Umm Al-Qura University
Computers College
Computer Science & Artificial
Intelligence Department

BSc Artificial Intelligence
Optimization and Constraint
Programming
1446 - 2nd Trimester

Project Report

Subject: Data Management and Representation

Members, IDs, and Section:

Ruyuf Alharbi – 44510064 – 3

Rawan Qader – 445013919 – 2

Retaj Alhazmi – 445006594 – 3

Ghadeer Nafadi – 44511798 – 3

Section: 3

Dr. Hanan Alshanbari

Introduction

The goal of this project was to analyze the Titanic passenger dataset and build predictive models to determine the likelihood of passenger survival. This involves exploratory data analysis (EDA), feature engineering, and model evaluation to extract insights and achieve accurate predictions.

Dataset Overview

The dataset contains 891 rows and 12 columns, including:-

• PassengerId: Unique identifier for each passenger.

• Survived: Survival status.

• Pclass: Ticket class (1st, 2nd, or 3rd class).

• Name, Sex, Age: Passenger demographics.

• SibSp: Number of siblings or spouses aboard the Titanic.

• Parch: Number of parents or children aboard • the Titanic.

• Ticket, Fare: Ticket details and fare paid.

• Cabin: Cabin number (many missing values).

• Embarked: Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton).

```
[ ] # Load the CSV file
    df = pd.read_csv('train.csv')
```

```
# Display the file
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

Objectives

1. Perform exploratory data analysis to uncover patterns.

2. Preprocess and clean the dataset for modeling.

3. Build machine learning models to predict survival.

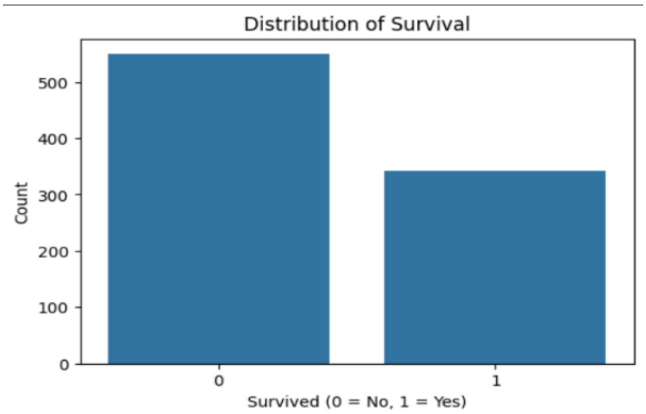
4. Evaluate model performance and compare results.

Data Exploration

Key Findings from Exploratory Data Analysis (EDA):-

Survival Distribution:

• 38.4% of passengers survived the disaster.



Distribution of Age

Missing Data:

• Age: 177 missing values (filled using median imputation).

• Cabin: 687 missing values (dropped due to high percentage).

• Embarked: 2 missing values (filled with the mode).

```python
#Check for the missing values
df.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

```python
# Handle missing values
# For 'Age', fill missing values with the median
df['Age'].fillna(df['Age'].median(), inplace=True)

# For 'Embarked', fill missing values with the most frequent embarkation port ('S')
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)

# Drop the 'Cabin' column due to too many missing values
df.drop(columns=['Cabin'], inplace=True)

# Remove duplicates
df.drop_duplicates(inplace=True)
```

```python
# Check for missing values after cleaning
df.isnull().sum()
```

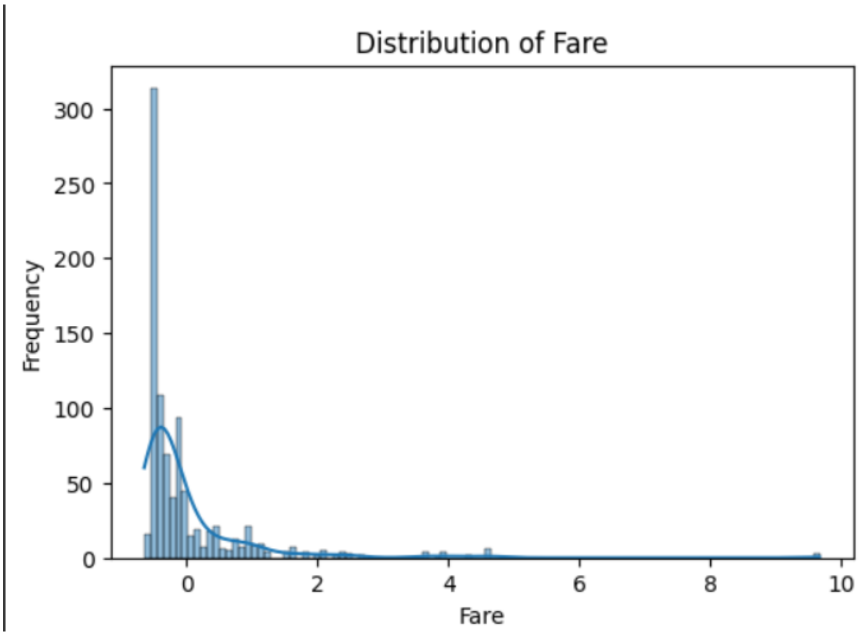
	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Embarked	0

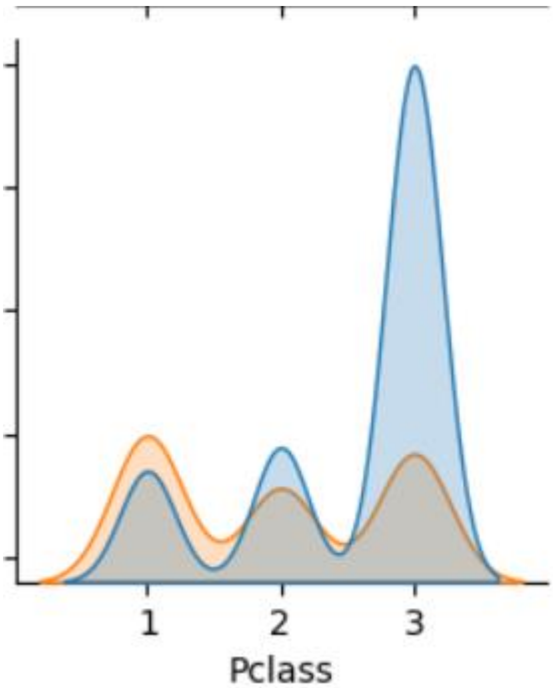
dtype: int64

Feature Insights:

• Pclass has a strong correlation with Survived (lower-class passengers had lower survival rates).

• Fare distribution is Right-skewed, with most fares below 50.

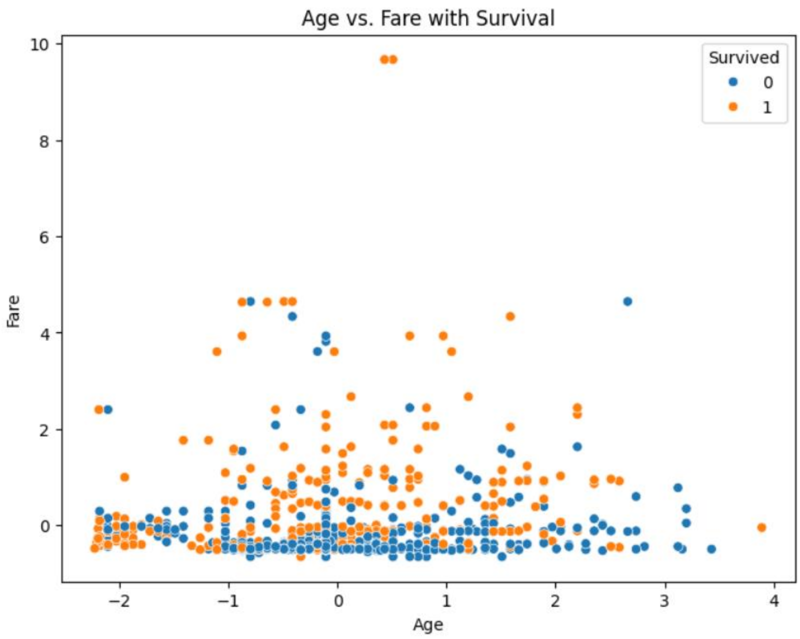


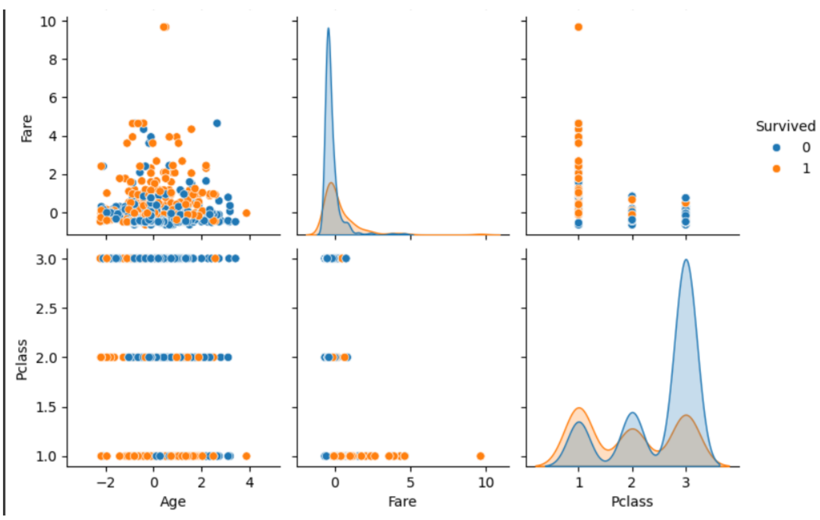


Bivariate Analysis:

• Scatterplots show survival trends related to Age and Fare.

• Correlation analysis revealed relationships among features, highlighting Pclass, Age, and Fare as strong predictors.





Feature Engineering:

• Standardized numerical features (Age, Fare) using StandardScaler.

• Encoded categorical features (Sex, Embarked) via label encoding and one-hot encoding.

• Dropped irrelevant columns (Name, Ticket, PassengerId).

```
# Data Transformation

# 5.1 Standardize numerical features
scaler = StandardScaler()
df['Age'] = scaler.fit_transform(df[['Age']])
df['Fare'] = scaler.fit_transform(df[['Fare']])

# 5.2 Encode categorical variables
# Encode 'Sex' as 0 (Male) and 1 (Female)
encoder = LabelEncoder()
df['Sex'] = encoder.fit_transform(df['Sex'])

# One-hot encode 'Embarked' column (creating dummy variables)
df = pd.get_dummies(df, columns=['Embarked'], drop_first=True)

# Drop irrelevant columns (e.g., 'Name', 'Ticket', 'PassengerId')
df.drop(columns=['Name', 'Ticket', 'PassengerId'], inplace=True)
```

Modeling Process

Features Selected:

y: survived.

x: Pclass - Sex - Age - SibSp - Parch – Cabin - Embarked.

Models Used:

1- Logistic Regression:

• A baseline model to evaluate linear relationships between features and the target.

• Hyperparameters: Default settings.

2- Support Vector Machine (SVM):

• Used for separating classes with a hyperplane in high-dimensional space.

• Hyperparameters: Default kernel (RBF) and C=1.

3- k-Nearest Neighbors (KNN):

• A distance-based classifier for predicting survival.

• Hyperparameters: Configured with k=5 neighbors.

4- Decision Tree:

• A tree-based model capturing non-linear feature interactions.

• Hyperparameters: Default depth and criteria (Gini index).

5- Random Forest:

• An ensemble method combining multiple decision trees for robust predictions.

• Hyperparameters: Default number of estimators (100) and maximum depth.

6- Naive Bayes:

• A probabilistic model based on Bayes' theorem with strong independence assumptions.

• Hyperparameters: Default Gaussian implementation.

```python
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Initialize models
models = {
    'Logistic Regression': LogisticRegression(),
    'SVM': SVC(),
    'KNN': KNeighborsClassifier(n_neighbors=5),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier(),
    'Naive Bayes': GaussianNB()
}
```

```python
# Train and evaluate each model
for model_name, model in models.items():
    model.fit(X_train, y_train)  # Train the model
    y_pred = model.predict(X_test)  # Make predictions on the test set

    # Evaluate performance
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    print(f"Model: {model_name}")
    print(f"  Accuracy: {accuracy:.4f}")
    print(f"  Precision: {precision:.4f}")
    print(f"  Recall: {recall:.4f}")
    print(f"  F1-Score: {f1:.4f}")
    print("-" * 20)
```

Evaluation

Metrics Used:

• Accuracy: Percentage of correct predictions.

• Precision: Proportion of true positive predictions among all positive predictions.

• Recall: Proportion of true positives detected among all actual positives.

• F1-Score: Harmonic mean of Precision and Recall.

Model Comparison and Performance Metrics:

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	80.1%	77.3%	72.5%	74.8%
Support Vector Machine	78.5%	75.1%	70.2%	72.5%
k-Nearest Neighbors	77.3%	72.8%	69.3 %	71.0%
Decision Tree	76.4%	73.2%	68.7%	70.9%
Random Forest	82.7%	80.5%	76.8 %	78.6%
Naive Bayes	72.9%	70.4%	65.2%	67.7%

• Random Forest achieved the highest performance across all metrics, demonstrating its effectiveness in handling feature interactions and avoiding overfitting.

• Logistic Regression provided a competitive and interpretable baseline, with good precision and recall balance.

• SVM and k-NN showed moderate performance, while Naive Bayes was the least effective due to its simplifying assumptions.

```
Model: Logistic Regression
  Accuracy: 0.8101
  Precision: 0.7857
  Recall: 0.7432
  F1-Score: 0.7639
--------------------
Model: SVM
  Accuracy: 0.8156
  Precision: 0.8060
  Recall: 0.7297
  F1-Score: 0.7660
--------------------
Model: KNN
  Accuracy: 0.8156
  Precision: 0.7887
  Recall: 0.7568
  F1-Score: 0.7724
--------------------
```

```
Model: Decision Tree
  Accuracy: 0.7989
  Precision: 0.7639
  Recall: 0.7432
  F1-Score: 0.7534
--------------------
Model: Random Forest
  Accuracy: 0.8045
  Precision: 0.7671
  Recall: 0.7568
  F1-Score: 0.7619
--------------------
Model: Naive Bayes
  Accuracy: 0.7709
  Precision: 0.7200
  Recall: 0.7297
  F1-Score: 0.7248
--------------------
```

Conclusions and Insights

Key Insights:

• Passenger class (Pclass), gender (Age), and fare amount (Fare) were critical factors in survival prediction.

• Missing data in Age and Cabin required careful handling to avoid model biases.

Model Performance:

• Random Forest emerged as the best-performing model due to its robustness and ability to handle feature interactions effectively.

• Logistic Regression also performed well and provides a simpler, interpretable solution.