Observations of many machine learning models on the Titanic Kaggle problem

Owen Roseborough
Dept. of Computer Science
Western University
London, Canada
orosebor@uwo.ca

Adam Nelson Palmer Ursenbach
Dept. of Engineering
Western University
London, Canada
aursenba@uwo.ca

Alan Kuang
Dept. of Computer Science
Western University
London, Canada
akuang4@uwo.ca

Aarish Lakhani
Dept. of Engineering
Western University
London, Canada
alakha33@uwo.ca

***Abstract*– Let's say that we have a ship, like the Titanic, that has sunk, or more recently the Costa Concordia and survivors and remains are found. However, there are still a few hundred people missing, and we would like to know whether they are likely to have survived in order to focus rescue efforts to those areas, as time is of the essence.**

In order to solve this problem, which is an interpretation of the famous Titanic Kaggle Challenge with real-world applications as mentioned above, our project is aimed to develop a predictive model for which passengers would survive the Titanic disaster using passenger data[1]. We used Logistic Regression, Random Forests, Support Vector Machines, k-nearest neighbours, and k-means to attempt to capture linear and complex dependencies in the data and find the most accurate solution in order to survey which model would work best in these types of situations.

I. Introduction

From the Kaggle problem, our task is to predict which passengers will survive based on binary classification from feeding the selected model. Our research goal right now is to compare and pick the best algorithm for predicting survivors and casualties out of the various machine learning methods laid out in the Abstract.

We eventually concluded that the Random Forests model achieved the highest predictive accuracy on the testing data, outperforming the Logistic Regression, SVM, both k-NN models, and k-Means. A surprise was that k-NN in both tested forms performed exceptionally at 82.12%, beating SVM while being simple to implement.

Realistically, our results and methods would not have much of an impact as this Kaggle Challenge has been pushed to its absolute limits and has significantly better solutions than ours for this challenge, like Bhardwaj's method using the random forest model, with isolated labels which scored within the top 9%[2].

However, as a rule of thumb the more a model can classify one thing i.e. the Titanic better, the more overfitted the model is and the less likely it would be useful in another disaster, say, if we applied that to the Costa Concordia or some other ship disaster, the model would perform poorly as the training is overfit. Overfitting is described in more detail on the last slide of Lecture 3 Regression in this course[3].

Our solutions this paper could potentially be more generalizable than these specialized high percentages if trained on more relevant data pertaining to modern ship disasters. They can possibly be useful in the near future's maritime accidents.

II. Background and Related Work

As mentioned in previous sections, this is a Kaggle Challenge with over 60000 submissions[1], but in the real world, the closest thing to this scenario would be human casualty prediction. A paper on casualty prediction using a two-step machine learning method utilizes a very similar process to ours, Data Preprocessing, then feeding it into the Machine Learning model[4]. Their dataset is from various terrorist attacks which have large amounts of lives lost[4, pp.244-245].

In their particular case, they do "(1) label defining; (2) feature selection; (3) missing data processing; and (4) feature processing"[4, p.244] for preprocessing before utilizing a set of machine learning models "the support vector machine (SVM), random forest (RF), gradient boosting decision tree (GBDT) and XGBoost models"[4, p.244] alongside 10-fold cross-validation [4, p.244].

Hu, Hu, & Hou's paper is near identical to our approach except for k-NN without cross-validation, where we do data preprocessing by paring it down and utilize 5-fold cross-validation. They do this to predict deaths from terrorist attacks[4] conversely we are here to figure out .

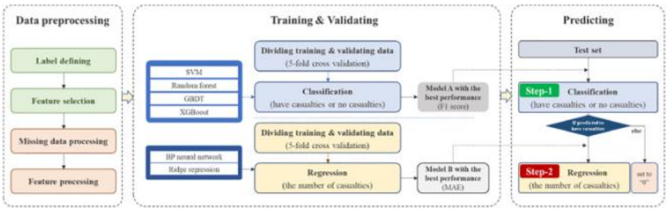
Data preprocessing
Label defining
Feature selection
Missing data processing
Feature processing
Training & Validating
SVM
Random forest
GBDT
XGBoost
BP neural network
Ridge regression
Dividing training & validating data
(5-fold cross validation)
Classification
(have casualties or no casualties)
Model A with the best performance
(F1 score)
Dividing training & validating data
(5-fold cross validation)
Regression
(the number of casualties)
Model B with the best performance
(MAE)
Predicting
Test set
Step-1
Classification
(have casualties or no casualties)
If predicted to have casualties
else
Step-2
Regression
(the number of casualties)
set to "0"

Figure 1. Hu, Hu, Hou's model for their approach for casualty prediction

Another related work is Fang et al.'s Earthquake casualty prediction where they use a shallow neural network to predict earthquake casualty numbers[5]. Their process is feeding data of the buildings that collapsed to determine the number of deaths by training it on past earthquakes[5, pp.2-4] into their shallow neural net to determine deaths and plot a rescue plan[5].

This approach is similar to ours, as Fang et al. also preprocesses data, e.g. magnitude, deaths, city, and year and feeds it to their model which calculates the deaths and people buried alive, and then they do some additional processing to get the rescue plan which is not relevant to our particular approach.

With those two related examples, we can safely conclude that our proposed methodologies in the introduction are on the right track.

III. Method

A. Research objectives

Hypothesis: Analytically, we should be able to determine that out of the models, the best to predict passenger survival and death. The more complex models, i.e. SVM or Random Forests, should yield a more accurate result, while the simpler models like k-NN should yield a significantly less precise result for classifying the passenger's survival.

We have two objectives.

1. O1 Train the selected models SVM, Random Forest, etc., on our preprocessed Titanic Kaggle dataset in order to determine the best model to predict survivors for maritime accidents.
2. O2 Compare the real-world ramifications of this model against all other models, whether they overfit or are generalizable.

B. Research Methodology

In data preprocessing, we extracted punctuation from the Name column so that we were left with only the words. Then we extracted the titles from each person like 'Mr', 'Mrs', 'Miss', 'Dr' and labelled them in a new column, the rationale being that the ML models may be able to pick up differences in outcomes between married women and unmarried women, or that doctors had higher survival rates due to their medical experience. We also label the sex column to be either 0 for women or 1 for men. We separated the ticket field into the ticket number and ticket prefix. We provided a new column that standardized the fare. Our 'y' or classification label 'Survived' was provided in either 0 or 1 from the dataset, so no need for preprocessing.

We extracted the correlations between survival and other variables from the correlation matrix and found that survival is heavily correlated to fare price as seen in figure 2.

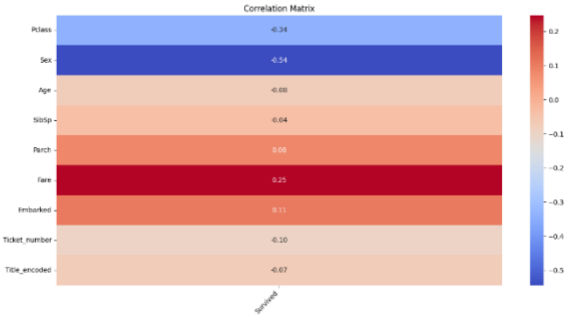

Figure 2. feature correlations extracted from feature matrix

In our dataset we have 549 who perished and 342 who survived, so if our model only predicted perished, it would be correct 62% of the time. This was our first baseline to test our models against.

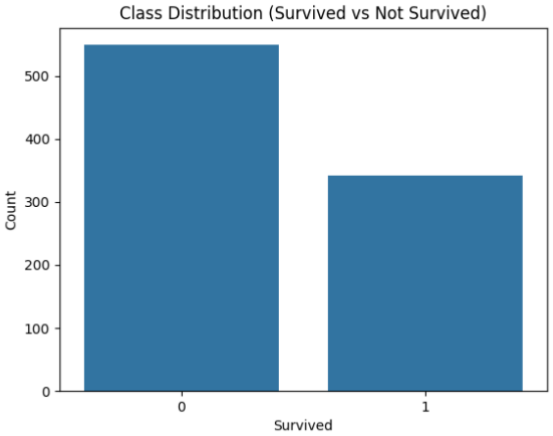

Figure 1. Survived vs Perished

Below we plot Fare, Pclass and Age colored by survival.

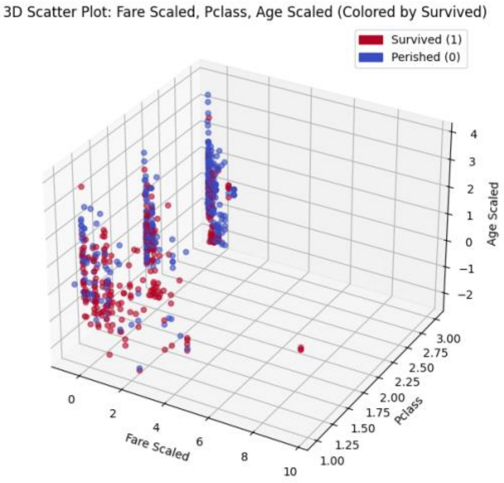

Figure 3. Fare vs Pclass vs Age

From the plot, we observe that higher class (lower Pclass value) and youth correlate with survival.

Next, we plot the Fare, Age, and SibSp (# of family members on Titanic "Sibling/Spouse").

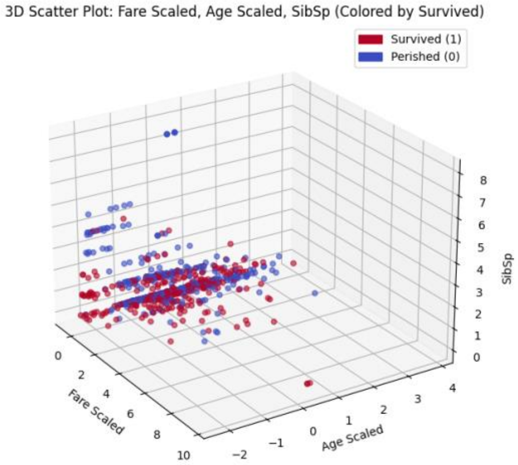

Figure 4. Fare vs Age vs SibSp

Being younger has a higher correlation with survival, as does having fewer family members (SibSp).

Finally, we plot Pclass, Age, and Sex.

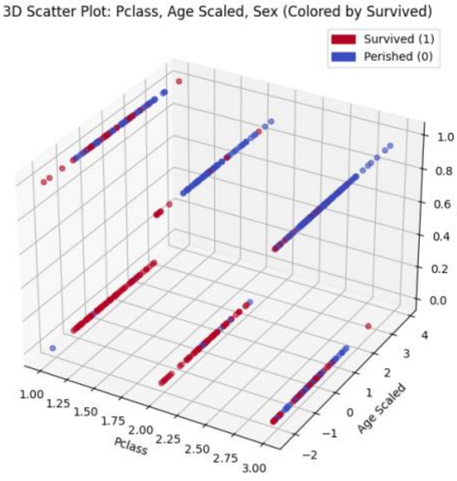

Figure 5. Pclass vs Age vs Sex

Gender (Sex) is highly correlated with survival, with the men (1) having fewer survivors than the women (0).

C. Logistic Regression

Since we are trying to do binary classification into 'y' the survival rate, we are interested in which features have more significant impacts on surviving. We implemented a logistic model with backwards feature selection, achieving an accuracy of 72% (0.72). Keeping half of the features left us with the most important ones: 'Pclass' with coefficient -0.98, 'Sex' with -2.86, 'SibSp' with -0.27, 'Parch' with -0.31 and 'Embarked' with 0.2. The calculated logarithmic probability from the Logistic model are defined mathematically as:

$$\log\left(\frac{\frac{1}{1+\ e^{-\left(B_{0+}\,B_1Pclass+\,B_2Sex+\,B_3SibSp^{+\cdots}\right)}}}{1-\ \frac{1}{1+\ e^{-\left(B_{0+}\,B_1Pclass+\,B_2Sex+\,B_3SibSp^{+\cdots}\right)}}}\right)=z$$
$$=\ \left(B_{0+\,B_1Pclass+\,B_2Sex+\,B_3SibSp}+\cdots\right)$$

The coefficient for Sex indicates that female passengers (value 0) have higher probability of surviving. Similarly with which class the passenger is, the closer to 0 (First class), the higher the number will be, and the higher the probability of survival, but the higher the Pclass variable (lower class in english semantics) is, say 2, (Third class), the more negative the number will be, indicating a lower probability of survival.

D. Random Forests

Random forests are a model consisting of multiple decision trees that are built to prevent overfitting by having more trees so that they are more generalizable[6].

It handles regression and classification well[6], and according to IBM, "Feature bagging also makes the random forest classifier an effective tool for estimating missing values as it maintains accuracy when a portion of the data is missing."[6]

From IBM's website, the reason why this model is good for classifying survivors is

> Easy to determine feature importance: Random forest makes it easy to evaluate variable importance, or contribution, to the model[6].

This is essential as we want to see which elements are important.

Overall, Random forests seem to be a good fit for our dataset as we want a model that predicts well and does not overfit, as overfitting is a massive problem that causes inaccuracies with new data.

To implement it we train a random forest classifier on the data. We used a grid search to tune the hyperparameters to the optimal values, varying the number of estimators, the maximum depth of the trees, the minimum number of samples required to split a node, and the minimum number of the samples needed to be at a leaf node to find the optimal random forest configuration.

E. SVM

Support Vector Machine models looked to be good candidates for our problem because they can handle complex, non-linear relationships between features and offer good generalization through margin maximization and regularization.

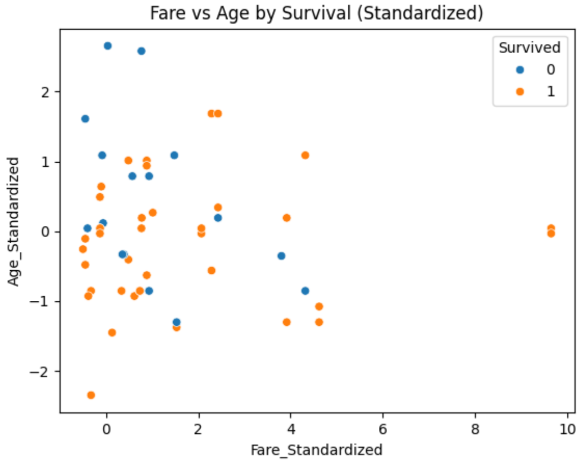

Figure 6. Age vs Fare and Survivability Plot

The relationship between Fare, Age and Survivability is graphed below. As we can see, there is not a linear separation between the two groups of points with respect to survivability.

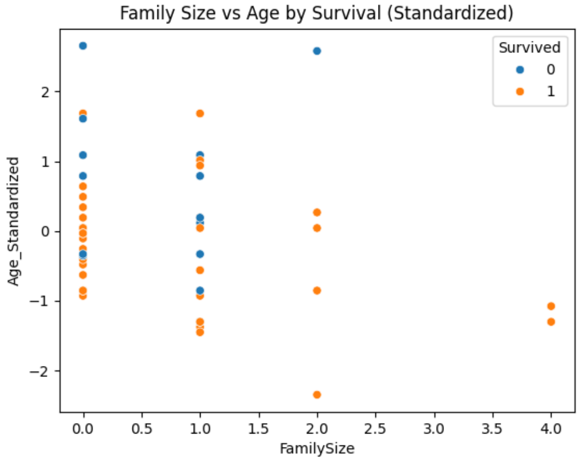

Figure 7. Age vs Family Size and Survivability Plot

Similarly, we have the relationship between Age, Family Size and Survivability below, no linear separability is evident.

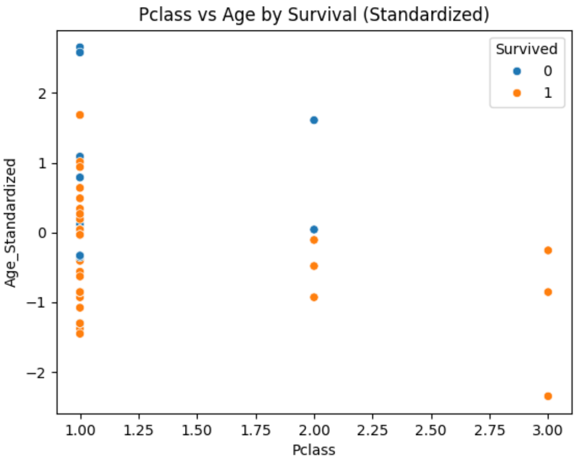

Figure 7. Age vs Pclass and Survivability Plot

We tried fitting a Support Vector Machine model to the data. We did a search through 50 random groupings of hyperparameters and used 5-fold cross validation to evaluate each parameter combination. We used kernel functions Radial Basis, Polynomial, and Sigmoid.

Our training process identified the best-performing hyperparameters as a polynomial kernel with a degree of 2, gamma = 0.1, coef0 = 1, and C = 10. We achieved an accuracy of 0.82. The mathematical equation for our kernel is shown below.

$$K(x, x') = (0.1x^T x' + 1)^2$$

This function computes a non-linear similarity between each pair of input vectors x and x', allowing the model to learn quadratic decision boundaries. These kernel values are then used in the SVM's decision function:

$$f(x) = \sum_i \alpha_i y_i K(x_i, x) + b$$

Where:

- $\alpha_i - learned\ weights\ from\ training$
- $y_i - class\ label\ of\ the\ support\ vector$
- $x_i - support\ vector$
- $b - bias\ term$

The sign of f(x) determines the prediction: if f(x) > 0, we predict class 1 (survived); otherwise, class 0 (did not survive).

F. k-NN

We also explored a k-Nearest Neighbors (k-NN) approach due to its ability to model non-linear decision boundaries without making assumptions about the data distribution. In this context, the model predicts a passenger's survival based on the survival outcomes of their most similar neighbors in the feature space. k-NN is particularly intuitive for this dataset, as passengers with similar characteristics (e.g., class, age, family size) are often likely to have similar outcomes.

The age and fare features were standardized to prevent differences in magnitude between their values and other features to skew the algorithm. The Pclass, SibSp, and Parch features were also standardized because their values have order to them and thus can be represented as distances, for example a Pclass of 1 and Pclass of 3 should be further apart in distance than a Pclass of 1 and 2.

A grid search was performed over three hyperparameters on number of neighbors, weights, and distance calculation. The number of neighbors was varied from 1 to 21, the weights were either uniform or distance, and the distance calculation was either Manhattan or Euclidean.

As k-Nearest Neighbors (k-NN) is known for its simplicity, we also explored a non-cross-validated implementation to assess how its performance compares to our cross-validated model. In this version, we still normalized the test data to ensure consistent distance calculations and prevent skew from feature scale differences, and utilized Euclidian distance.

G. Unsupervised Learning: K-means Clustering

While our primary goal was to use supervised models to predict survival, we also explored an unsupervised approach using K-Means clustering to identify natural groupings in the Titanic dataset without using the target variable (Survived). Our goal was to see if the unsupervised clusters would align meaningfully with actual survival outcomes.

Mathematical Rationale for K-means Clustering
K-Means clustering is an unsupervised learning algorithm that partitions the data into clusters by minimizing intra-cluster variance. The method attempts to find groupings of passengers who are similar based on features such as age, fare, class, and family size.
The objective function for K-Means can be defined as:

$$\boldsymbol{J = \sum_{j=1}^{k} \sum_{x \in C_j} ||x_i - \mu_j||^2}$$

Where:

- k is the number of clusters
- $C_j$is the set of data points assigned to cluster j
- μ_j is the mean(centroid) of cluster j
- $||x_i - \mu_j||^2$is the squared Euclidean distance between a data point and its assigned cluster center

The algorithm runs in two main steps:

1. Assignment step – Each data point is assigned to the nearest cluster centroid.
2. Update step – Each centroid is updated to be the mean of the points in its cluster.

These steps are repeated until the algorithm converges (i.e., cluster assignments no longer change significantly or the cost function improvement plateaus).
This approach is well-suited to our dataset for a few key reasons:

- It helps us explore latent structure in the feature space, revealing how passengers may naturally group together based on shared traits.

- It is computationally efficient, making it practical for iterative experimentation across different values of k.
- It aligns well with our interest in visualization – using PCA to project the clusters into two dimensions allows us to inspect the separation visually.

That said, a limitation is that K-Means assumes that clusters are spherical and roughly equal in size, which may not always reflect real-world distributions. Still, it provides a simple and interpretable way to explore structure in the data, and it gives us a baseline to compare how well natural clusters align with actual survival outcomes.

We first applied Principal Component Analysis (PCA) to reduce the dimensionality of the feature space for visualization. The initial clustering was done with k=2, aiming to roughly separate the passengers into two groups.

Clusters were relatively well-separated in the PCA space. To evaluate how well these clusters aligned with actual survival, we compared them using a confusion matrix. This yielded moderate alignment, with 440 actual non-survivors and 173 actual survivors correctly grouped.

Formulas

For each model we will be displaying these statistics to gauge its effectiveness.

$$\text{Precision} = \frac{TP}{TP + FP}.$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

IV. Results

A. Logistic Regression

Below we have a statistical summary table for our model coefficients:

	B	p	Exp(B)
Const	3.723	5.226e-23	41.392
Pclass	-1.016	1.020e-15	0.362
Sex	-3.030	2.577e-38	0.048
SibSp	-0.295	1.399e-02	0.744
Parch	-0.156	2.324e-01	0.855
Embarked	0.204	2.196e-01	1.226

Table 1. Logistic Regression Statistical Summary Table

After running it through the logistic regression model, we have a B value of 3.723 for our intercept term; this means if all features are at 0 (female, first class, no children or parents, embarked from port S), the odds of survival would be approx. 97.6% (0.976). We infer from the dataset that mothers would be motivated to save their children, first-class passengers have priority, females have priority, etc. The P values for Pclass and Sex are especially small. This indicates there is virtually no chance that their effects on the outcome happened randomly. For the Exp(B) column, we have a value of 0.744 for the SibSp feature. This means that having more family members on the Titanic would reduce your chance of survival by 25.6% (0.256)

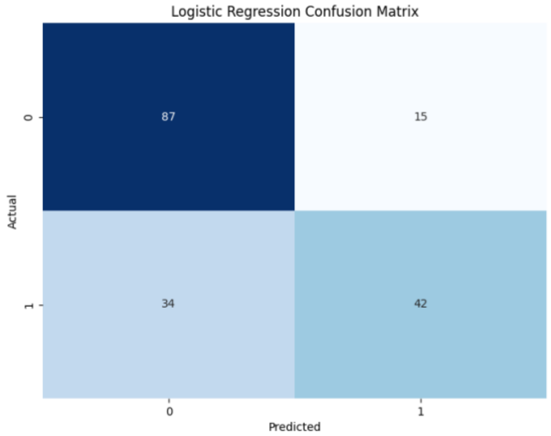

Figure 9. Logistic Regression Confusion Matrix

The logistic regression model assumes a linear relationship between the features and the logarithmic odds of survival. But in our dataset, the relationship between fare and survival may not be linear, with jumps between first and lower class fares. The relationship between age and survival may also not be linear, with children being prioritized over all other age groups. Logistic regression also does not handle interactions between features, and is not robust to outliers since it tries to find a best linear boundary. For these reasons, we do not think this is the best model.

B. Random Forests

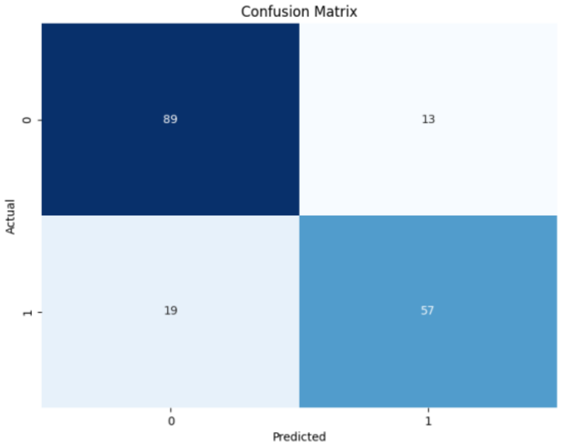

Figure 10. Random Forests Confusion Matrix

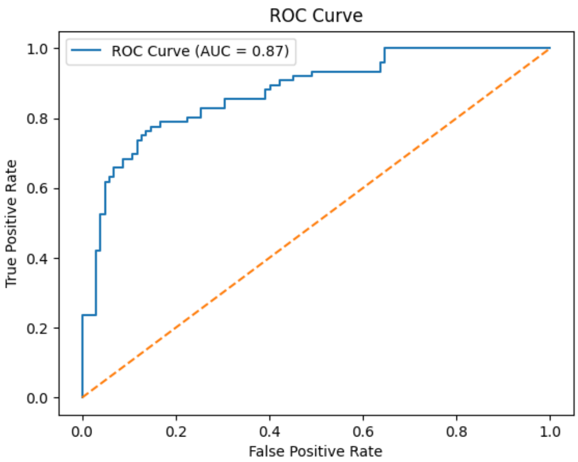

Figure 11. Random Forests ROC Curve Plot

We achieved an accuracy of 0.82 and a mean cross-validation accuracy of 0.84 with the random forest model. Our summary statistics are below.

Survival	precision	recall	F1-score	Support
0	0.87	0.82	0.84	102
1	0.75	0.81	0.78	76
Accuracy:	0.82		Total S:	178

Table 2. Random Forests Statistical Summary Table

The model had slightly better performance predicting those who perished (recall = 0.82) versus those who survived (recall = 0.81), this is to be expected since our classes are slightly unbalanced, with more perishing than surviving. The F1-scores 0.85 and 0.78 for classes 0 and 1 respectively indicate the model performs decently across both classes. The confusion matrix is below.

We obtained an AUC of 0.87, which indicates that the model can distinguish between survivors and non-survivors with 87% accuracy across all possible classification thresholds. We have a steep jump as we move from left to right across the graph, meaning that we can predict people who perished without predicting people perishing falsely. The most important features are below.

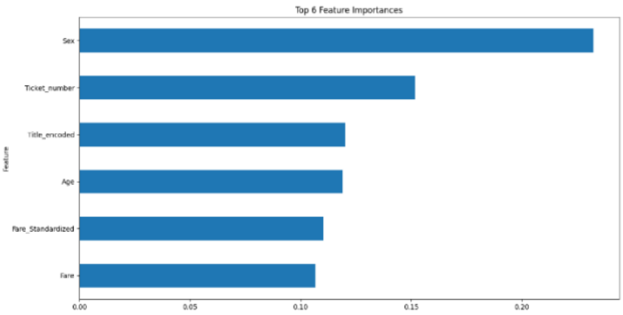

Figure 12. Random Forests Feature Importance Plot (Top 6)

From this graph, the most important features were Sex, Ticket_number, Title_encoded, Age, Fare_Standardized, and Fare. This indicates that when the Random Forest model split the data based on these features — particularly Sex — it achieved the greatest improvements in node purity, meaning these features were the most helpful in separating survivors from non-survivors. While Sex contributed the most, the other top features also played significant roles, though to a lesser degree. These results largely align with the findings from our logistic regression model, especially when we consider the conceptual overlap between Pclass and Fare, and between Sex and Title_encoded.

While the Random Forest model performed well, we wanted to try for a better performing classifier.

C. SVM

Using a polynomial kernel of degree 2 allows the model to learn quadratic decision boundaries, capturing non-linear relationships between features. The regularization parameter C = 10 strikes a balance between fitting the training data well and avoiding overfitting. This setting encourages the model to form a confident decision boundary while still generalizing to unseen data. The gamma value of 0.1 provides a moderate level of curvature in the decision surface — not too sharp and not too smooth — helping the model capture meaningful patterns without overfitting noise.

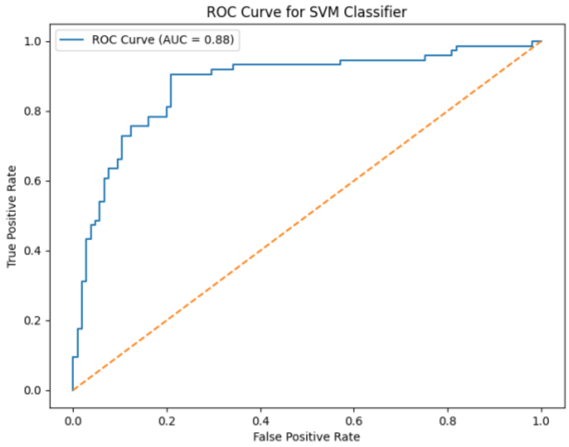

Figure 13. SVM ROC Curve Plot

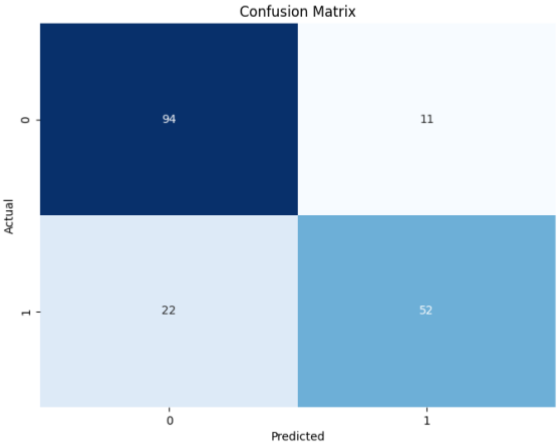

Figure 14. SVM Confusion Matrix

We achieve an ROC curve comparable to the Random Forests model. Our statistical summary table is below.

Survival	Precision	Recall	F1-score	Samples
0	0.90	0.81	0.85	105
1	0.70	0.83	0.76	74
Accuracy:	0.82		Total S:	179

Table 3. SVM Statistical Summary Table

The model had better performance predicting those who did not survive (precision = 0.90) compared to those who did survive (precision = 0.70). This is expected, as the dataset is slightly imbalanced, with more passengers perishing than surviving. The F1-scores of 0.85 for class 0 and 0.76 for class 1 indicate that the model performs reasonably well across both classes, though with slightly stronger performance for predicting non-survivors.

D. k-NN with Cross Validation

Our best model used 15 neighbors, uniform distance weights, with a Manhattan distance calculation, giving us a mean cross validation accuracy of 0.814.

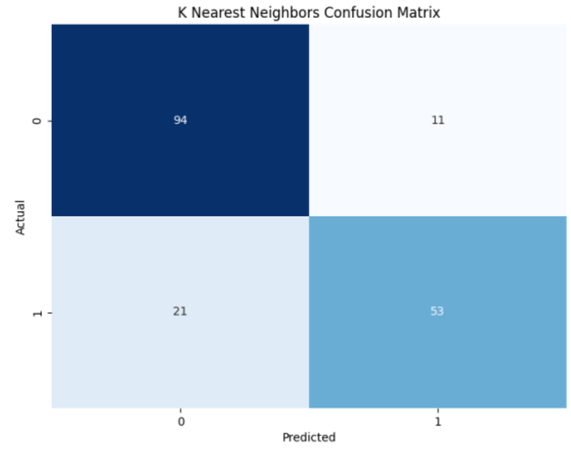

Figure 15. k-NN Confusion Matrix

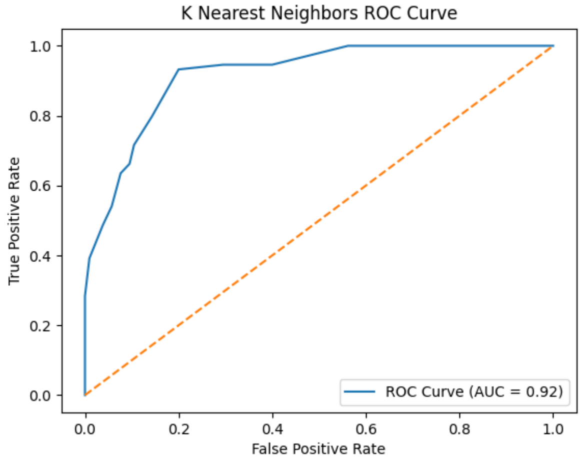

Figure 16. k-NN ROC Curve

In the bottom left of our ROC curve, we start with a high threshold for predicting survival, and we thus have very few true positives. As we move to the right the threshold is lowered to predict survival, points that have fewer and fewer neighbors who survived are allowed to be predicted as positive. The relatively steep jump is indicative of the success of our model in predicting true positives without admitting many false positives. The statistical summary table is below:

Survival	precision	recall	F1-score	Samples
0	0.90	0.82	0.85	105
1	0.71	0.83	0.77	74
Accuracy:	0.82		Total S:	179

Table 4. k-NN Statistical Summary Table

E. k-NN No Cross Validation

We implemented this version using PyTorch, utilizing GPU acceleration to improve computational speed. After computing the Euclidean distance between each test point and every other normalized training point, we used the topk function to retrieve the k nearest neighbors. Through experimentation with values of k from 1 to 13, we observed that k = 3 produced the best result.

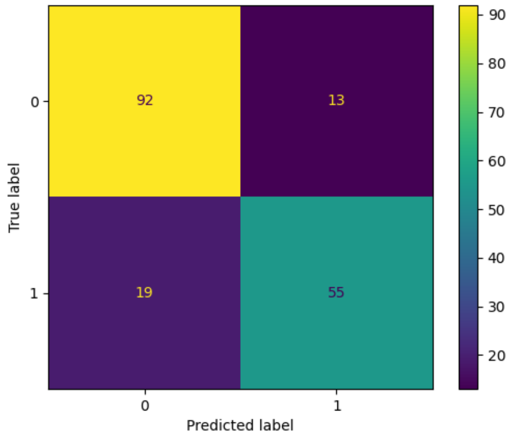

Figure 17. k-NN non cross validated confusion matrix

Despite the lack of cross-validation, the model is still almost on par with k-NN cross validated. It produced slightly more false positives (2 additional) and fewer false negatives (2 fewer) compared to the cross-validated k-NN implementation. This result suggests that while cross-validation offers marginal gains in reducing false positives, the overall predictive performance remains similar.

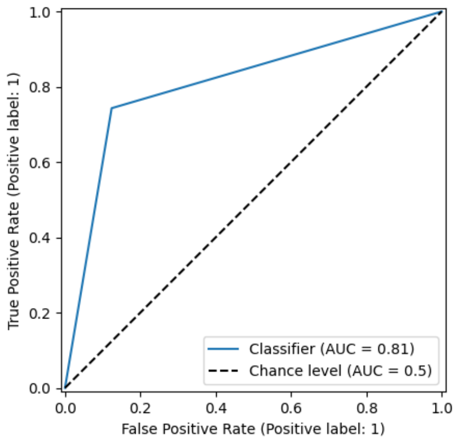

Figure 18. k-NN non cross validated ROC curve

The ROC curve for this non-cross-validated model reveals two near-linear segments. The initial steep segment on the left indicates the model's strength in minimizing false negatives, which is particularly desirable in a survival prediction context. Ideally, a perfect classifier would generate an ROC curve consisting of a vertical line from (0,0) to (0,1), followed by a horizontal line to (1,1).

The only thing of note is that the AUC for the non cross validated one is lower than the cross validated one, however this does not seem to affect the performance as much.

Here is the statistical summary table for the non-cross validated k-NN:

Survival	Precision	Recall	F1-score	Samples
0	0.88	0.83	0.84	105
1	0.74	0.81	0.77	74
Accuracy:	0.82		Total S:	179

Table 5 k-NN non cross validated Statistic Summary table

As with the cross-validated k-NN model, the non-cross validated version is better at deciding survivals over deaths. Nevertheless, it achieved an impressive accuracy of 82.12%, comparable to that of the SVM model and virtually identical to the cross-validated k-NN implementation. This demonstrates that even without advanced techniques, a basic k-NN model remains a remarkably strong baseline in this classification problem.

F. Unsupervised Learning: k-Means Clustering

PCA Projection and Initial Clustering (k=2)

We first applied Principal Component Analysis (PCA) to reduce the dimensionality of the feature space for visualization. The initial clustering was done with k=2, aiming to roughly separate the passengers into two groups. Clusters were relatively well-separated in the PCA space. To evaluate how well these clusters aligned with actual survival, we compared them using a confusion matrix. This yielded moderate alignment, with 440 actual non-survivors and 173 actual survivors correctly grouped.

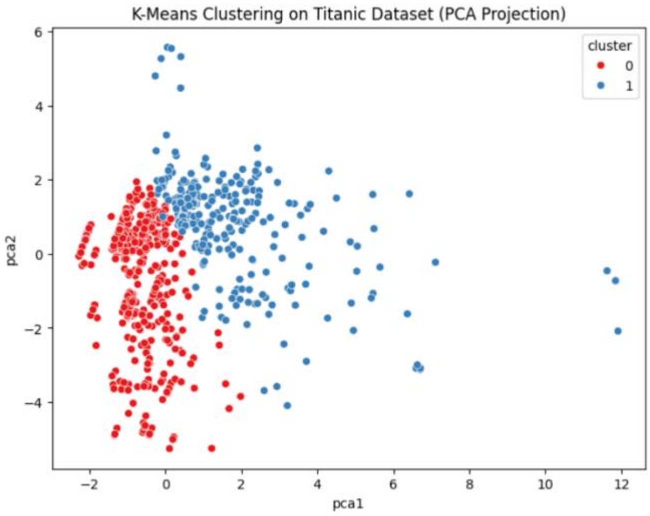

Figure 19: PCA projection of Titanic passenger data clustered using K-Means (k=2).

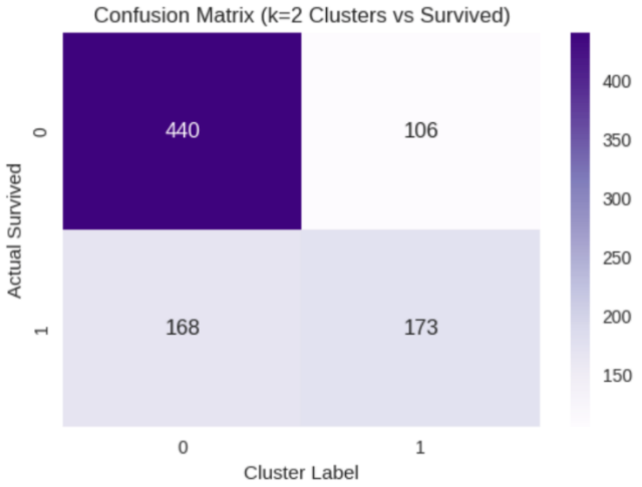

Figure 20: Confusion matrix comparing K-Means clustering results (k=2) with actual survival labels.

Choosing the Optimal Number of Clusters

We used the Elbow Method to find the optimal number of clusters by plotting distortion score vs. number of clusters. The "elbow" appears at k=3, suggesting that 3 clusters may provide better structure without overfitting.

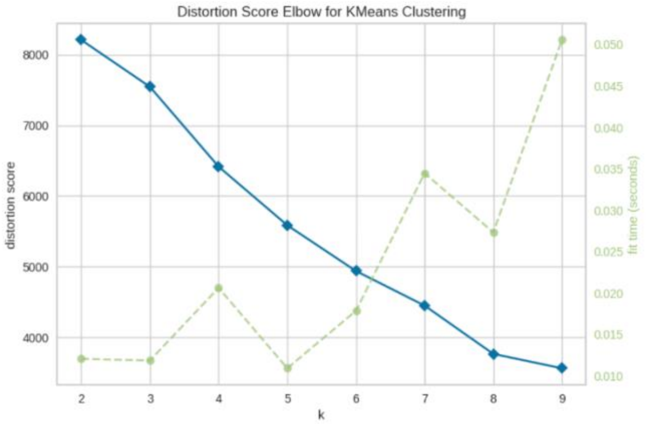

Figure 21: Elbow plot showing distortion score versus number of clusters (k) for K-Means clustering.

K=3 Clustering Results

With k=3, we observed more detailed groupings. The clusters began capturing different types of passengers (e.g., maybe based on class or age), but alignment with actual survival decreased slightly compared to k=2.

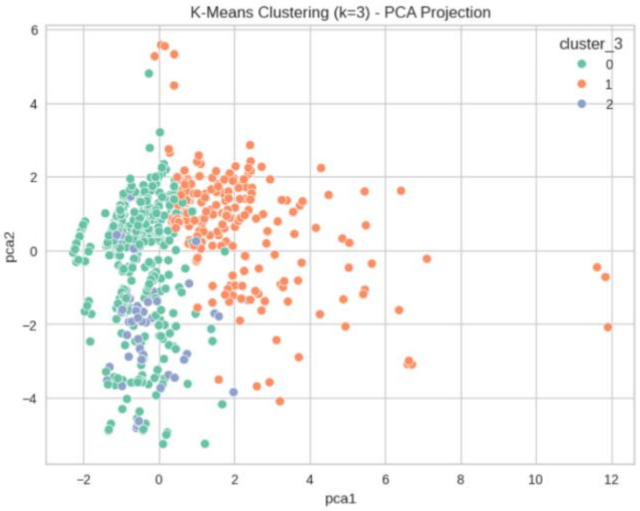

Figure 22: PCA projection of Titanic passenger data clustered using K-Means with k=3.

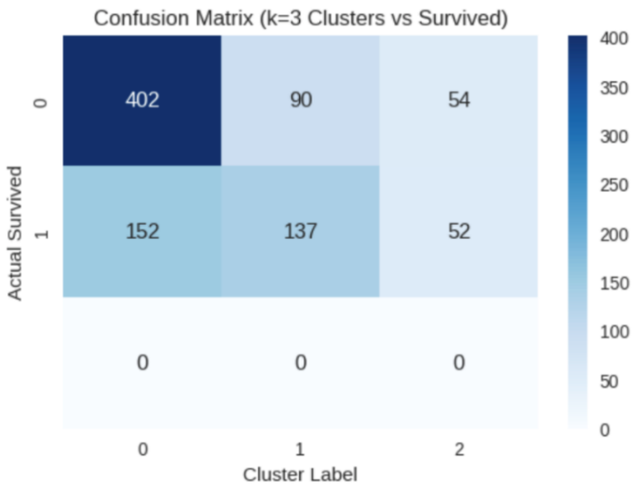

Figure 23: Confusion matrix comparing K-Means clustering results (k=3) with actual survival labels.

K=4 Clustering Results

Testing with k=4, we obtained even finer groupings. With more clusters, there was greater granularity, but interpretability and survival alignment did not clearly improve.

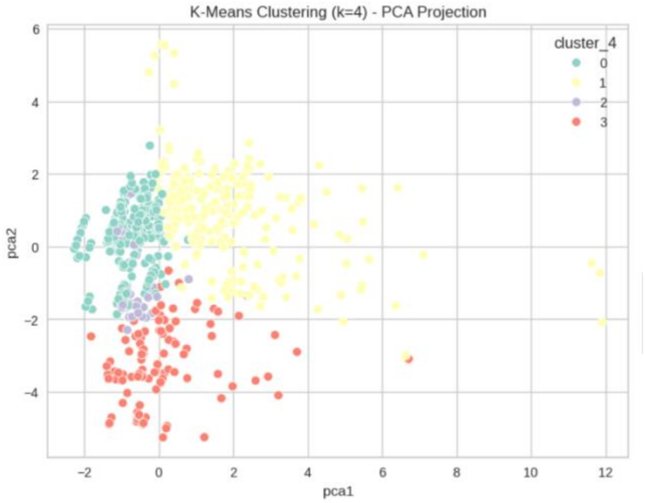

Figure 24: PCA projection of Titanic passenger data clustered using K-Means with k=4.

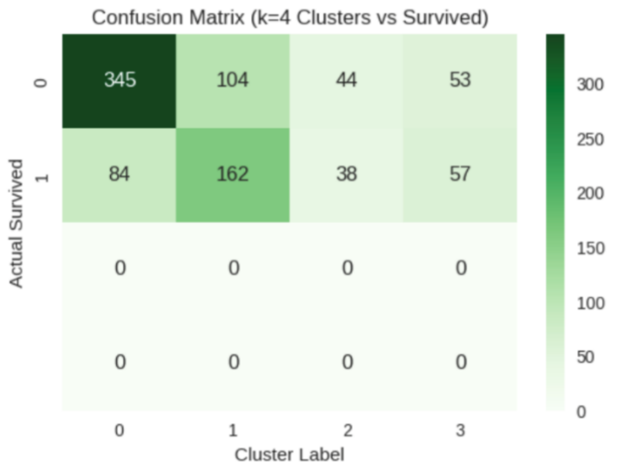

Figure 25: Confusion matrix comparing K-Means clustering results (k=4) with actual survival labels.

Interpretation and Insights

- Clustering generally grouped passengers based on similar demographic and ticket/fare characteristics, as seen in PCA plots.
- The clusters partially aligned with survival, but not as strongly as supervised models. For instance, in the k=2 setup, many actual survivors were placed in the wrong cluster.
- k=2 or 3 offers the most interpretable unsupervised grouping. Beyond that, clusters become hard to associate with survival outcome.

Here is our statistical summary table

	Precision	Recall	F-1	k	Samples	Accuracy
0	0.81	0.72	0.76	2	608	0.69
1	0.51	0.62	0.56	2	279	0.69
0	0.74	0.65	0.63	3	552/608	0.62
1	0.39	0.49	0.37	3	227/279	0.56
0	0.54	0.57	0.39	4	429/608	0.51
1	0.63	0.58	0.58	4	266/279	0.66

Table 6 k-Means statistical summary table

Note: for k > 2 values are scaled based on how many are missing from samples in wrong classes.

V. CONCLUSION

In this project, we explored a range of supervised learning models to predict passenger survival on the Titanic, including Logistic Regression, Random Forests, Support Vector Machines, and k-Nearest Neighbors (k-NN), as well as experimented with using k-Means. Our objective was not only to achieve predictive accuracy but also to understand how different features contributed to survival outcomes.

The most performant machine learning model observed was the k-NN with cross-validation with an AUC of 0.92, which means that it can classify passenger survival with 92% accuracy, and has an accuracy of ~82% (0.82). The two runner up ones SVM and Random forests also performed admirably with an AUC of 0.88 for SVM and 0.87 for Random forests, also with an accuracy of ~82% (0.82). Therefore for this challenge k-NN with cross validation is the best.

What surprised us was that k-NN in it's simple non cross validated form while having an AUC of just 0.80 was still able to be almost on par in terms of precision, recall, and accuracy with all the other supervised models that were not linear regression with an accuracy of ~82% (0.82). This means that the more complex a model is doesn't mean that it would perform better significantly.

In terms of real world applications, this does show that even simple machine learning models can be 'good enough' in terms of performance, and that we can chalk up most of the performance of the models that we observed really just comes down to the quality of the training data preprocessing, as most of the supervised models hovered around an 82% (0.82) accuracy.

A. Future Work

With our experiment the k-means on part of the dataset, when you let the unsupervised model run loose, it does do better than the average 62% baseline with an accuracy of 69% (0.69) for k = 2 in binary classification. Because k-means managed to classify the two classes better than the baseline which holds exciting prospects for the future. If we were to do this project again this avenue with some more experiments could hold promise in binary classification, and potentially if enough time and effort is dedicated could have a unsupervised model perform on par or surpass supervised learning.

REFERENCES

[1] W. Cukierski, "Titanic - machine learning from disaster," Kaggle, https://www.kaggle.com/competitions/titanic (accessed Apr. 6, 2025).

[2] A. Bhardwaj, "Kaggle's Titanic Challenge for Absolute Beginners," Medium, https://medium.com/@anjalibhardwaj2700/kaggles-titanic-challenge-for-absolute-beginners-5458053bf86e (accessed Apr. 6, 2025).

[3] Wang. Boyu. (2025). Regression [Powerpoint Slides]. Available: https://westernu.brightspace.com/content/enforced/69961-COMPSCI4442B001LEC2FW24/Slides%20-%20Part%201/Lecture%203_Regression.pdf

[4] X. Hu, J. Hu, and M. Hou, "A two-step machine learning method for casualty prediction under emergencies," Journal of Safety Science and Resilience, vol. 3, no. 3, pp. 243–251, Sep. 2022. doi:10.1016/j.jnlssr.2022.03.001

[5] Z. Fang et al., "An earthquake casualty prediction method considering burial and rescue," Safety Science, vol. 126, pp. 1–8, Feb. 2020. doi:10.1016/j.ssci.2020.104670

[6] "What is Random Forest?," IBM, https://www.ibm.com/think/topics/random-forest (accessed Apr. 6, 2025).