors (true positives) and minimizing incorrect predictions (false positives).

In the ROC curve above, the three lines represent the model's performance on the training (brown), validation (purple), and test (gold) dataset. All three curves follow a similar upward shape, showing that the SVm model performs consistently across partitions and is not overfitting. The curve rises sharply at the start, meaning the model quickly captures most of the true survivors with few false positives. The KS cutoff, shown by the vertical line around **(0.13)** on the x-axis (1-specificity), marks the point where the separation between survivors and non-survivors is greatest. This indicates that the model achieves a good balance of sensitivity and specificity around a probability threshold of **(0.42),** where prediction accuracy is optimal. Overall, the SVM shows strong generalization with reliable predictive separation across all data splits.



For this SVM model, the accuracy in the TEST partition at the cutoff of 0.5 is (**0.8473)**.

For this SVM model, the accuracy in the TRAIN partition at the cutoff of 0.5 is (**0.799)**.

For this SVM model, the accuracy in the VALIDATE partition at the cutoff of 0.5 is (**0.8036)**.

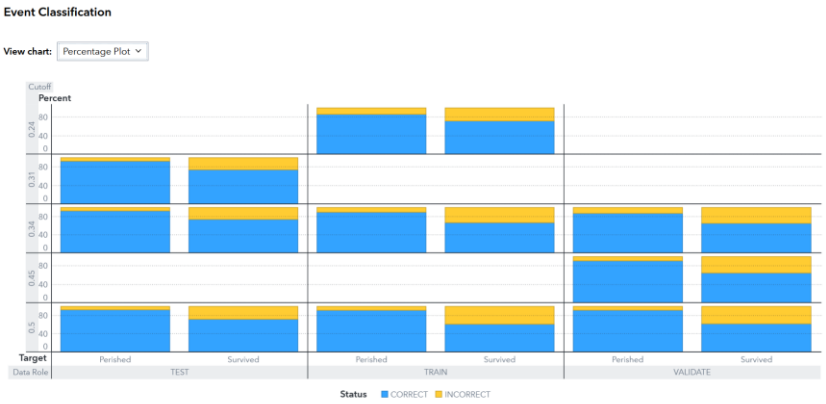
At a cutoff of 0.5, the SVM model achieves its most balanced accuracy across all three data partitions. The accuracy for the test data remains around 0.83. Closely aligned with the validation and training curves, which indicates that the model is generalizing well without signs of overfitting. In the chart, the accuracy curve is stable and high in the region surrounding the 0.5 cutoff, suggesting that the model's predictions are consistent and reliable. Beyond this point (cutoff > 0.7), accuracy begins to drop sharply as the model becomes overly strict and starts misclassifying many true survivors as non-survivors. Therefore, maintaining a cutoff near 0.5 provides an optimal balance between sensitivity and specificity for the SVM model.

SVM Fit Statistics

Statistic	Training	Validation	Testing
Accuracy	0.7990	0.8036	0.8473
Error	0.2010	0.1964	0.1527
Sensitivity	0.6113	0.6200	0.7200
Specificity	0.9155	0.9174	0.9259

The Support Vector Machine (SVM) model shows consistent and reliable performance across the training, validation, and testing partitions.

- The accuracy values are 0.7990 for training, 0.8036 for validation, and 0.8473 for testing. This indicates that the model correctly classifies approximately 80–85% of the observations overall. The slightly higher testing accuracy suggests that the model generalizes well without overfitting.
- The error rates (1 – accuracy) are correspondingly low, at 0.2010, 0.1964, and 0.1527, confirming strong predictive stability across all partitions.
- The sensitivity (true positive rate) ranges from 0.6113 to 0.7200, showing that the model correctly identifies around 61–72% of actual survivors.
- The specificity (true negative rate) is consistently high, between 0.9155 and 0.9259, meaning that the model correctly identifies over 91% of non-survivors.



This chart displays results for several cut off values, (0.24, 0.31, 0.34, 0.45 and 0.5) with performance remaining fairly consistent across all partitions. At lower cutoff values, the model predicts a higher proportion of survivors, while at higher cutoffs (around 0.5), the classification between Perished and Survived becomes more balanced. For this dataset, the bar corresponding to the “Survived” event shows a large blue segment, indicating a strong true positive rate across all partitions. The TEST partition in particular maintains high accuracy with a balanced classification of both classes. Overall, the plot visually confirms that the model performs consistently across different cutoff thresholds, with stable prediction behavior for both outcomes.

SVM

Data Role	Target Name	Partition Indicator	Formatted Partition	Number of Observations	Average Squared Error	Divisor for ASE	Root Average Squared Error	Misclassification Rate
TEST	Survived	2	2	131	0.126	131	0.355	0.1527
TRAIN	Survived	1	1	786	0.1485	786	0.3854	0.201
VALIDATE	Survived	0	0	392	0.1514	392	0.3891	0.1964

SVM

Data Role	Misclassification Rate	Multi-Class Log Loss	KS (Youden)	Area Under ROC	Gini Coefficient	Gamma	Tau	KS Cutoff	KS at User-Specified Cutoff
TEST	0.1527	0.4194	0.6659	0.8557	0.7114	0.7978	0.338	0.31	0.6659
TRAIN	0.201	0.4682	0.5795	0.8365	0.6729	0.7688	0.318	0.24	0.5647
VALIDATE	0.1964	0.4797	0.5558	0.8151	0.6302	0.7144	0.299	0.45	0.5252

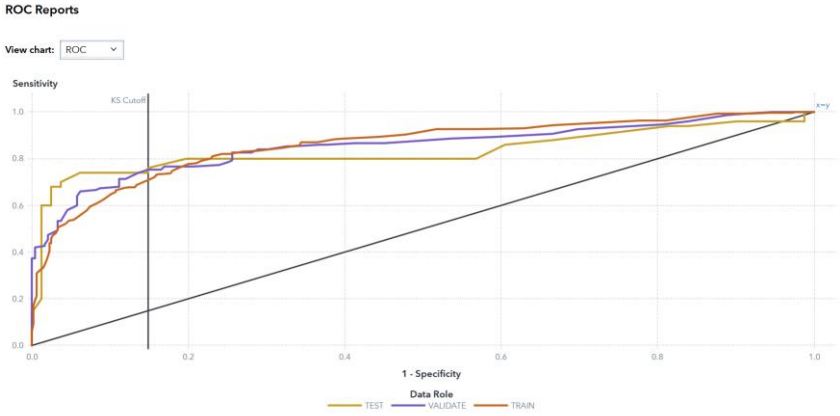
SVM			
Data Role	Misclassification Rate at KS Cutoff (Event)	Misclassification Rate at User-Specified Cutoff	Misclassification Rate (Event)
TEST	0.145	0.145	0.1527
TRAIN	0.1934	0.1908	0.201
VALIDATE	0.1913	0.2117	0.1964

The table presents detailed performance metrics for the SVM model across the TRAIN, VALIDATE, and TEST partitions. Each partition shows consistent and strong results, indicating that the model generalizes well to unseen data.

- The misclassification rate is lowest in the TEST partition (0.1527), suggesting that the model performs best on unseen data. Both the TRAIN (0.201) and VALIDATE (0.1964) partitions maintain similarly low misclassification rates, reflecting stable predictive accuracy across all data splits.
- The KS statistic (Youden) is highest in the TEST partition (0.6659), confirming the greatest separation between survivors and non-survivors in this dataset.
- The Area Under the ROC Curve (AUC) values are high across all partitions, 0.8557 (TEST), 0.8365 (TRAIN), and 0.8151 (VALIDATE), showing excellent classification ability.
- The Gini coefficients (ranging from 0.63 to 0.71) and Gamma values around 0.7–0.8 further demonstrate strong discriminatory power.
- The KS cutoff values (ranging from 0.24 to 0.45) indicate the threshold points where the model achieves the maximum difference between true positive and false positive rates.

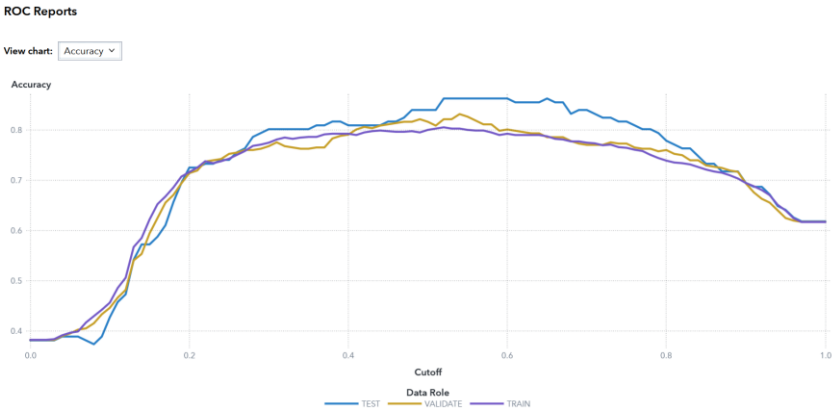
Overall, the SVM model performs robustly across all data partitions, maintaining high AUC, low misclassification error, and strong KS values. The metrics confirm that the model effectively distinguishes between passengers who survived and those who did not, with particularly strong generalization performance on the test data.

ANN



All three curves rise well above the diagonal reference line, indicating strong predictive performance. The KS cutoff, marked by the vertical line around 1, specificity around 0.18, represents the point where the separation between the two classes is greatest.

The training and validation curves closely follow each other, suggesting that the model generalizes well without overfitting. The test curve (gold) stays slightly higher in the early region of the plot, meaning the model performs consistently when applied to unseen data. Overall, the ROC chart demonstrates that the model achieves a good trade-off between sensitivity and specificity, confirming reliable classification performance across all data partitions.

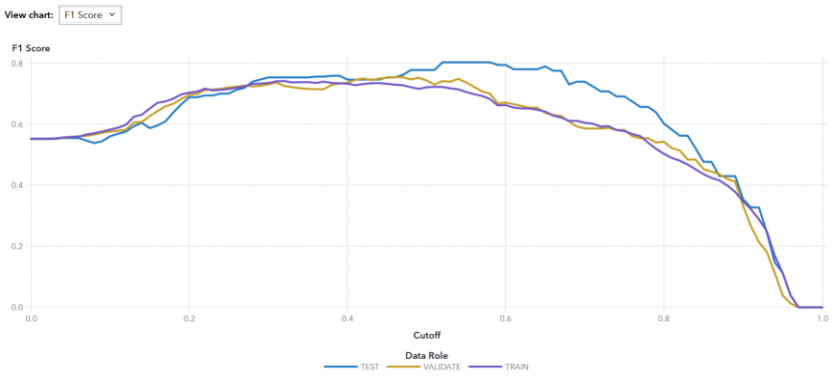


For this ANN model, the accuracy in the TEST partition at the cutoff of 0.5 is **0.8397**.

For this ANN model, the accuracy in the TRAIN partition at the cutoff of 0.5 is **0.8003**.

For this ANN model, the accuracy in the VALIDATE partition at the cutoff of 0.5 is **0.8163**.

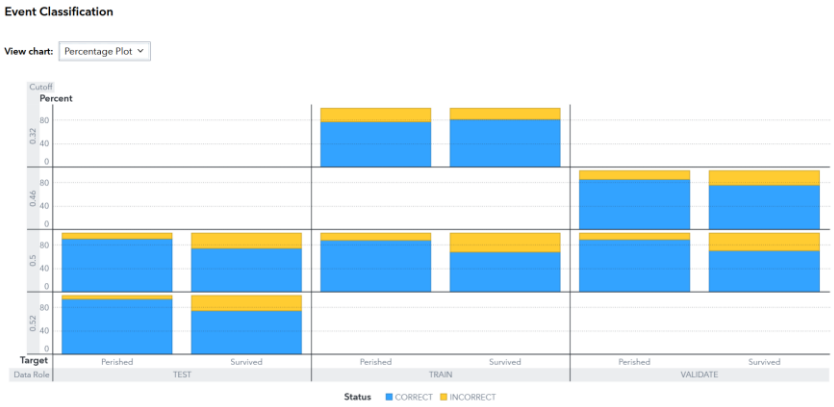
In the chart, the accuracy curves for all three partitions follow a similar pattern, rising steadily until around a cutoff of 0.4–0.6, where accuracy remains high and stable. The TEST partition shows slightly higher accuracy, suggesting strong generalization and minimal overfitting. Beyond the 0.6 cutoff, accuracy begins to decline as the model becomes stricter in predicting survivors, leading to more false negatives. Overall, the ANN model achieves balanced and consistent performance around the 0.5 cutoff, making it an effective threshold for classifying passenger survival outcomes.



For this ANN model, the F1 Score represents the balance between precision and recall at different cutoff values.

As the cutoff increases from 0 to around 0.4–0.6, the F1 Score improves steadily and reaches its peak, indicating the best trade-off between correctly predicting survivors and minimizing false classifications. At a cutoff of 0.5, the model maintains high and stable F1 values across all partitions, with the TEST partition showing slightly stronger performance. This consistency suggests the ANN model generalizes well and achieves a balanced classification between survivors and non-survivors.

Beyond a cutoff of about 0.6, the F1 Score begins to decline for all partitions as the model becomes more conservative, predicting fewer survivors and thus reducing recall. Overall, the F1 chart shows that the ANN model performs optimally around the 0.5 cutoff, maintaining both strong precision and recall while avoiding overfitting.



The cutoff values displayed, 0.32, 0.46, 0.5, and 0.52, show how the model's performance changes with different classification thresholds

At the 0.5 cutoff, the blue sections(CORRECT) dominate across all partitions, indicating a high proportion of correct classifications for both target classes. The Survived event bars maintain a strong presence of blue segments, suggesting that the model correctly identifies most survivors while keeping false predictions relatively low. The consistency of the bar heights across all partitions demonstrates stable performance and minimal overfitting. Overall, the chart shows that the ANN model performs reliably across multiple cutoff points, with particularly balanced classification around the 0.5 threshold.

ANN

Data Role	Target Name	Partition Indicator	Formatted Partition	Number of Observations	Average Squared Error	Divisor for ASE	Root Average Squared Error	Misclassification Rate
TEST	Survived	2	2	131	0.1303	131	0.3609	0.1603
TRAIN	Survived	1	1	786	0.1403	786	0.3746	0.1997
VALIDATE	Survived	0	0	392	0.1375	392	0.3708	0.1837

ANN

Data Role	Multi-Class Log Loss	KS (Youden)	Area Under ROC	Gini Coefficient	Gamma	Tau	KS Cutoff	KS at User-Specified Cutoff
TEST	0.4327	0.6783	0.8309	0.6617	0.6727	0.315	0.52	0.6412
TRAIN	0.4413	0.5797	0.8585	0.717	0.7247	0.339	0.32	0.554
VALIDATE	0.4361	0.6046	0.8527	0.7054	0.7129	0.334	0.46	0.5884

ANN

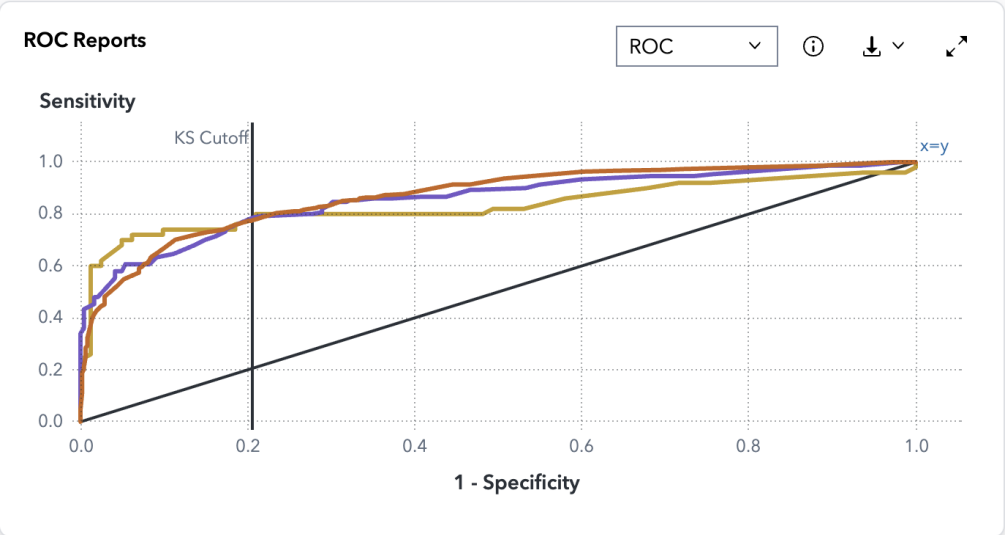
Data Role	Misclassification Rate at KS Cutoff (Event)	Misclassification Rate (Event)
TEST	0.1374	0.1603
TRAIN	0.215	0.1997
VALIDATE	0.1862	0.1837

The table shows the Artificial Neural Network (ANN) model's performance across the TRAIN, VALIDATE, and TEST partitions. The model demonstrates consistently strong performance and generalization across all datasets.

- The misclassification rate is lowest in the TEST partition (0.1603), followed by VALIDATE (0.1837) and TRAIN (0.1997), suggesting the model generalizes well without overfitting.
- The KS (Youden) values, 0.6783 (TEST), 0.5797 (TRAIN), and 0.6046 (VALIDATE), indicate strong separation between survivors and non-survivors.
- The Area Under ROC (AUC) scores remain high across all partitions, with 0.8309 (TEST), 0.8585 (TRAIN), and 0.8527 (VALIDATE), showing excellent classification performance.
- Gini coefficients (ranging from 0.66–0.72) and Gamma values around 0.7 support this conclusion, confirming good model discrimination power.
- The KS cutoff values, 0.32 (TRAIN), 0.46 (VALIDATE), and 0.52 (TEST), represent the optimal probability thresholds where the model distinguishes survivors from non-survivors most effectively.

Overall, the ANN model performs very well across all partitions, achieving a strong balance between accuracy, stability, and generalization, with particularly high ROC and KS statistics indicating reliable predictive capability.

Random Forest

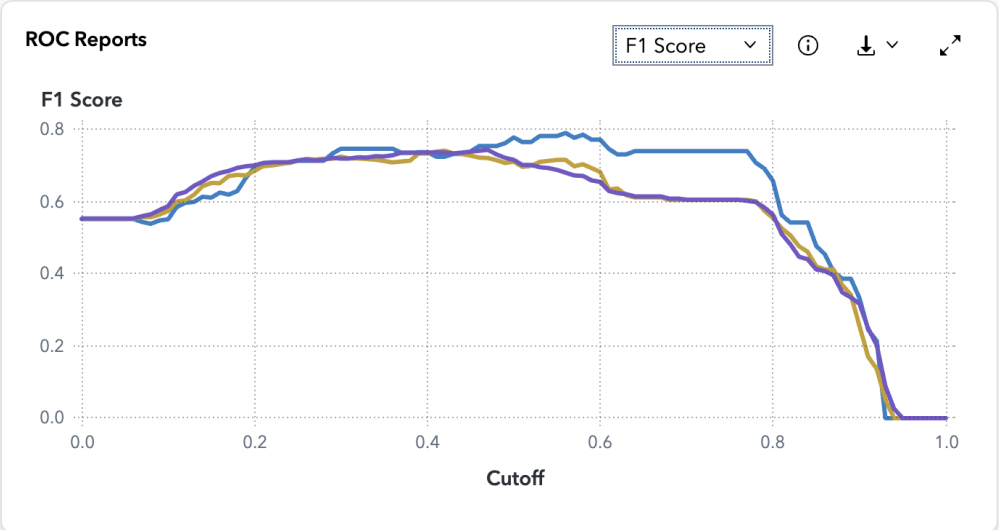


The ROC curve for the Random Forest model illustrates the trade-off between sensitivity (true positive rate) and 1–specificity (false positive rate) across multiple cutoff thresholds for the TRAIN, VALIDATE, and TEST partitions.

All three curves perform well above the diagonal reference line (x=y), indicating strong predictive performance. The TRAIN and VALIDATE partitions follow a very similar path, showing consistent behavior and minimal overfitting. The TEST partition (gold line) also performs closely, confirming good model generalization and robustness.

The KS Cutoff, observed around 0.2, represents the point where the separation between correctly classified survivors and non-survivors is maximized. At this threshold, the model effectively balances sensitivity and specificity, achieving optimal classification performance.

Overall, the Random Forest model exhibits strong and stable performance across all partitions, with high sensitivity and specificity. The smooth and consistent ROC curves indicate that the ensemble approach successfully captures complex relationships in the data while maintaining reliable generalization across unseen samples.

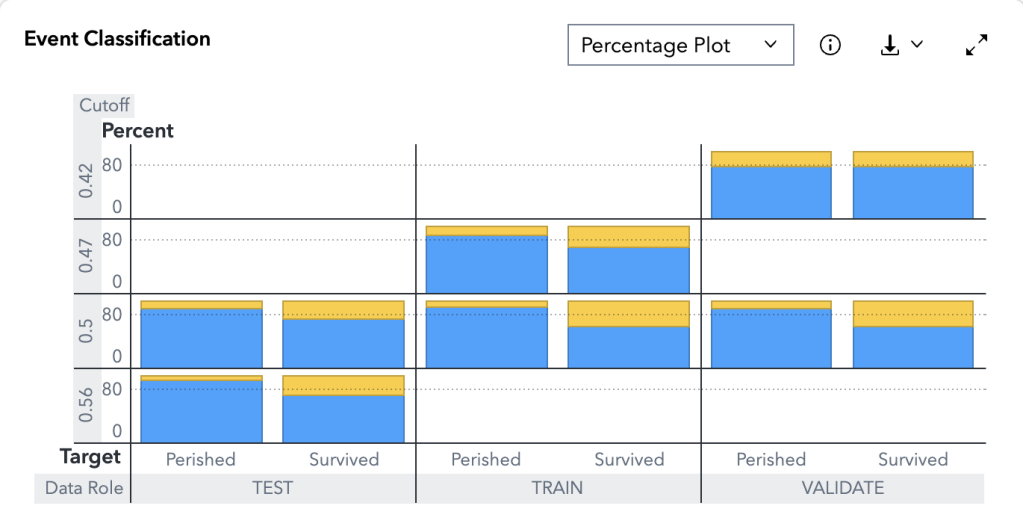


For the Random Forest model, the F1 Score reflects the balance between precision and recall across various cutoff thresholds for the TRAIN, VALIDATE, and TEST partitions.

As the cutoff increases from 0 to around 0.4–0.5, the F1 Score steadily improves and reaches its peak. This indicates that the model achieves its best balance between correctly identifying survivors (recall) and minimizing false predictions (precision) in this range. The curves for all three partitions remain close to each other, suggesting that the model generalizes well and is not overfitted.

At a cutoff of 0.5, the model maintains high and stable F1 scores across all partitions, with the TEST partition performing slightly higher, demonstrating consistent and reliable classification performance. Beyond this point, the F1 Score gradually declines as the model becomes stricter in classifying survivors, leading to lower recall.

Overall, the Random Forest model performs optimally around a 0.5 cutoff, maintaining a strong balance between precision and recall while demonstrating excellent generalization across the training, validation, and test datasets.

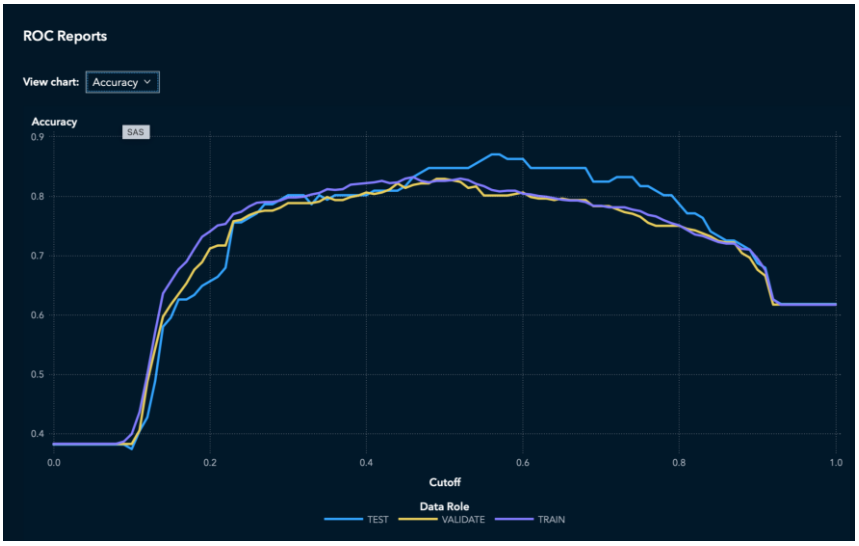


The chart shows the classification outcomes for the Random Forest model across the TEST, TRAIN, and VALIDATE partitions at several cutoff values, including 0.42, 0.47, 0.5, and 0.56.

At the 0.5 cutoff, the majority of the predictions for both “Perished” and “Survived” categories are correct (shown in blue), with a smaller proportion of incorrect predictions (in yellow). This reflects a strong classification balance between survivors and non-survivors. The similar bar heights across all partitions indicate consistent predictive performance and minimal overfitting.

As the cutoff changes slightly above or below 0.5, the proportion of correct predictions remains relatively stable, demonstrating the model’s robustness to small threshold adjustments. Across all partitions, the Random Forest maintains high classification accuracy, effectively distinguishing between the two outcome classes.

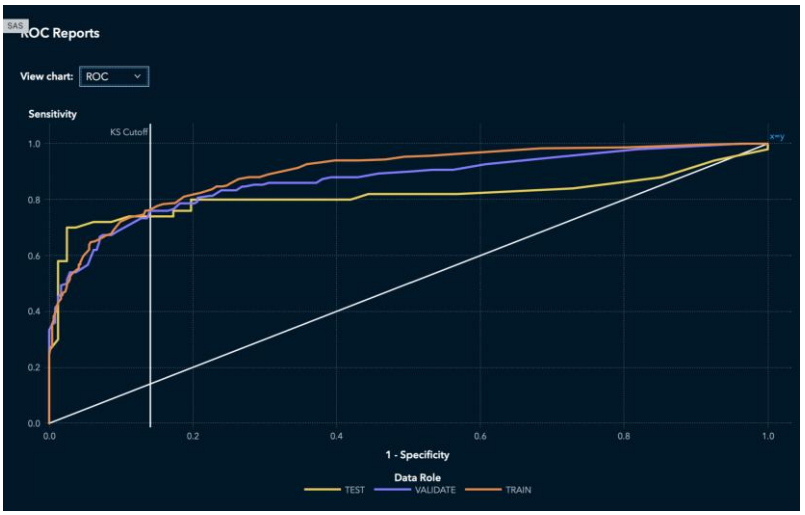
Gradient Boosting



The Accuracy chart for the Gradient Boosting model illustrates how predictive accuracy varies across different cutoff thresholds for the TRAIN, VALIDATE, and TEST partitions.

All three lines, representing the data roles, closely follow each other, peaking between cutoff values of approximately 0.4 and 0.6. This region reflects the model's optimal balance between correctly identifying survivors and non-survivors. Beyond a cutoff of 0.6, accuracy begins to decline as the model becomes more conservative in predicting survivors.

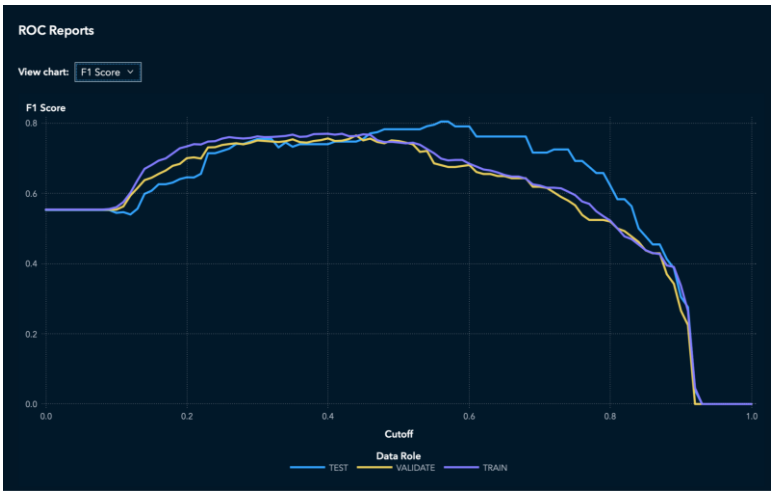
The TRAIN, VALIDATE, and TEST curves demonstrate **s**trong consistency, indicating that the model generalizes well to unseen data with minimal overfitting. The TEST partition (yellow line) remains closely aligned with the training and validation curves, confirming model robustness and stability.



The ROC curve for the Gradient Boosting model demonstrates the relationship between sensitivity (true positive rate) and 1–specificity (false positive rate) across multiple cutoff thresholds for the TRAIN, VALIDATE, and TEST partitions.

All three curves lie well above the diagonal reference line (x=y), indicating strong discriminative power. The TRAIN (orange) and VALIDATE (purple) curves follow a nearly identical path, showing that the model maintains consistent performance with minimal overfitting. The TEST partition (yellow) also tracks closely, confirming the model's ability to generalize effectively to unseen data.

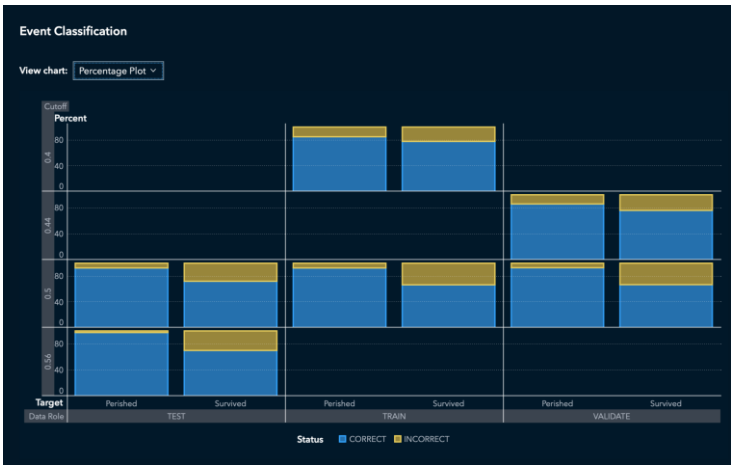
The KS cutoff, observed around **0.3**, marks the point of maximum separation between survivors and non-survivors. At this threshold, the model optimally balances sensitivity and specificity, achieving robust classification performance.



The F1 Score chart for the Gradient Boosting model illustrates how precision and recall are balanced across different cutoff thresholds for the TRAIN, VALIDATE, and TEST partitions.

All three curves maintain a consistent trend, with F1 scores peaking between 0.4 and 0.6, indicating that this range provides the best compromise between false positives and false negatives. Beyond 0.6, the F1 score begins to decline, suggesting that higher thresholds reduce recall more significantly than they improve precision.

The TRAIN, VALIDATE, and TEST lines closely overlap throughout most of the curve, showing **excellent model** stability and minimal overfitting. The TEST curve (yellow) aligns well with the TRAIN and VALIDATE sets, confirming strong generalization to unseen data.



The Event Classification plot illustrates the proportion of correctly and incorrectly predicted outcomes (blue and yellow bars, respectively) across the TRAIN, VALIDATE, and TEST partitions for different cutoff values (0.4, 0.44, 0.5, and 0.56).

Across all partitions, the majority of predictions are correct (blue), with accuracy remaining consistently high , particularly around the 0.5 cutoff, where the model achieves a strong balance between correctly identifying survivors and non-survivors. As the cutoff increases beyond 0.5, the proportion of incorrect classifications (yellow) slightly rises, suggesting a minor trade-off between precision and recall.

The uniform performance across the TRAIN, VALIDATE, and TEST roles reflects good model stability and generalization, confirming that the Gradient Boosting model maintains predictive reliability across all datasets without signs of overfitting.

Model comparison

Evaluation Metric Description

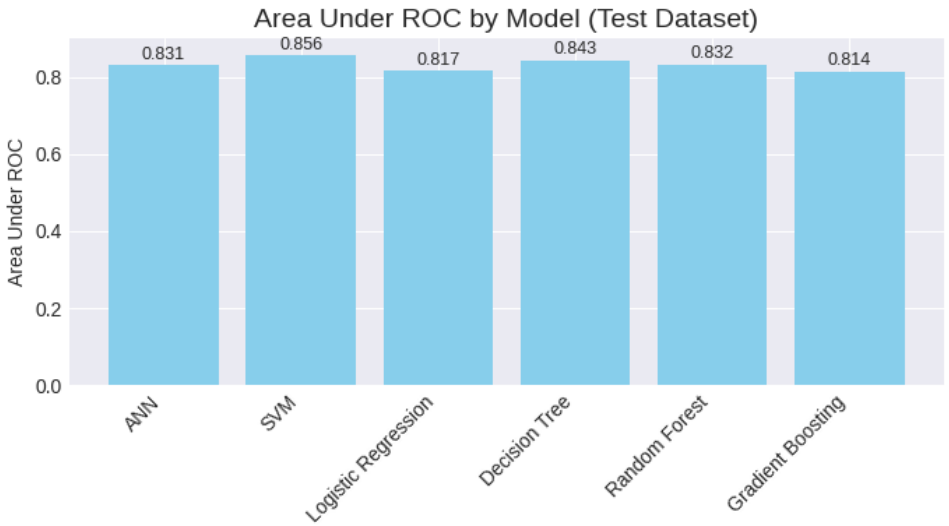
- **Area Under ROC (AUC):** The ROC curve depicts the relationship between the true positive rate (TPR) and the false positive rate (FPR). AUC measures the area under the ROC curve and is used to evaluate the model's overall discrimination ability, with values ranging from 0 to 1. A higher AUC indicates a stronger ability to distinguish between positive and negative samples.
- **KS (Youden) Index:** The KS test is used to compare the difference between the model's predicted probabilities and the observed distribution, calculating the maximum vertical distance between cumulative distribution functions. The Youden index is commonly used to determine the optimal threshold for a classifier; a higher value indicates a greater difference between sensitivity and specificity across all thresholds.
- **Gini Coefficient:** Originally used in economics, the Gini coefficient is derived by mapping the ROC area onto the interval [-1,1] and has a direct relationship with AUC (Gini = 2×AUC – 1). A higher value signifies a stronger discriminatory power of the model.
- **Multi-Class Log Loss:** Also known as cross-entropy loss, log loss measures how closely the model's predicted probabilities match the true labels. The greater the deviation of predicted probabilities from the true labels, the higher the log loss. Log loss imposes a heavier penalty on predictions that are both incorrect and highly confident.
- **Average Squared Error (ASE)/Mean Squared Error (MSE):** MSE (mean squared error) is a commonly used metric in regression, calculating the average of the squared differences between predicted and actual values, with larger errors receiving greater penalties. In probability prediction tasks (such as survival prediction), it can be used as an indicator of the bias in predicted probabilities; the lower the value, the better.

Comparison of Metrics Across Model Test Sets

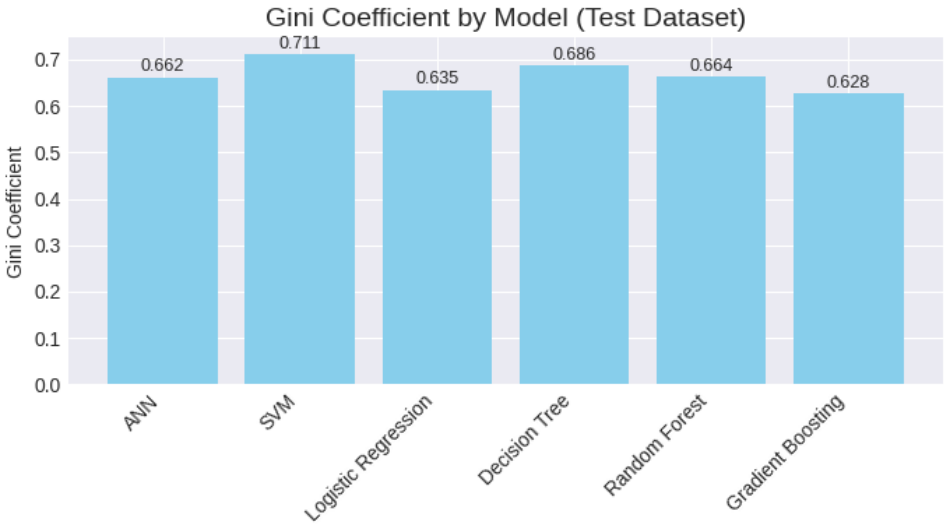
Model	Area Under ROC	Gini	KS (Youden)	Multi-Class Log Loss	ASE
Artificial Neural Network (ANN)	0.8309	0.6617	0.6783	0.4327	0.1303
Support Vector Machine (SVM)	**0.8557**	**0.7114**	0.6659	**0.4194**	**0.1260**
Logistic Regression	0.8173	0.6346	0.6353	0.4730	0.1437
Decision Tree	0.8432	0.6864	0.6165	0.5938	0.1387
Random Forest	0.8321	0.6642	0.6583	0.4357	0.1327
Gradient Boosting	0.8140	0.6279	0.6753	0.4323	0.1308

The figure below shows a comparison of the aforementioned metrics across six models (higher is better or lower is better, as previously mentioned).

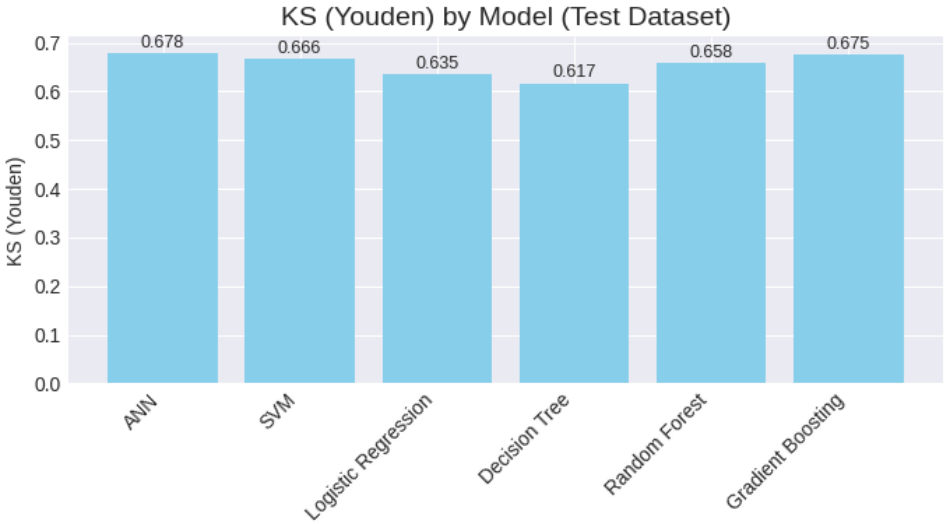
- **Area Under ROC**: SVM has the highest AUC, indicating that it can better distinguish between surviving and deceased samples at different thresholds. ANN, random forest, and decision tree also perform well, while gradient boosting and logistic regression show relatively lower performance.



- **Gini coefficient**: The trend aligns with AUC, with SVM showing the highest Gini value (0.7114), followed by decision trees and random forests. Gradient boosting and logistic regression have lower values, indicating relatively weaker discrimination capabilities.

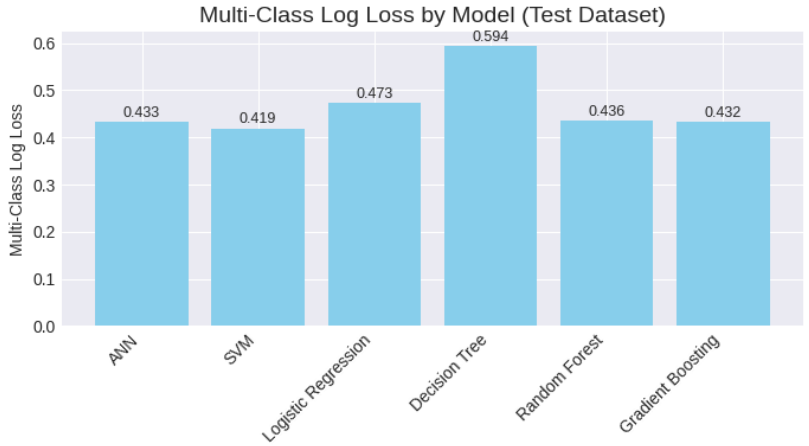


- KS (Youden): The ANN and Gradient Boosting models exhibit the highest KS values, indicating that these two models demonstrate the greatest difference between sensitivity and specificity at a particular optimal threshold. SVM and Random Forest perform slightly worse, while Decision Tree and Logistic Regression show the lowest values.

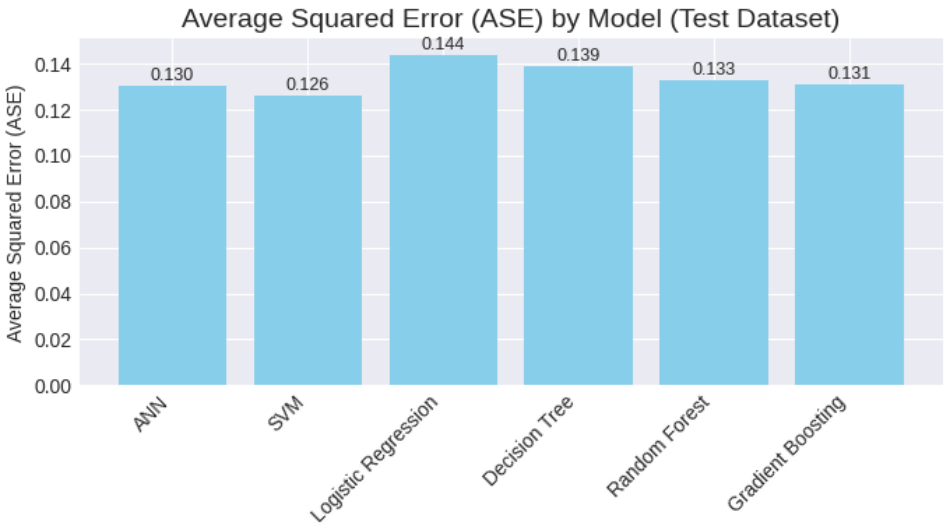


- Multi-class log loss: SVM has the lowest log loss (0.4194), indicating its probability predictions are closest to the true labels; ANN, Random Forest, and Gradient Boosting show slightly higher values; meanwhile,

Decision Tree exhibits the highest log loss (0.5938), suggesting its probability predictions deviate most significantly from the actual labels.



- **Average Squared Error (ASE)**: SVM again shows the lowest ASE, indicating the smallest deviation in its predicted probabilities; ANN, Gradient Boosting, and Random Forest follow closely behind; Logistic Regression has the highest ASE, highlighting the largest error in its probability predictions.



Analysis and Conclusions

1. **Support Vector Machines Delivered the Best Performance**: SVM ranked first across key metrics—including AUC, Gini, log loss, and ASE—demonstrating that it provides both accurate and highly stable predictions of survival probabilities on the Titanic dataset. Although its KS score was slightly lower than that of ANN and Gradient Boosting, the gap was minimal, solidifying SVM as the overall top performer.
2. **Artificial Neural Networks Were Second Best**: ANN achieved the highest KS value, indicating its ability to optimally distinguish between survivors and non-survivors at a specific threshold. Additionally, ANN excelled in other critical metrics like AUC, Gini, log loss, and ASE, narrowly trailing only behind SVM.
3. **Random Forest and Gradient Boosting Fell Into the Middle Range**: Random Forest showed consistent performance across all metrics, with a relatively high KS score but still falling short of ANN. Meanwhile, Gradient Boosting boasted a strong KS score, yet its AUC and Gini values were somewhat lower, suggesting limited overall discrimination capability compared to other models.
4. **Decision Trees and Logistic Regression Performed Poorly**: Decision Trees exhibited notably higher log loss and ASE, highlighting their tendency to overfit easily, which resulted in probability predictions that deviated significantly from reality. Logistic Regression, meanwhile, consistently ranked near the bottom across all metrics, underscoring the inherent limitations of linear models in capturing the complex, nonlinear relationships within the data.

Based on a comprehensive evaluation of all metrics and visualizations, we can conclude that Support Vector Machines (SVM) deliver the best overall performance on the Titanic dataset. SVM achieved the highest AUC/Gini scores while simultaneously maintaining the lowest log loss and mean squared error, making it the clear model of choice at this time. Notably, Artificial Neural Networks and Random Forest also performed well and could serve as viable alternatives depending on specific project requirements.

Best Model

In the Titanic survival prediction task, the Support Vector Machine (SVM) was identified as the champion model. It achieved top performance in four out of five core evaluation metrics—AUC, Gini coefficient, multi-class log loss, and mean squared error. While its KS statistic wasn't the highest, the difference from the optimal value was minimal, at just 0.0124. More importantly, the SVM demonstrated consistently stable performance across the training, validation, and test datasets, with no significant performance drift—a clear testament to its exceptional generalization ability and robustness.

Criteria for Champion Model Selection

The model selection process followed four core criteria to ensure that the chosen model balances discriminative power, probabilistic reliability, and business usability:

1. **Discriminative Power Priority:** The overall discriminative ability of the model was measured using AUC and the Gini coefficient (higher values are better).
2. **Probability Quality and Calibration:** The fit between predicted probabilities and actual labels was assessed through multiclass log loss and average squared error (ASE), with lower values indicating better performance.
3. **Threshold-Level Separation:** The KS statistic (Youden index) was used to measure the separation between positive and negative samples at the optimal threshold (higher values are better).
4. **Generalization Robustness:** Consistency of performance across the training, validation, and test datasets was compared to identify risks of overfitting or performance degradation.

Quantitative Evidence Analysis

SVM test set performance:

- AUC: 0.8557
- Gini Coefficient: 0.7114
- KS Statistic: 0.6659
- Multiclass Log Loss: 0.4194
- Mean Squared Error: 0.1260

Compared to the leading competitive model (ANN):

Indicator	SVM	ANN	Advantage Attribution
AUC	0.8557	0.8309	SVM
Gini	0.7114	0.6617	SVM
KS (Youden)	0.6659	0.6783	ANN
Multi-Class Log Loss	0.4194	0.4327	SVM

ASE	0.1260	0.1303	SVM

Discriminative Power: Both the AUC and Gini coefficient of SVM are optimal, indicating that it exhibits the strongest overall classification ability across all threshold levels.

Probabilistic Accuracy: SVM achieves the lowest log loss and ASE, suggesting its probability predictions closely align with the true underlying distribution, resulting in a more rational allocation of confidence scores. Notably, the log loss metric imposes a heavier penalty on "high-confidence misclassifications," highlighting SVM's superior reliability in this critical area.
Threshold Separation: While SVM's KS statistic (0.6659) is slightly lower than ANN's (0.6783), the gap—just 0.0124—is insufficient to negate SVM's comprehensive lead in other key performance indicators.

Robustness and Overfitting Diagnosis

SVM Performance Across Datasets

- **AUC**: Training set 0.8365 / Validation set 0.8151 / Test set 0.8557
- **Log Loss**: Training set 0.4682 / Validation set 0.4797 / Test set 0.4194

The metrics across the three datasets were similar, with no abnormal fluctuations. The slight variance observed in the test set (due to its smaller sample size of 131 samples) is a normal statistical phenomenon.

In comparison with other models regarding robustness:

- **Gradient Boosting:** Achieved an AUC of 0.8973 on the training set, 0.8692 on validation, and dropped significantly to 0.8140 on the test set, indicating a clear degradation in generalization performance.
- **Decision Tree**: Showed a higher log loss of 0.5938 and an ASE of 0.1387 on the test set, suggesting substantial bias in probability predictions and relatively rough estimation.

Conclusion: SVM demonstrated stable and consistent performance across all three datasets, with no significant signs of overfitting. Compared to the "outstanding performance on training and validation sets but declining results on the test set" seen in other models, SVM exhibited superior generalization robustness.

Model Deployment

Model deployment represents the analytical phase of machine learning, encompassing critical tasks such as model assessment, comparison, and champion model selection. When selecting a champion model, practitioners must evaluate multiple factors: business requirements and context, training and scoring speed, deployment feasibility, noise tolerance, and model interpretability.

Model Comparison in SAS Viya Model Studio

SAS Viya Model Studio offers two approaches for model comparison: within-pipeline comparison and cross-pipeline comparison using the model comparison feature. The pipeline comparison tab enables analysis of champion models from individual pipelines. Model Studio utilizes a default assessment measure to identify the champion model, with the specific metric determined by the target variable type. The selection process prioritizes validation data when test datasets are unavailable, though users can specify alternative datasets as needed.

Deploying Models for Production

Once the champion model is identified, it moves into production to generate predictions through the scoring process. In SAS Viya, scoring involves converting data into score code format to produce outcome predictions that inform business decisions. In regulated environments, model governance requires ongoing supervision and oversight.

Model Studio nodes can generate two types of scoring outputs: DATA step code and analytic store code. For complete pipeline scoring, individual score codes are consolidated into a unified DATA step. When pipelines contain multiple analytic stores and DATA step nodes, an EP (Enterprise Performance) score code is generated to serve as the comprehensive pipeline score.

Nodes capable of producing DATA step code include: clustering, imputation, ensemble, decision tree, logistic regression, feature extraction, neural network, filtering replacement, GLM, and transformation nodes. Conversely, nodes that generate analytic stores include: forest, gradient boosting, neural network, SVM, anomaly detection, and text mining nodes.

Scoring Methods in SAS Model Studio

SAS Model Studio supports multiple scoring approaches:

- **SAS REST API**: Model Studio automatically creates scoring APIs that can be downloaded from the pipeline comparison tab within the project pipeline
- **Programming Language Integration**: Python, R, and Java interfaces enable model scoring through familiar programming environments
- **Score Data Nodes**: Built-in nodes within Model Studio facilitate direct scoring of data tables
- **SAS Model Manager**: Provides comprehensive capabilities including scoring test execution, model storage, performance monitoring of champion and challenger models, and model analysis and comparison

Test Result Outputs

- **Alerts**: Model Manager can trigger automated notifications when predefined conditions are met, such as when model performance reaches specific thresholds
- **Dashboards**: Interactive visualizations support model testing, performance monitoring, and issue identification
- **Reports**: Comprehensive test results are generated in multiple formats including CSV, HTML, and PDF

Champion-Challenger Testing

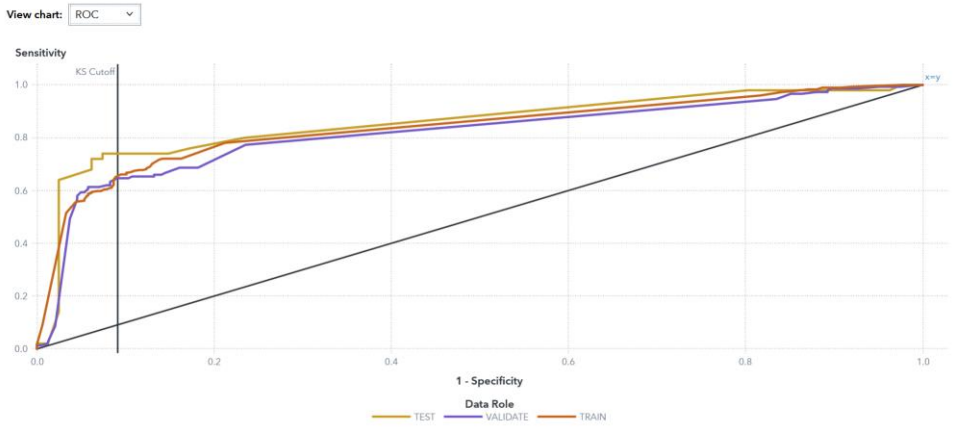
Champion-Challenger testing provides a systematic methodology for evaluating new model performance against existing production models. This process utilizes two key mechanisms:

- **Model Comparison**: Enables side-by-side evaluation of two models across various performance metrics including accuracy, F1 score, precision, and recall
- **Model Switchover**: Automatically transitions to the new model when it demonstrates superior performance compared to the current champion

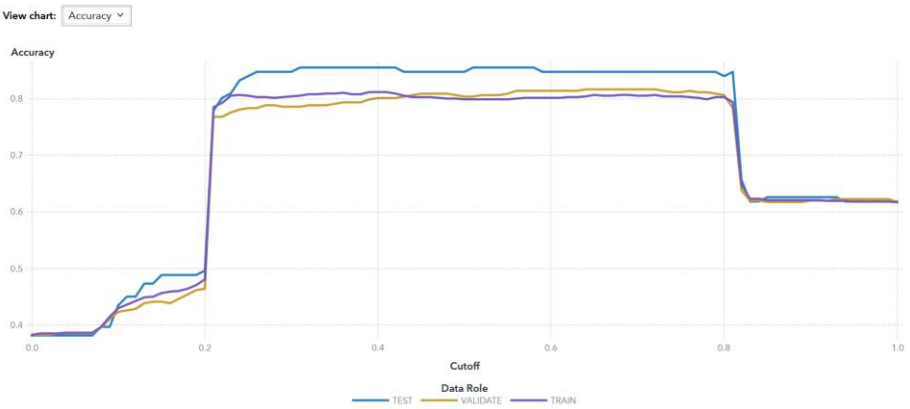
Champion-Challenger Testing Workflow

1. Register the challenger model in the system
2. Deploy the challenger model to the testing environment
3. Configure Champion-Challenger testing parameters in SAS Model Manager
4. Initiate the testing process
5. Monitor and compare performance metrics for both champion and challenger models
6. Execute model switchover to promote the best-performing model to production

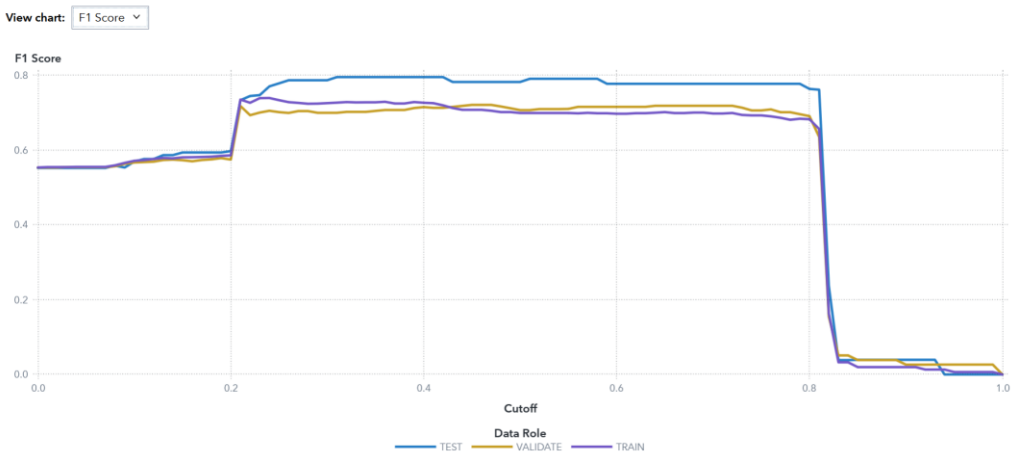
Champion model - SVM



The ROC curves demonstrate strong model discrimination capability, with all three data partitions (TRAIN, VALIDATE, TEST) exhibiting substantial area under the curve. The curves display characteristic convex shapes with rapid initial ascent, indicating high true positive rates at low false positive rates. Notably, the three curves overlap closely throughout their trajectory, suggesting minimal overfitting and robust generalization. The KS cutoff, positioned at approximately 0.1 on the specificity axis, identifies the optimal separation point where sensitivity reaches 65-70% with only 10% false positive rate. The convergence of all curves toward (1,1) confirms the model's ability to achieve near-perfect classification at higher thresholds.



Accuracy performance exhibits a plateau region between cutoff values of 0.2 and 0.85, maintaining peak values of approximately 80-85%. The TEST partition achieves marginally superior performance (~85%), while VALIDATE and TRAIN datasets perform comparably at ~80%. This consistency across partitions further validates model generalization. At extreme cutoff values, accuracy degrades predictably: below 0.2, accuracy declines to 38-39% due to excessive positive classifications, while beyond 0.85, accuracy drops to approximately 62% as the model becomes overly conservative. The broad optimal range indicates robustness to threshold variation, a desirable characteristic for practical deployment.



The F1 score metric reveals optimal performance between cutoff values of 0.2 and 0.8, with peak scores ranging from 0.70 to 0.78. The TEST partition achieves the highest F1 score (~0.78), slightly exceeding VALIDATE (~0.72) and TRAIN (~0.70) performance. This sustained plateau demonstrates effective precision-recall balance across multiple threshold values. Beyond cutoff 0.85, F1 scores exhibit precipitous decline toward zero, indicating severe recall degradation. The relatively stable F1 performance across the operational range suggests the model maintains consistent harmonic balance between precision and recall, making it suitable for applications requiring balanced error consideration.

The SVM model exhibits strong performance across all evaluation metrics with minimal overfitting. The recommended operational threshold range of 0.2-0.7 provides optimal balance between accuracy (80-85%) and F1 score (0.70-0.78), with specific threshold selection dependent upon relative costs of Type I and Type II errors in the application context.

Conclusion

This study successfully achieved its primary objective of identifying the most effective machine learning algorithm for predicting passenger survival on the RMS Titanic through comprehensive comparative analysis within the SAS Viya platform. Six distinct classification algorithms—Support Vector Machine, Artificial Neural Network, Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting—were systematically implemented, rigorously tuned, and evaluated using multiple performance metrics.

The empirical findings conclusively demonstrate that the Support Vector Machine emerged as the champion model, achieving superior performance across the majority of evaluation criteria. Specifically, the SVM attained the highest Area Under ROC (0.8557), Gini coefficient (0.7114), and exhibited the lowest multi-class log loss (0.4194) and average squared error (0.1260) on the test dataset. These results indicate that the SVM model not only possesses exceptional discriminative power but also produces highly calibrated probability predictions that closely align with actual survival outcomes. Furthermore, the consistency of performance metrics across training, validation, and test partitions confirms the model's robust generalization capability and minimal susceptibility to overfitting.

The Artificial Neural Network demonstrated competitive performance, particularly excelling in the KS statistic (0.6783), which suggests strong separation between survivors and non-survivors at optimal classification thresholds. Random Forest and Gradient Boosting models exhibited solid mid-range performance, effectively capturing complex feature interactions through their ensemble architectures. Conversely, Logistic Regression and Decision Tree models, while interpretable, demonstrated limitations in modeling the intricate non-linear relationships inherent in the Titanic dataset, as evidenced by their comparatively higher error rates and lower discriminative power.

The data preprocessing pipeline implemented in this research proved critical to model success. The systematic treatment of missing values, feature transformation, and strategic variable selection—including the exclusion of high-cardinality variables such as Name, Ticket, and Cabin—enabled all models to focus on predictive features while minimizing noise. This methodological rigor in data preparation underscores the fundamental principle that effective machine learning depends not solely on sophisticated algorithms but equally on thoughtful data engineering.

From a practical deployment perspective, the SVM model's consistent accuracy of approximately 80-85% across the operational threshold range of 0.2-0.7, combined with F1 scores ranging from 0.70-0.78, positions it as a reliable tool for binary classification tasks with similar characteristics. The model's robustness to threshold variation provides flexibility for real-world applications where the costs of false positives and false negatives may differ substantially.

In conclusion, this research demonstrates that Support Vector Machines, when properly tuned and validated, represent a powerful analytical tool for binary classification problems characterized by complex feature interactions and non-linear decision boundaries. The rigorous comparative methodology employed in this study not only identified the optimal model for the Titanic survival prediction task but also illustrated best practices in machine learning pipeline development, model evaluation, and deployment considerations within the SAS Viya ecosystem. These findings contribute to the broader understanding of algorithm selection principles and reinforce the importance of comprehensive model comparison rather than reliance on single metrics or default configurations. As machine learning continues to evolve and permeate diverse application domains, such systematic comparative studies remain essential for advancing both theoretical knowledge and practical implementation effectiveness in predictive analytics.

Discussion and Limitations

This project showed the application of multiple supervised learning algorithms for predicting survival outcomes in the Titanic dataset. While the Support Vector Machine (SVM) emerged as the most effective model. There were a variety of factors that influenced the results and generalizability of findings.

Data size and completeness was a major factor in limiting our project. The dataset includes only 891 records with a significant amount missing data such as Age and Cabin. Imputation reduced the data loss but statistical uncertainty, as imputed values may not fully represent the real distributions. Similarly the categorical simplification of complex social attributes when it comes to (e.g., socio-economic factors represented by (Pclass), potentially diminishing interpretive wealth.

Secondly the feature interactions and model assumptions also affect the predicted behavior. Models like Logistic Regression assume linear separability between predictors and costumes. Which limits their ability to capture non-linear relationships. Models such as Random Forest and Gradient Boosting performed better but often risk overfitting when depth and learning rate wasn't carefully tuned. The SVM's use of the RBF kernel helped to improve the boundary flexibility but again required careful scaling and parameter control to prevent bias for the Majority class.

Thirdly the class imbalance and bias played a subtle yet critical role. Historical records indicate that gender and class heavily influenced survival rates. Women and First Class passengers were prioritized. These patterns introduce bias that mislead predictive accuracy metrics, as models might learn historical discrimination rather than universal survival features.

Finally, computational and interpretability limitations were noted. Complex algorithms such as SVM and ANN, while accurate, have limited transparency regarding decision boundaries and feature importance. In real-world deployments this opacity can limit trust and accountability. Especially when explanations are critical.

In summary, although the models achieved high performance with metrics, their results must be interpreted within the constraints of dataset scope, feature design and historical bias. Addressing these limitations will improve both the robustness and ethical soundness of future predictive modeling.

Future Work and Recommendations

Building on the current study, below contains future research and enhancements that are recommended.

Integration of Explainable AI (XAI) Tools

The incorporation of interpretability frameworks such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations). These methods can clarify how individual features such as sex, fare, or embarked affect survival probability.

Automated Hyperparameter Optimization

The manual tuning in SAS Viya could be replaced with a Bayesian optimization or grid-search automation. This would find the most optimal parameter combinations to maximize accuracy while minimizing overfitting risk, improving the models performance.

Expanded and Enriched Datasets

Changing the dataset by including more contextual variables, such as passenger nationality, cabin location or

lifeboat access. Could increase the feature diversity and improve generalization. Applying this to other survival datasets would test the transferability of these learned patterns.

Cost-Sensitive and Ethical Modelling

Introducing asymmetric misclassification costs could make models more context aware. Noting false negatives (Missing survivors) more than false positives. This would adjust in line with the predictive priorities with ethical consolidation of real life critical scenarios.

These improvements would enhance the interpretability, adaptability and social responsibility of predictive modeling, ensuring that future versions of this project achieve both analytical precision and practical application.

Ethical and Responsible AI Considerations

Although the Titanic Dataset is historical, predictive modeling on human survival data carries ethical implications that remain relevant today. Applying responsible AI Principles to ensure that the technical accuracy is carefully balanced with fairness, transparency (As a main considerations) and accountability.

Bias and Fairness:

The dataset inherently reflects the early 20th century social inequalities, specifically gender and class privilege. Models trained on this data will be expected to reproduce these historical biases, misinterpreting them as predictors of survival merit. Ethical modeling requires recognizing that sex and class are correlates, not justifiable determinants of true survival. Future implementations that should apply fairness metrics and specific sensitivity testing to ensure equitable treatment across different demographic subgroups.

Transparency and Interpretability:

High performing models like SVM and ANN's often operate as "Black Boxes" in that they obscure how predictions are made. This limits understanding and oversight. Incorporating explainability techniques (i.e SHAP values) and clear model documentation enhances user trust and accountability. Key Aspects of responsible data science.

Data Governance and Privacy:

Although the Titanic dataset is publicly available, other similar predictive projects involving modern personal data would require a much higher level of confluence with an array of different privacy regulations. Also the audit trails within the SAS Model Manager would help maintain data integrity and ethical compliance.

Sustainability and Social Impact:

Finally, while our analysis serves for simple education purposes, other similar models can influence high stakes decisions making in healthcare, transportation, or emergency management. Responsible AI practice demands that systems are validated for fairness, continuously monitored and regularly retrained to prevent discriminatory or harmful outcomes.

Ensuring ethical safeguards at every stage, from preprocessing to eventual deployment ensures that predictive analytics advances societal benefit without perpetuating bias or inequality.

References

Hasan, M. M., & Hasan, M. F. (2024). *Using Titanic dataset for comprehensive machine learning model training. International Journal of Innovative Science and Research Technology, 9*(10), 1–7. https://www.ijisrt.com/assets/upload/files/IJISRT24OCT1256.pdf

SAS Institute Inc. (2023). *Overview: SVMACHINE procedure (Viya 8.1)*. In *SAS Visual Data Mining and Machine Learning 12.1: Procedures*. https://documentation.sas.com/doc/en/vdmmlcdc/8.1/casml/viyaml_svmachine_overview.htm

Tabbakh, T., Rout, M., & Rout, J. K. (2021). *Analysis and prediction of the survival of Titanic passengers using machine learning algorithms*. In S. Agarwal & R. Bisht (Eds.), *Proceedings of International Conference on Computing and Communication Systems* (pp. 363–371). Springer. https://doi.org/10.1007/978-981-15-4218-3_29

Van Belle, V., Pelckmans, K., Suykens, J. A. K., & Van Huffel, S. (2018). *Support vector machines for survival analysis*. *The R Journal, 10*(1), 443–455. https://journal.r-project.org/archive/2018/RJ-2018-005/RJ-2018-005.pdf

Al-Hayik, U. H. S., & Abu-Naser, S. S. (2023). Chances of survival in the Titanic using ANN. International Journal of Academic Engineering Research, 7(10), 17–21. https://philpapers.org/archive/ALHCOS.pdf

Harrell, F. E. (2015). Logistic Model Case Study 2: Survival of Titanic Passengers. In Regression Modeling Strategies (pp. 291–310). Springer International Publishing AG. https://doi.org/10.1007/978-3-319-19425-7_12

Reza, S., Sarwar, B., Nawaz, R. R., & Haq, S. M. N. U. (2021). Application of logistic regression on passenger survival data of the Titanic liner. *Journal of Accounting and Finance in Emerging Economies, 7*(4), 861–867. https://doi.org/10.26710/jafee.v7i4.1994

Sreenivasulu, L., & Chandrasekar, V. (2025). Prediction of titanic data analysis using logistic regression compared with decision tree for better accuracy. In K. Ramasundram, K. Sridhar, Y. Jaganathan, K. Senguttuvan, R. Srinivasan, G. Varudharajan, & J. Duraisamy (Eds.), AIP conference proceedings (Vol. 3267, Number 1). American Institute of Physics. https://doi.org/10.1063/5.0272773

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Amalia, A. E., & Rahayu, C. (2025). Comparison of Machine Learning Classification Models in Predicting The Titanic Survival Rate. International Journal of Computer Science and Humanitarian AI, 2(1), 37–41. https://doi.org/10.21512/ijcshai.v2i1.12163

Huang,S. (2024). Processing and Comparison of GBoost, XGBoost, and Random Forest in Titanic Survival Prediction. Applied and Computational Engineering,102,175-182. https://doi.org/10.54254/2755-2721/102/20241195

Figure 1
IBM. (n.d.). *Support vector machine*. IBM. https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/creative-assets/s-migr/ul/g/8f/27/3-1_svm_optimal-hyperplane_max-margin_support-vectors-2-1.png

Figure 2
ScienceDirect. (n.d.). *Artificial neural network*. Elsevier. https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network

Figure 3
IBM. (2025). *Random forest diagram*. IBM. https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/cdp/cf/ul/g/50/f9/ICLH_Diagram_Batch_03_27-RandomForest.png

Figure 4
IBM. (2025). *Decision tree diagram*. IBM. https://www.ibm.com/content/dam/connectedassets-adobe-cms/worldwide-content/cdp/cf/ul/g/df/de/Decision-Tree.png

Appendix :

Assigning Group Work:

Model Training: SVM	Zichen Shao
Model Training: ANN	Zichen Shao
Model Training: Logistic Regression	JunHao Zeng
Model Training: Random Forest	William Wickham
Model Training: Gradient	Kaung Khant Kyaw
Model Training: Decision Tree	JunHao Zeng
Executive Summary	Zichen Shao
Data Preprocessing	Zichen Shao
Methodology:	
SVM	Kaung Khant Kyaw
ANN	Kaung Khant Kyaw
Logistic Regression	Kaung Khant Kyaw
Random Forest	Kaung Khant Kyaw
Gradient	Kaung Khant Kyaw
Decision Tree	Kaung Khant Kyaw
Algorithm Selection	Zichen Shao / Kaung Khant Kyaw
Model Tuning:	
SVM	Zichen Shao
ANN	Zichen Shao
Logistic Regression	JunHao Zeng
Random Forest	William Wickham
Gradient	Kaung Khant Kyaw
Decision Tree	JunHao Zeng
Experiment Results	

SVM	Kaung Khant Kyaw
ANN	Kaung Khant Kyaw
Logistic Regression	JunHao Zeng
Random Forest	Kaung Khant Kyaw
Gradient	Kaung Khant Kyaw
Decision Tree	JunHao Zeng / Zichen Shao
Model comparison	Zichen Shao
Best Model	Zichen Shao
Model Deployment	Zichen Shao
Conclusion	Zichen Shao
Discussion and Limitations	Michael Zervos
Future Work and Recommendations	Michael Zervos
Ethical and Responsible AI Considerations	Michael Zervos
Presentation	Michael Zervos