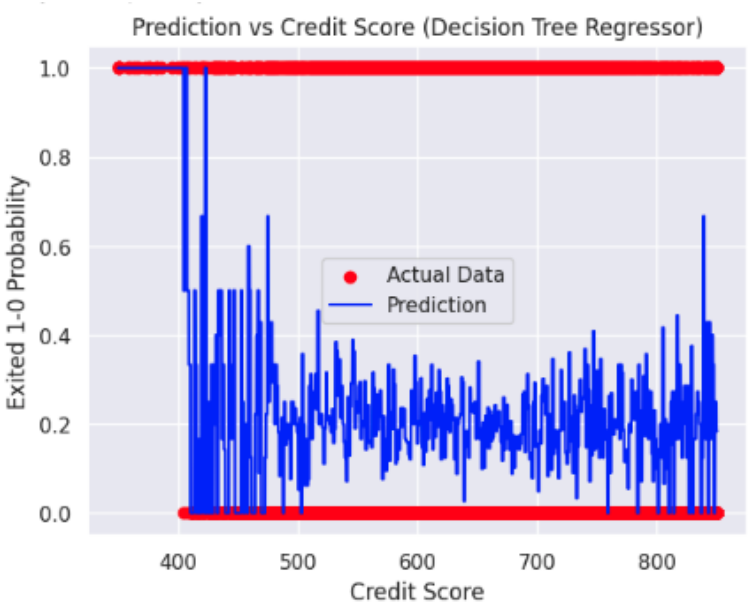
Classification Models on Provided Data

The provided dataset was the Churn_Modelling.csv

Decision Tree Regressor for Continuous Data on Provided Dataset

The Decision Tree Regressor for Churn_Modelling was applied to predict a person's exit probability based on features such as CreditScore and Balance.

- Steps and Results
  - Selected both ‘CreditScore’ and ‘Balance’ as the predictor variables with ‘Exited’ as the target.
  - Trained a ‘DecisionTreeRegressor’ to predict the exit status
  - Plotted ‘CreditScore’ against ‘Exited’ and kept other factors constant.

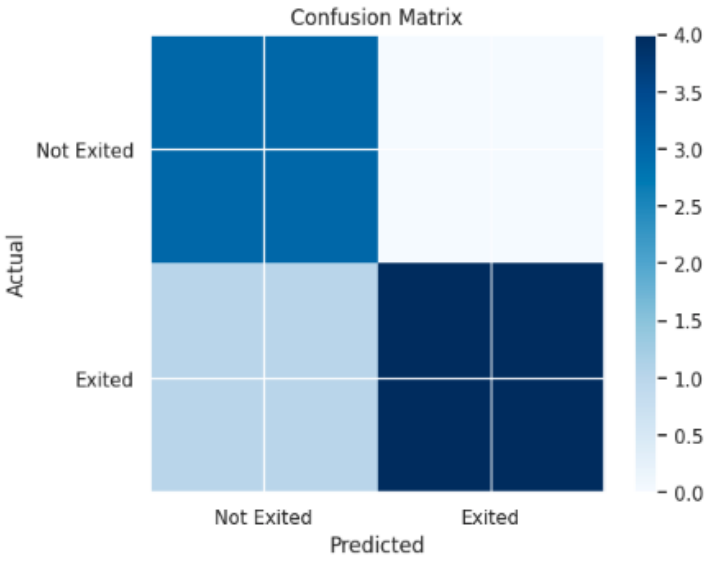


- The plot shows a weak relationship between credit score alone and exit probability. Meaning that the model has trouble determining between customers that are likely to stay or leave based on their credit score alone. Since the ‘Exited’ column is composed of either a one or zero it is harder to predict.

Decision Tree Classifier for Categorical Data on Provided Dataset

- The decision tree classifier was used to classify customers as ‘exited’ or ‘not exited’ using categorical features.
  - Steps & Results
    - Selected ‘CreditScore’, ‘Age’, ‘Balance’ and “NumOfProducts’ as features and ‘Exited” as the target again.
    - Trained a DecisionTreeClassifier with entropy as the splitting criterion.
    - Predicted the exit status for new customer data and analyzed the accuracy.

- Performance
  - A confusion matrix displayed the model accuracy with metrics such as precision, recall, and F1 score.
  - The classification determined that it is very precise in predicting customers who will exit with a precision of 1.00. However, the model missed one exiting customer, recall for exited of 0.80 meaning it was a bit less accurate in catching all exiting cases.




```
Accuracy on test data is 0.88
F1 score on test data is 0.89
Precision Score on test data is 1.00
Recall score on test data is 0.80

Classification Report:
              precision    recall  f1-score   support

  Not Exited       0.75      1.00      0.86         3
      Exited       1.00      0.80      0.89         5

    accuracy                           0.88         8
   macro avg       0.88      0.90      0.87         8
weighted avg       0.91      0.88      0.88         8
```

Feature Section and Data Splitting (Cross-Validation) on Provided Dataset

- Feature Selection
  - Selected ‘CreditScore’, ‘Age’. ‘Balance’ ‘NmOfProducts’, and a binary encoded gender as the predictor variable.
  - It ensured clean and relevant features for model accuracy.
- Data Splitting
  - Applied a 70-30 train-test split and performed cross-validation to avoid overfitting and ensuring consistent data processing.
    - This ensures that the data is ready for model training and testing.

Running Classification Models on Provided Dataset

- Logistic Regression:
  - Accuracy: Good accuracy, due to adjusting the C parameter to find the best fit.
  - Result: Easy to understand the impact of each feature on the exit prediction
- Naive Baynes:
  - Accuracy: Decent accuracy but not the highest.

    - Result: Reasonable accuracy but limited by the assumption of feature independence
- Support Vector Machine SVM
    - Accuracy: Strong accuracy.
    - Result: Adjusting the ‘C’ value made it simple at finding the line between customers who stay and those who leave
- K-Nearest Neighbor (KNN)
    - Accuracy: Moderate accuracy.
    - Result: Accuracy not as high as SVM or ANN.
- Artificial Neural Network (ANN)
    - Accuracy: High accuracy.
    - Results: Captured more complex patterns in the data making it the best performing model in terms of predicting accuracy.

Outcome:

The Artificial Neural Network (ANN) and Support Vector Machine (SVM) with RBF kernel outperformed the other models in terms of accuracy.

Classification Models on Our Dataset

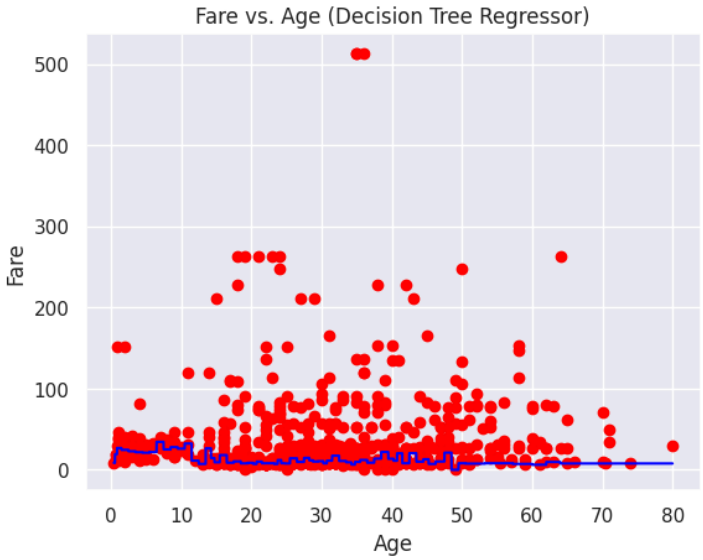
Our dataset was the titanic_trains.csv & the titanic_test.csv

Decision Tree Regressor for Continuous Data

The Decision Tree Regressor was applied to predict the ‘fare’ based on continuous features, specifically ‘pclass’ and ‘age’

- Steps & Results
    - Selected ‘pclass’ and ‘age’ as predictor variables and ‘fare’ as the target.
    - Trained a DecisionTreeRegressor on the training dataset to predict ‘fare’.
    - Visualized predictions by plotting ‘age’ against ‘fare’, using constant values for ‘pclass’.

The plot showed the actual fare versus the age values, indicating a weak relationship, therefore age does not have a strong influence on fare, as the Decision Tree Regressor line appears mostly flat and close to zero, and remains relatively constant across all age values.

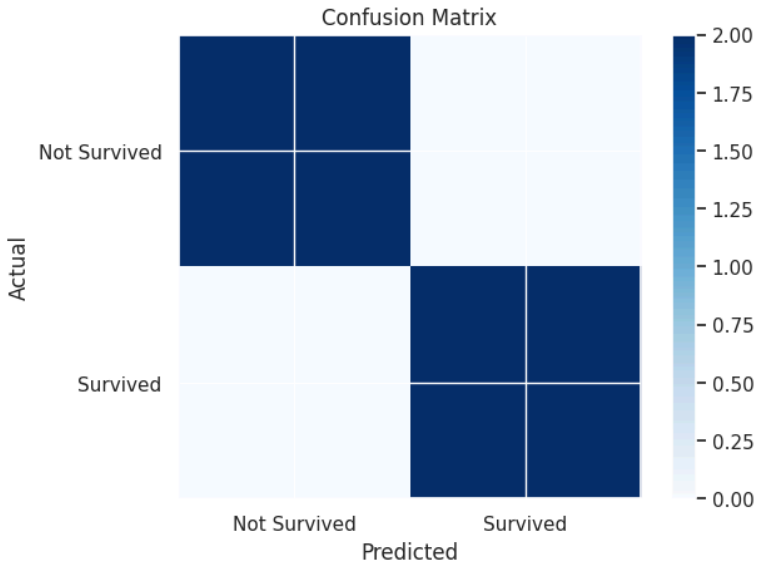


Decision Tree Classifier for Categorical Data

The Decision Tree Classifier was used to classify passengers as either ‘survived’ or ‘not survived’. We worked to predict categorical outcomes based on selected passenger attributes.

- Steps & Results
    - Selected ‘pclass’, ‘age’, ‘fare’, and ‘sibSp’ as features and ‘survived’ as the target.
    - Trained a DecisionTreeClassifier with entropy as the splitting criterion and a maximum tree depth of 3 to avoid overfitting.
    - Predicted survival status for new passenger data and analyzed the accuracy.
- Performance
    - Used confusion matrix to visually display the model accuracy through metrics of: precision, recall, and F1 score.

    - The ‘not survived’ class was 1.0, indicating the model was highly effective at identifying passengers who did not survive



Predicted

```
Accuracy on test data is 1.00
F1 score on test data is 1.00
Precision Score on test data is 1.00
Recall score on test data is 1.00
              precision    recall  f1-score   support

Not Survived       1.00      1.00      1.00         2
    Survived       1.00      1.00      1.00         2

    accuracy                           1.00         4
   macro avg       1.00      1.00      1.00         4
weighted avg       1.00      1.00      1.00         4
```

All these metrics are 1.00 for both classes ("Not Survived" and "Survived"), meaning the model has perfect precision (no false positives), recall (no false negatives), and F1-score (a balance of precision and recall).

Feature Section and Data Splitting (Cross-Validation)

- Feature Selection
    - We choose ‘pclass’, ‘age’, ‘fare’, ‘sibsp’, and ‘sex’ (encoded as binary) as variables
        - This ensures for relevant cleaned out featured data
- Data Splitting
    - Applied a 70-30 train-test split and performed cross-validation to avoid overfitting and ensuring consistent data processing.
        - This ensures that the data is ready for model training and testing.

Running Classification Models

- We trained each classification model using selected features and validate on the test set
- Training Process:

- We utilized classification techniques like decision tree, logistic regression, naive bayes, support vector machine (SVM), K-nearest neighbor (KNN), and artificial neural network (ANN)
    - Decision Tree: Used entropy as the criterion and limited tree depth to prevent overfitting.
    - Logistic Regression: Tuned C values to identify the best fit.
        - Result: Provided good accuracy and interpretability, as it's a linear model.
    - Naive Bayes: Implemented Gaussian Naive Bayes, considering the continuous nature of features like 'fare' and 'age'.
        - Result: Reasonable accuracy but limited by the assumption of feature independence (not fully valid in the dataset for Titanic)
    - Support Vector Machine (SVM): Tested both linear and RBF kernels, adjusting C parameter to optimize performance.
        - Result: Performed well
    - K-nearest neighbor (KNN): Experimented with different values of K.
        - Result: Moderately well
    - Artificial Neural Network (ANN): Built and trained a neural network.
        - Result: Achieved high accuracy after 100 epochs
- Outcome
    - The ANN and SVM models with an RBF kernel outperformed other models in terms of accuracy

Team Contributions:
Both team members, Marilyn Sarabia Ortiz & Isabel Santoyo-Garcia, contributed equally to all aspects of this project, including data preprocessing, model development, analysis, and documentation.