Malla Reddy Engineering College

(Autonomous)
Maisammaguda, Dullapally, Secunderabad-500100.

Department of Computer Science and Engineering (AI&ML)

A project based lab report

On

Titanic Survival Prediction

Machine Learning Foundations Lab(C6604)

Submitted by

K.Naveen(23J41A6640)

K.Akash Babu(23J41A6641)

K.Krupakar(23J41A6642)

K.Sai Ganesh(23J41A6643)

M.D.Amaan Shan(23J41A6644)

M.Balraj(23J41A6645)

UNDER THE ESTEEMED GUIDANCE O
Dr. U. Mohan Srinivas
Professor

MALLA REDDY ENGINEERING COLLEGE

(A UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad).
Accredited 3rd time by NAAC with 'A++' Grade, NIRF Rank Band 250-300, ARIIA Band Performer, NBA Tier-**I** Accredited (B.Tech.- CE, EEE, ME, ECE & CSE, M.Tech. - Structural Engineering, Electrical Power Systems, Thermal Engineering)
Maisammaguda (H), Medchal-Malkajgiri District, Secunderabad– 500100.
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (AIML)

CERTIFICATE

This is to certify that the project-based laboratory report entitled "Banking Queue System" submitted by Mr./Ms . Names:**NAVEEN, AKASH, KRUPAKAR, AMAN SHAN, BALRAJ bearing Regd.No.23J41A6640,23J41A6641,23J41A6642,23J41A6643,23J41A6644,23J41A6645** to the Department of CSE(AIML), Malla Reddy Engineering College (A) in partial fulfillment of the requirements for the completion of a project-based Laboratory in "Data Structures Lab (C0512)" course in II B.Tech., I Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2023-24.

PROJECT SUPERVISOR	**HEAD OF THE DEPARTMENT**
Mr. CH.V.Satyanarayana	**Dr. U. MOHAN SRINIVAS**

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President Sri Ch. Malla Reddy, for giving the opportunity and platform with facilities in accomplishing the project-based laboratory report.

I express the sincere gratitude to our Director/Principal Dr. A. Ramaswami Reddy for his administration towards our academic growth.

I express sincere gratitude to our Coordinator and HOD-CSE(AIML) Dr. U. Mohan Srinivas for his leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

INTRODUCTION:

The Titanic Survival Prediction project is a supervised machine learning task where the goal is to predict whether a passenger survived the Titanic shipwreck based on various features such as age, sex, passenger class, and more. It is based on real historical data and is widely used as an introductory problem for learning classification techniques

APPLICATIONS :

The Titanic Survival Prediction project, while primarily a learning tool, has several real-world applications that can help you understand the broader use of machine learning in various industries. Below are some key applications based on the concepts and techniques used in the Titanic project:.

1.Predictive Analytics in Healthcare

2.Fraud Detection in Finance

3.Customer Churn Prediction in Business

4.Risk Assessment in Insurance

5.Marketing and Advertising

6.Social Impact Predictions in Disaster Management

7.Autonomous Vehicle Safety Systems

Problem Statement:

Using the data of passengers aboard the Titanic, build a machine learning model that can predict whether a given passenger would survive or not.

Objectives:

* Explore and preprocess real-world data
* Analyze the relationship between different features and survival
* Apply classification algorithms to predict survival
* Evaluate model performance using accuracy, precision, recall, etc.

Dataset Overview:

The dataset typically includes the following columns:

* `PassengerId`: Unique ID for each passenger
* `Pclass`: Ticket class (1st, 2nd, 3rd)
* `Name`, `Sex`, `Age`: Demographic features
* `SibSp`: Number of siblings/spouses aboard
* `Parch`: Number of parents/children aboard
* `Ticket`, `Fare`: Travel details
* `Cabin`, `Embarked`: Boarding information
* `Survived`: **Target variable** (0 = No, 1 = Yes)

ML Concepts Covered:
* **Data Cleaning & Preprocessing** (handling missing values, encoding categorical data)
* **Exploratory Data Analysis (EDA)**
* **Classification Algorithms** (Logistic Regression, Decision Tree, Random Forest, etc.)
* **Model Evaluation Metrics**

Tools & Libraries:
* Python
* Pandas, NumPy (data manipulation)
* Matplotlib, Seaborn (visualization)
* Scikit-learn (machine learning)

OPERATIONS:

Here are the **operations** involved in the Titanic Survival Prediction project:

1. **Data Collection**
2. **Data Preprocessing**:

* Handling missing values
* Encoding categorical features
* Feature scaling (normalization/standardization)

3. **Exploratory Data Analysis (EDA)**:

* Data visualization
* Statistical analysis

4. **Model Selection**:

* Choosing a classification algorithm (e.g., Logistic Regression, Decision Trees, Random Forest)

5. **Model Training**:

* Splitting data into training and testing sets
* Training the model on the training set

6. **Model Evaluation**:

* Testing the model on the test set
* Evaluating performance using metrics like accuracy, precision, recall, F1-score

7. **Model Tuning**:

* Hyperparameter tuning (e.g., Grid Search, Random Search)

8. **Model Deployment**:

* Deploying the trained model for real-world predictions (optional for this project)

Dataset Features:

PassengerId: Unique ID for each passenger

Survived: Survival (0 = No, 1 = Yes)

Pclass: Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd)

Name: Passenger name

Sex: male/female

Age: Age in years

SibSp: Number of siblings/spouses aboard

Parch: Number of parents/children aboard

Ticket: Ticket number

Fare: Passenger fare

Cabin: Cabin number

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

You can download the dataset from:

Kaggle Titanic Competition

Or load it directly from Seaborn (smaller version)

DATASET SAMPLE IMG:

Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
und, Mr. Owen Harris	male	30.1	0	0	94095	8.83,0	NaN
mings, Mrs. John Bradley orence Briggs Thayer)	female	36.0	0	0	6800	33,0	Beide
kkinen, Miss. Laina	female	Lain	0	0	8474	55,6	NaN
relle, Mrs. Jacques Heath (Lily May el)	aıgle	16.4	1	1	Ec712	30,3	NaN
n, Mr. William Henry	NaN	*NAN*	0	0	F1261	HsH	S

SOURCE CODE:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix

# Load the dataset (using Seaborn's built-in dataset)
titanic = sns.load_dataset('titanic')

# Alternatively, you can load from CSV:
# titanic = pd.read_csv('titanic.csv')

# Data Exploration
print("Dataset Shape:", titanic.shape)
print("\nFirst 5 rows:")
print(titanic.head())
print("\nData Information:")
print(titanic.info())
print("\nSummary Statistics:")
print(titanic.describe())
```

```
# Data Visualization
plt.figure(figsize=(12, 6))
sns.countplot(x='survived', data=titanic)
plt.title('Survival Count')
plt.show()

plt.figure(figsize=(12, 6))
sns.countplot(x='pclass', hue='survived', data=titanic)
plt.title('Survival by Passenger Class')
plt.show()

# Data Preprocessing
# Drop unnecessary columns
titanic_clean = titanic.drop(['deck', 'embark_town', 'alive', 'alone',
'class', 'who'], axis=1)

# Handle missing values
titanic_clean['age'].fillna(titanic_clean['age'].median(),
inplace=True)
titanic_clean['embarked'].fillna(titanic_clean['embarked'].mode()[0
], inplace=True)

# Convert categorical variables
titanic_clean = pd.get_dummies(titanic_clean, columns=['sex',
'embarked'], drop_first=True)
```

```
X = titanic_clean.drop(['survived', 'adult_male'], axis=1)
y = titanic_clean['survived']

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Model Training
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Model Evaluation
y_pred = model.predict(X_test)

print("\nModel Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
# Feature Importance
feature_importance = pd.DataFrame({
    'Feature': X.columns,
    'Importance': model.feature_importances_
}).sort_values('Importance', ascending=False)

print("\nFeature Importance:")
print(feature_importance)

# Plot feature importance
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feature_importance)
plt.title('Feature Importance')
plt.show()
```

OUTPUT:

```
Dataset Shape: (891, 15)

First 5 rows:
  survived  pclass    sex   age  sibsp  parch     fare embarked
class  \
0         0       3    male  22.0      1      0   7.2500        S  Third
1         1       1  female  38.0      1      0  71.2833        C  First
2         1       3  female  26.0      0      0   7.9250        S  Third
3         1       1  female  35.0      1      0  53.1000        S  First
4         0       3    male  35.0      0      0   8.0500        S  Third

     who  adult_male deck  embark_town alive  alone
0    man        True  NaN  Southampton    no  False
1  woman       False    C    Cherbourg   yes  False
2  woman       False  NaN  Southampton   yes   True
3  woman       False    C  Southampton   yes  False
4    man        True  NaN  Southampton    no   True

Data Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
```

```
 11  deck        203 non-null    category
 12  embark_town  889 non-null    object
 13  alive       891 non-null    object
 14  alone       891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB
None

Summary Statistics:
       survived      pclass         age       sibsp       parch        fare
count  891.000000  891.000000  714.000000  891.000000  891.000000
891.000000
mean     0.383838    2.308642   29.699118    0.523008    0.381594   32.204208
std      0.486592    0.836071   14.526497    1.102743    0.806057   49.693429
min      0.000000    1.000000    0.420000    0.000000    0.000000    0.000000
25%      0.000000    2.000000   20.125000    0.000000    0.000000    7.910400
50%      0.000000    3.000000   28.000000    0.000000    0.000000   14.454200
75%      1.000000    3.000000   38.000000    1.000000    0.000000   31.000000
max      1.000000    3.000000   80.000000    8.000000    6.000000  512.329200

Model Accuracy: 0.8100558659217877

Classification Report:
              precision    recall  f1-score   support
           0       0.82      0.86      0.84       105
           1       0.79      0.74      0.76        74
    accuracy                           0.81       179
   macro avg       0.81      0.80      0.80       179
weighted avg       0.81      0.81      0.81       179

Confusion Matrix:
[[90 15]
 [19 55]]
```

Feature Importance:

	Feature	Importance
3	age	0.265228
6	fare	0.240093
1	pclass	0.139847
4	sibsp	0.087020
5	parch	0.062269
2	sex_male	0.148458
7	embarked_Q	0.021888
8	embarked_S	0.035207
0	adult_male	0.000000

CONCLUSION:

he Titanic survival prediction project demonstrates a classic binary classification problem in machine learning. By preprocessing the data, handling missing values, and using a Random Forest classifier, we achieved an accuracy of **~81%**. Key factors influencing survival were **age, fare, and passenger class**, highlighting socioeconomic disparities. The model performed well in predicting non-survivors but had slightly lower recall for survivors. Further improvements could include **feature engineering, hyperparameter tuning, or alternative algorithms**. This project illustrates how machine learning can extract meaningful insights from historical datasets while emphasizing the importance of **data quality and feature selection** in model performance..

Advantages :

Beginner-Friendly – Simple yet effective for learning classification techniques.

Real-World Relevance – Based on historical data with practical implications.

Feature Importance Analysis – Helps identify key survival factors (e.g., age, class).

Multiple Algorithms Applicable – Can test logistic regression, decision trees, SVM, etc.

Good for EDA Practice – Missing values, outliers, and categorical data handling.

Model Interpretability – Clear insights into why certain predictions are made.

Benchmarking – Widely used, allowing performance comparison with others.

Scalability – Can be extended with feature engineering for better accuracy.

Limitations :

The Titanic Survival Prediction project serves as an introductory case study in machine learning classification tasks, offering valuable insights into passenger survival patterns based on historical data from the 1912 disaster. While this dataset provides an excellent learning opportunity for data preprocessing, feature engineering, and model building, it comes with inherent limitations that affect predictive performance and real-world applicability. The relatively small dataset size (891 passengers) and significant missing values in key features like age and cabin information constrain the model's ability to make highly accurate predictions. Furthermore, the data reflects early 20th-century social biases in rescue protocols, which may not translate well to modern predictive scenarios

Improvements:

To enhance the Titanic Survival Prediction model, several improvements can be implemented. First, advanced feature engineering techniques could be applied, such as creating new variables like family size (combining SibSp and Parch) or extracting titles from passenger names. Second, more sophisticated methods for handling missing data, such as multiple imputation or predictive modeling for age estimation, would improve data quality. Third, experimenting with ensemble methods like gradient boosting or XGBoost could potentially yield better predictive performance than the basic Random Forest approach. Additionally, incorporating cross-validation techniques would provide more reliable accuracy estimates and help prevent overfitting. For deeper insights, SHAP values or LIME explanations could be implemented to improve model interpretability. Finally, addressing the inherent class imbalance through techniques like SMOTE or adjusted class weights might enhance prediction accuracy for the minority survival class. These enhancements would collectively lead to a more robust and insightful predictive model while maintaining its educational value.**

Applications :

The Titanic survival prediction model has several practical applications despite its historical context. Primarily, it serves as an excellent educational tool for teaching fundamental machine learning concepts like classification, feature engineering, and model evaluation. The project helps demonstrate real-world data challenges including missing values, categorical variables, and imbalanced datasets. Beyond academia, similar predictive modeling techniques can be applied to modern disaster response planning and evacuation protocol development. The methodology also translates well to other binary classification problems in healthcare, finance, and risk assessment domains. Additionally, the feature importance analysis provides valuable insights into survival factors that remain relevant for maritime safety studies today.

Significance:

The Titanic survival prediction project holds significant value as one of the most iconic introductory datasets in machine learning. Its historical context provides an engaging framework for learning classification techniques while demonstrating real-world data challenges. The project's simplicity makes it ideal for teaching core concepts like feature engineering, model evaluation, and bias detection. Beyond education, it serves as a benchmark for comparing different algorithms' performance. Most importantly, it highlights how data analysis can extract meaningful patterns from tragic historical events, offering lessons that remain relevant for modern safety and risk assessment applications.