INTEL UNNATI INDUSTRIAL TRAINING 2024

Problem Statement:

KNOWLEDGE REPRESENTATION AND INSIGHTS GENERATION FROM STRUCTURED DATASETS

TEAM NAME: TECH AVENGERS

GUIDE:

Mr. A. Saleem

Associate Professor

TEAM MEMBERS:

1.PODILAPU KEERTHI (Team Lead)

2.VELIKANTI MYTHILI

3.ARISINAPALLI SUMANTH

4.KELIGE VINAYAK

5.SYED SHOAIBUDDIN

COLLEGE:

Malla Reddy College of Engineering and Technology

1.INTRODUCTION:

- **Problem Statement:**
  Developing an AI-based solution for knowledge representation and insight generation from structured datasets.

- **Objectives:**
  - Analysis and processing of structured data
  - Representing knowledge effectively
  - Generating meaningful insights to aid decision making

In the era of big data, organizations across various sectors are generating massive amounts of data every day. This data, if processed and analyzed correctly, can provide valuable insights that can significantly improve the decision-making process.

However, the challenge lies in effectively representing this knowledge and extracting useful insights from it.

You will be provided with a structured dataset. Your solution should be able to process this dataset, represent the knowledge contained within it effectively, and generate meaningful insights.

The solution should include the following features:

- **Data Pre-processing**:

  The solution should be able to clean and pre-process the dataset to make it suitable for further analysis.

- **Knowledge Representation**:

  The solution should effectively represent the knowledge contained within the dataset. This could be in the form of graphs, charts, or any other visual representation that makes the data easy to understand.

- **Pattern Identification**:

  The solution should be able to identify patterns within the dataset. This could include identifying trends, anomalies, or any other patterns that could provide valuable insights.

- **Insight Generation**:

  Based on the identified patterns, the solution should generate meaningful insights. These insights should be presented in a clear and understandable manner.

- **Scalability**:

  The solution should be scalable. It should be able to handle datasets of varying sizes and complexities.

- **User-friendly Interface**:

  The solution should have a user-friendly interface that allows users to easily interact with it and understand the generated insights

2.DATASET DESCRIPTION:

- **Dataset Source:** Dataset is taken from Kaggle
- **Name of the Dataset:** Titanic-Machine Learning from Disaster
- **Link for the Dataset:** https://www.kaggle.com/competitions/titanic/data

Key features of the dataset: Dataset consists of 891 rows and 12 columns.

1. **PassengerId:** A unique identifier for each passenger.
2. **Survived:** Indicates whether the passenger survived (1) or not (0).
3. **Pclass:** Ticket class (1st, 2nd, or 3rd class).
4. **Name:** Passenger's name.
5. **Sex:** Passenger's gender.
6. **Age:** Passenger's age.
7. **SibSp:** Number of siblings/spouses aboard the Titanic.
8. **Parch:** Number of parents/children aboard the Titanic.
9. **Ticket:** Ticket number.
10. **Fare:** Fare paid for the ticket.
11. **Cabin:** Cabin number where the passenger stayed (if available).
12. **Embarked:** Port of embarkation (C = Cherbourg, Q = Queenstown, S = Southampton).

These columns provide various details about each passenger aboard the Titanic, which can be used to analyze factors influencing survival rates.

Preprocessing steps:

- **Handling Missing Data:** Identifying columns with missing values (NaNs). For numerical data (like Age), replacing missing values with the mean, median, or mode. For categorical data (like Cabin or Embarked), replacing missing values with the most frequent category or create a new category for missing values.

- **Handling Categorical Variables:** Encoding categorical variables into numerical representations suitable for machine learning models: **One-hot encoding:** Creating dummy variables for each category**. Label encoding:** Converting each category into a numerical label.

- **Feature Selection:** Selecting relevant features that contribute most to predicting the target variable (Survived).

- **Data Splitting:** Split the dataset into training and testing sets to evaluate model performance.

- **Feature Scaling:** Scale numerical features to a similar range, which can be important for certain models (e.g., SVM, KNN)

These steps ensure that the data is cleaned, transformed, and prepared in a way that maximizes the effectiveness of machine learning algorithms applied to the Titanic dataset.

3.METHODOLOGY:

Algorithms for Knowledge Representation and techniques for pattern Identification:

Bar plot: It is used to compare categorical data, such as the number of survivors by passenger class or gender etcs.

Histogram: It shows the distribution of a numerical variable, such as the age of passengers.

Scatter plot: It is useful for understanding relationships between two numerical variables.

Count plot: It is useful for visualizing the frequency distribution of categorical variables.

K-Means Clustering: K-Means clustering is used to partition the data into clusters.

DBSCAN Clustering: DBSCAN is used for density-based clustering.

Tools:

- **Data Preprocessing tools:** Pandas, Numpy, Scikit-learn.
- **Knowledge Representation Tools:** Matplotlib, Seaborn.
- **Pattern Identification Tools:** Scikit-learn,
- **User-friendly interface:** Svelte

Technologies used

Machine Learning (ML) :

Python: Utilized for data analysis, model training, and prediction.

Libraries like Pandas for data manipulation, Scikit-learn for machine learning models (e.g., logistic regression), and Matplotlib/Seaborn for data visualization.

Flask: Developed a RESTful API to serve machine learning models. Used to integrate the logistic regression model for predicting heart attack risk based on user inputs.

Colab: Used for exploratory data analysis (EDA) and initial model training. **Colab notebooks** were integrated into the website for transparency and reproducibility of analysis.

Website Building:

Svelte: Frontend framework used for building the interactive user interface (UI) and managing user interactions.

Tailwind CSS: Used for styling and layout of the website. Tailwind's utility first approach facilitated quick prototyping and customization of UI components.

JavaScript: Integrated to handle frontend logic and interactions, ensuring dynamic content updates and seamless user experience

4.RESULTS AND DISCUSSION:

Key Findings from Data analysis:

1.SURVIVAL RATE

Overall Survival Rate: Typically, around 38% of passengers survived.

2.IMPACT OF PASSENGER CLASS (Pclass)

Class and Survival:

First-class passengers had a higher survival rate compared to second and third-class passengers.

1st Class: ~62% survival rate

2nd Class: ~47% survival rate

3rd Class: ~24% survival rate

3.IMPACT OF GENDER

Gender and Survival: Females had a significantly higher survival rate than males.

Female: ~74% survival rate

Male: ~19% survival rate

4.IMPACT OF AGE

Age and Survival: Younger passengers had a higher chance of survival, particularly children.

Children (age < 16): ~55% survival rate

5.IMPACT OF FARE

Fare and Survival: Higher fare-paying passengers were more likely to survive. This correlates with the class of the ticket.

6.IMPACT OF EMBARKATION PORT

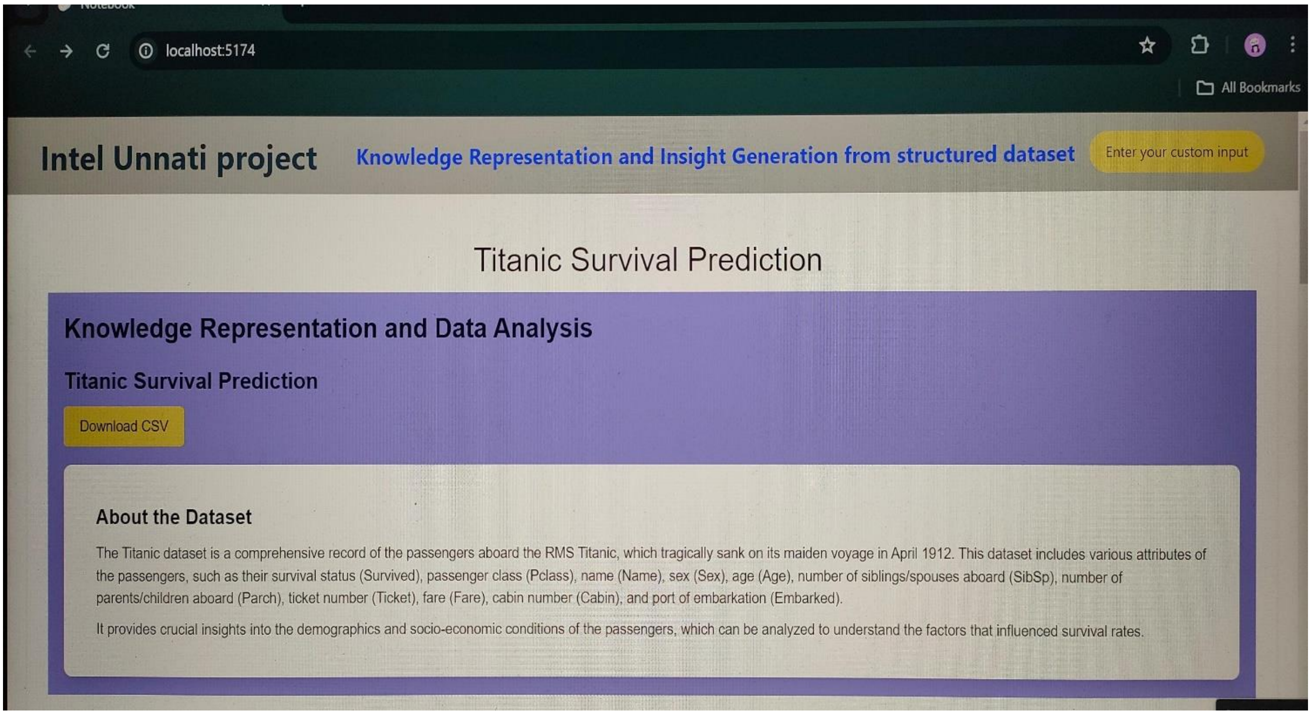
Embarkation Port and Survival: Passengers who embarked at Cherbourg (C) had a higher survival rate compared to those who embarked at Queenstown (Q) and Southampton (S).

Cherbourg: ~55% survival rate

Queenstown: ~39% survival rate

Southampton: ~34% survival rate

➢ VISUALIZATIONS:



Intel Unnati project

Knowledge Representation and Insight Generation from structured dataset

Titanic Survival Prediction

Knowledge Representation and Data Analysis

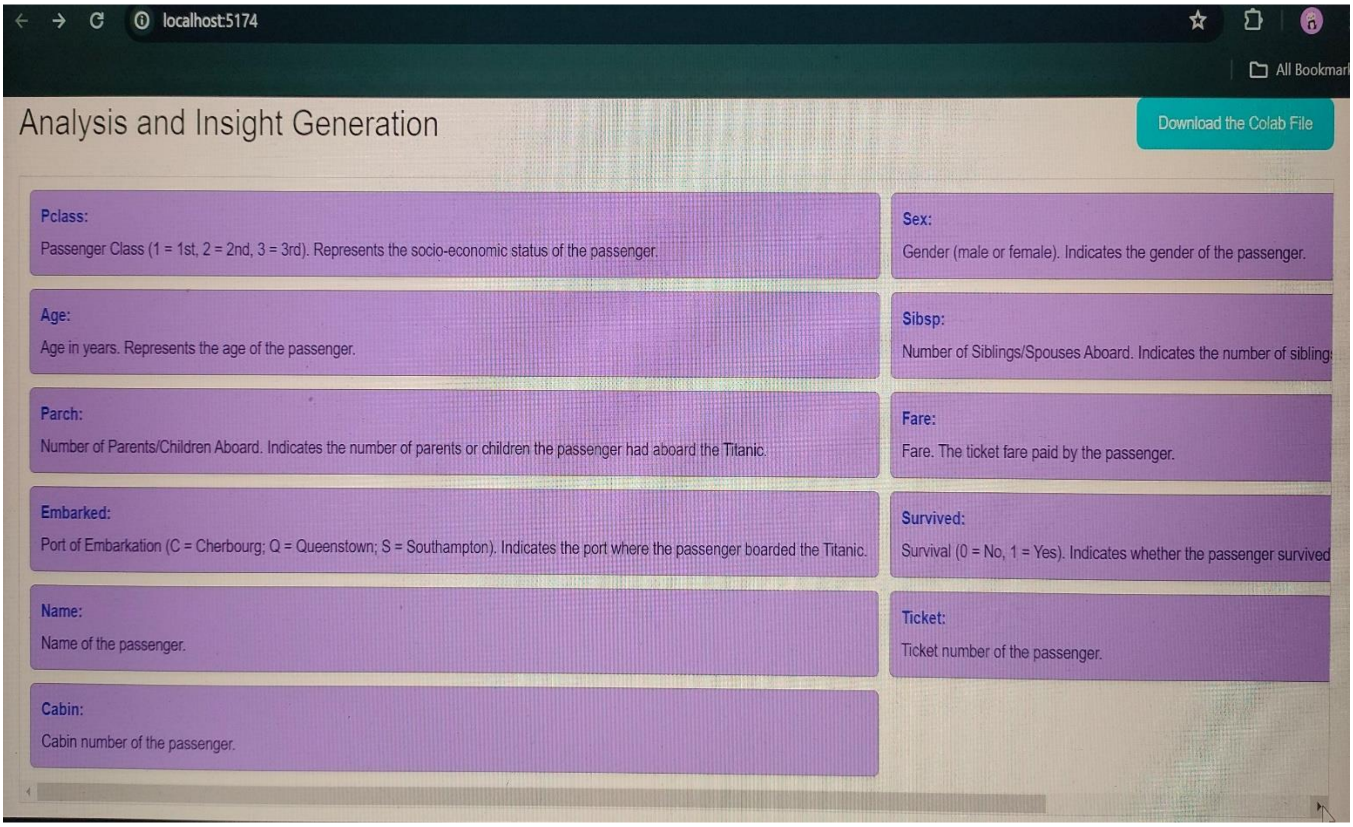
Titanic Survival Prediction

About the Dataset

The Titanic dataset is a comprehensive record of the passengers aboard the RMS Titanic, which tragically sank on its maiden voyage in April 1912. This dataset includes various attributes of the passengers, such as their survival status (Survived), passenger class (Pclass), name (Name), sex (Sex), age (Age), number of siblings/spouses aboard (SibSp), number of parents/children aboard (Parch), ticket number (Ticket), fare (Fare), cabin number (Cabin), and port of embarkation (Embarked).

It provides crucial insights into the demographics and socio-economic conditions of the passengers, which can be analyzed to understand the factors that influenced survival rates.

This is the output screen



Analysis and Insight Generation

Pclass:
Passenger Class (1 = 1st, 2 = 2nd, 3 = 3rd). Represents the socio-economic status of the passenger.

Sex:
Gender (male or female). Indicates the gender of the passenger.

Age:
Age in years. Represents the age of the passenger.

Sibsp:
Number of Siblings/Spouses Aboard. Indicates the number of sibling

Parch:
Number of Parents/Children Aboard. Indicates the number of parents or children the passenger had aboard the Titanic.

Fare:
Fare. The ticket fare paid by the passenger.

Embarked:
Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton). Indicates the port where the passenger boarded the Titanic.

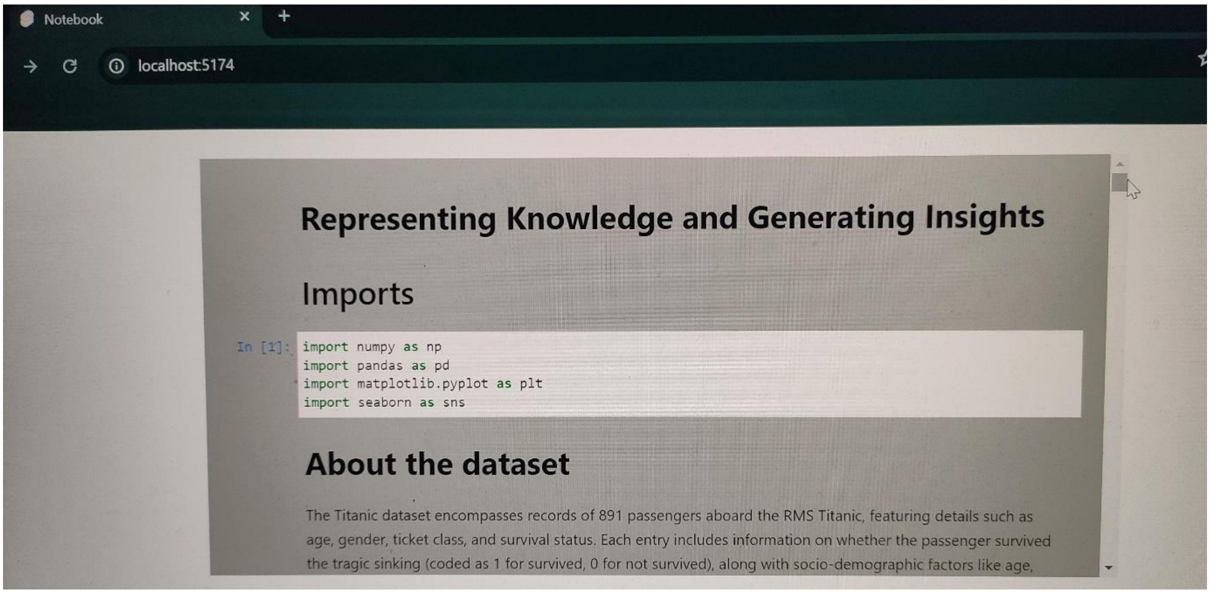
Survived:
Survival (0 = No, 1 = Yes). Indicates whether the passenger survived

Name:
Name of the passenger.

Ticket:
Ticket number of the passenger.

Cabin:
Cabin number of the passenger.

These are some insights about the features of the dataset.



Representing Knowledge and Generating Insights

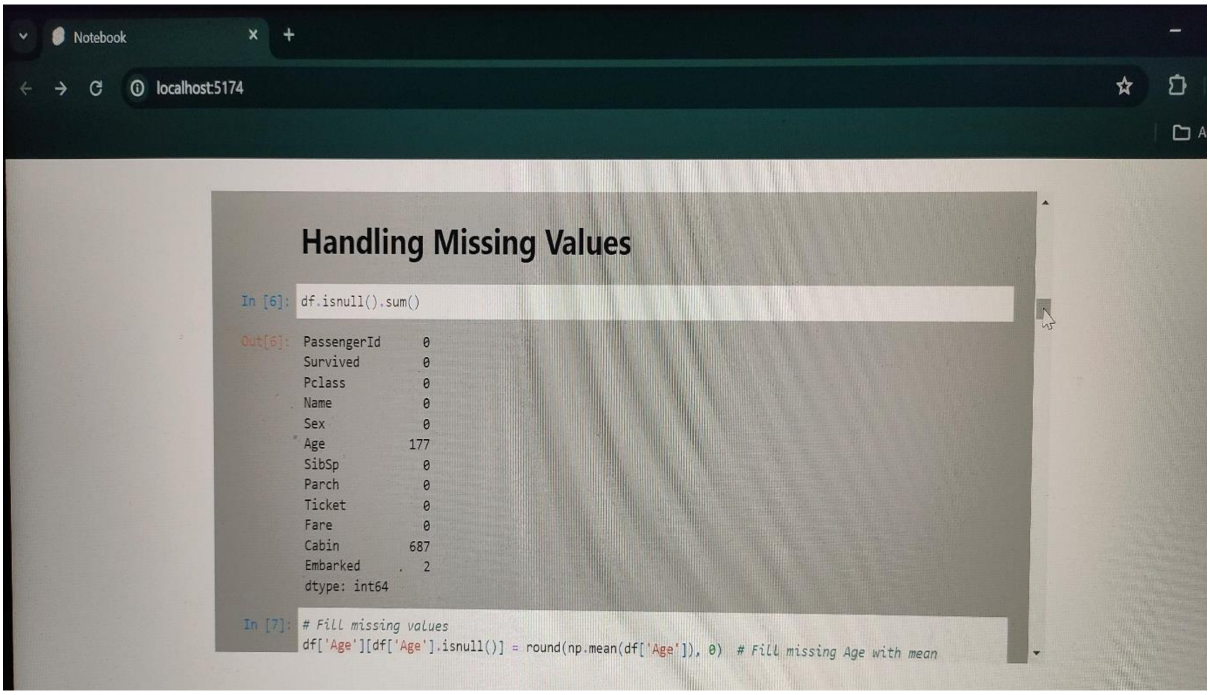
Imports

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

About the dataset

The Titanic dataset encompasses records of 891 passengers aboard the RMS Titanic, featuring details such as age, gender, ticket class, and survival status. Each entry includes information on whether the passenger survived the tragic sinking (coded as 1 for survived, 0 for not survived), along with socio-demographic factors like age,

This is the Google colab file that we have worked upon to represent knowledge and generate insights.



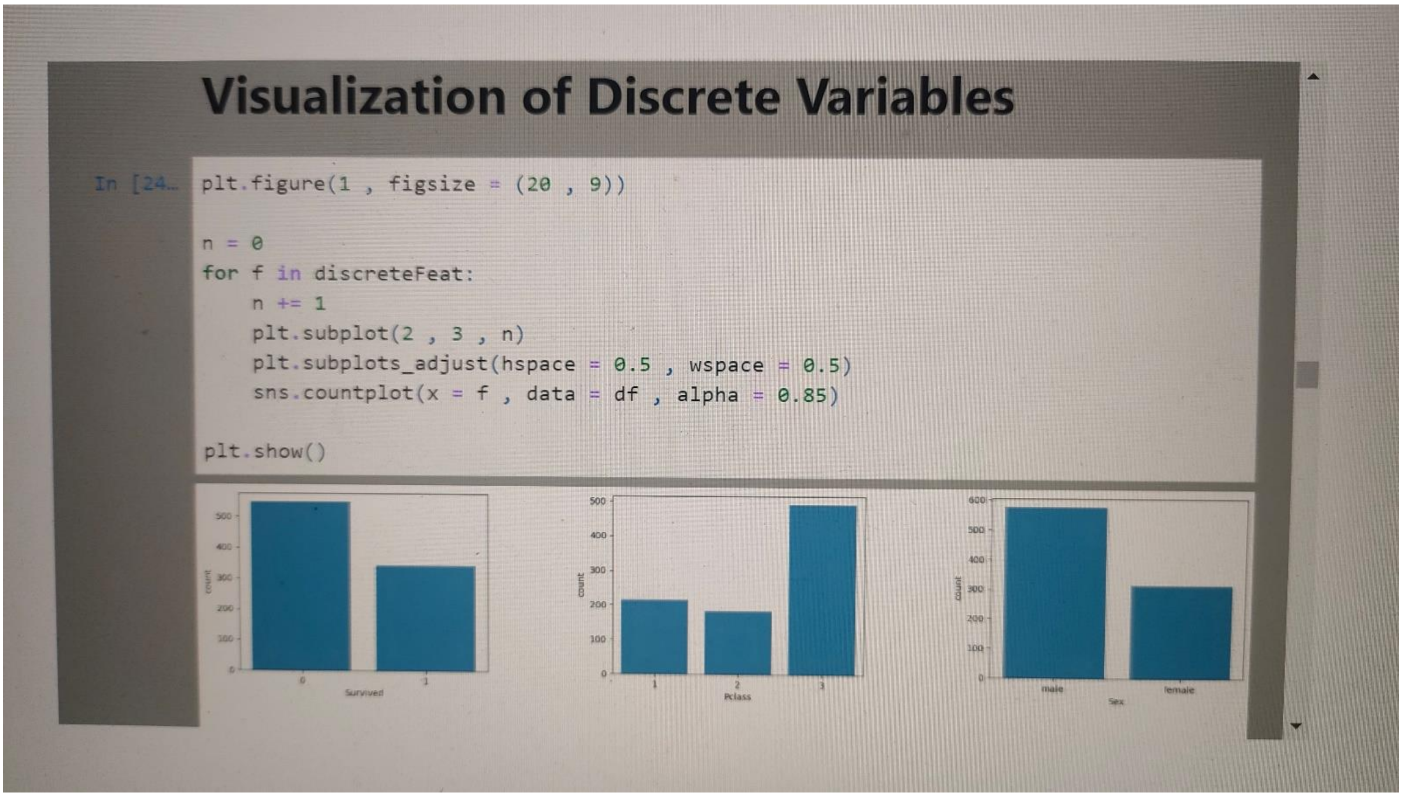
Handling Missing Values

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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
```

In [7]:
```
# Fill missing values
df['Age'][df['Age'].isnull()] = round(np.mean(df['Age']), 0)  # Fill missing Age with mean
```

In this screen we can see that we have some missing values so we handled them using median values.



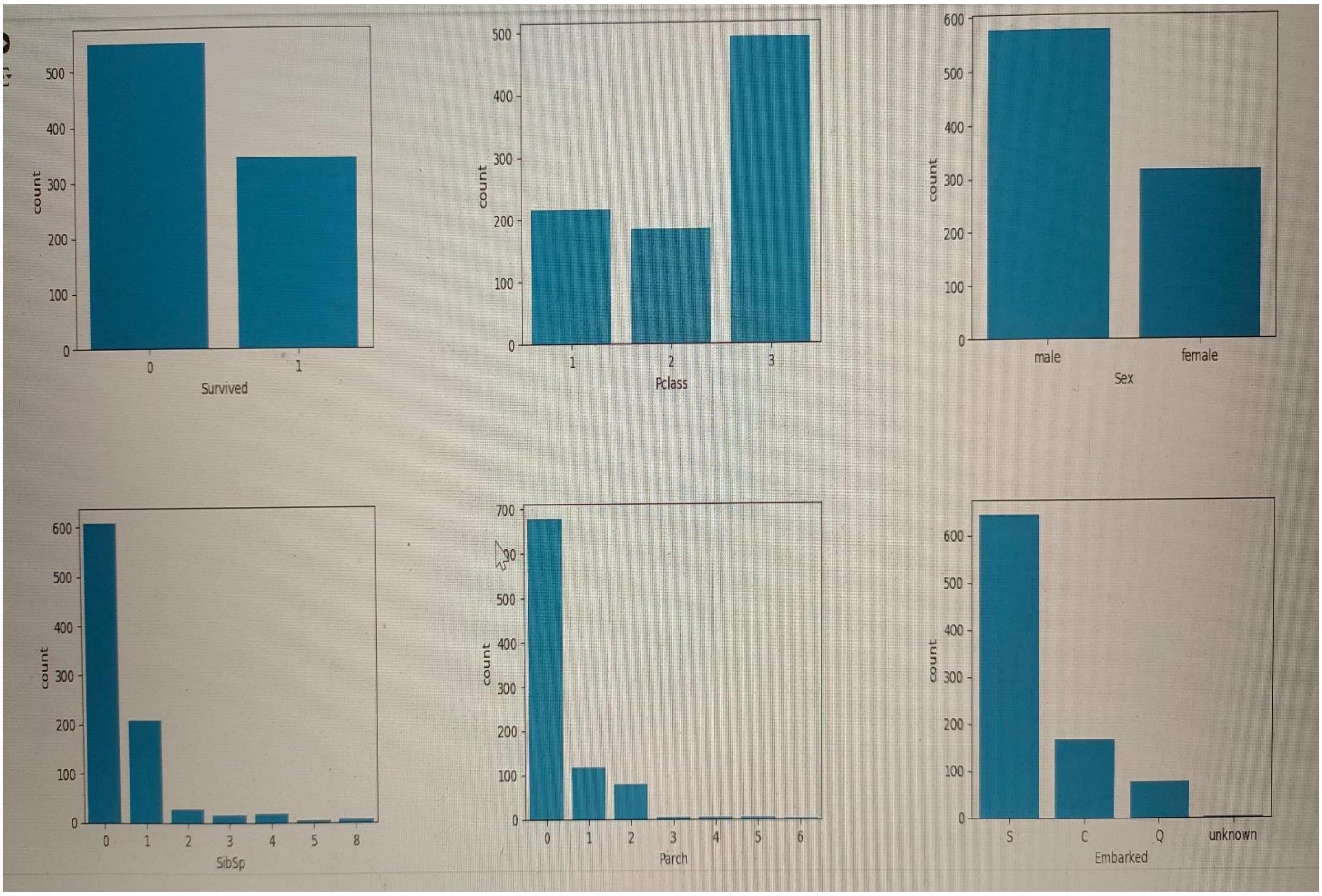
Visualization of Discrete Variables

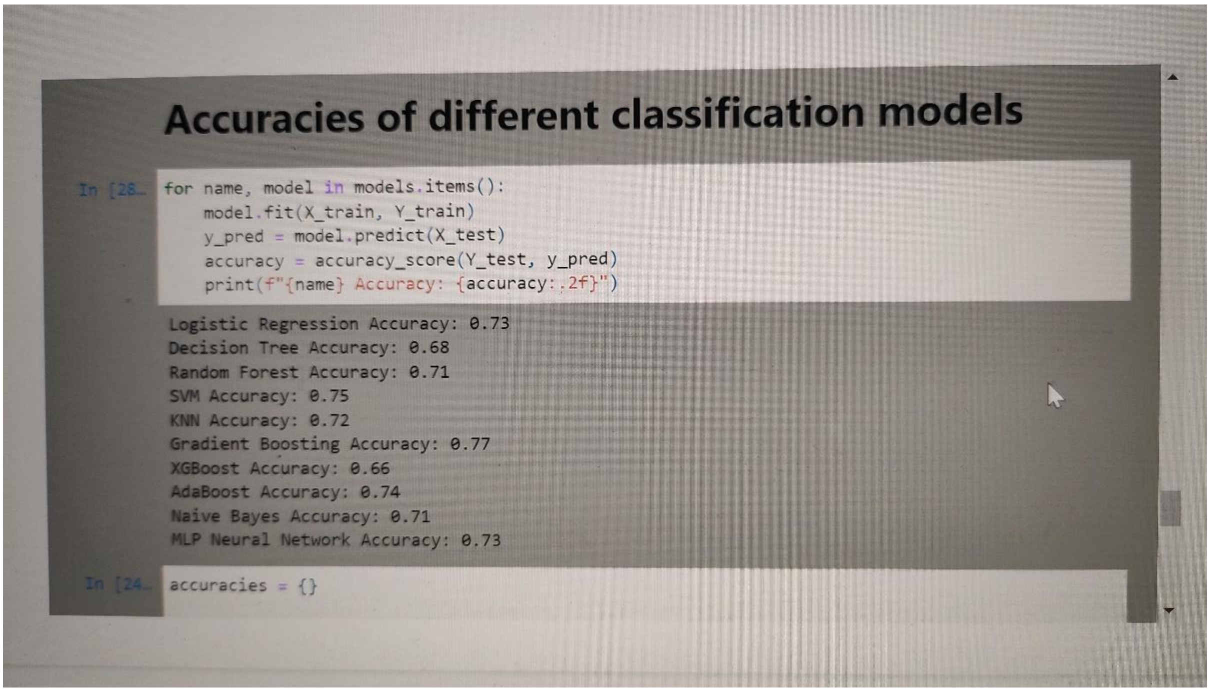
```
In [24… plt.figure(1 , figsize = (20 , 9))

        n = 0
        for f in discreteFeat:
            n += 1
            plt.subplot(2 , 3 , n)
            plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
            sns.countplot(x = f , data = df , alpha = 0.85)

        plt.show()
```

Here we are visualizing Discrete Variables using count plot





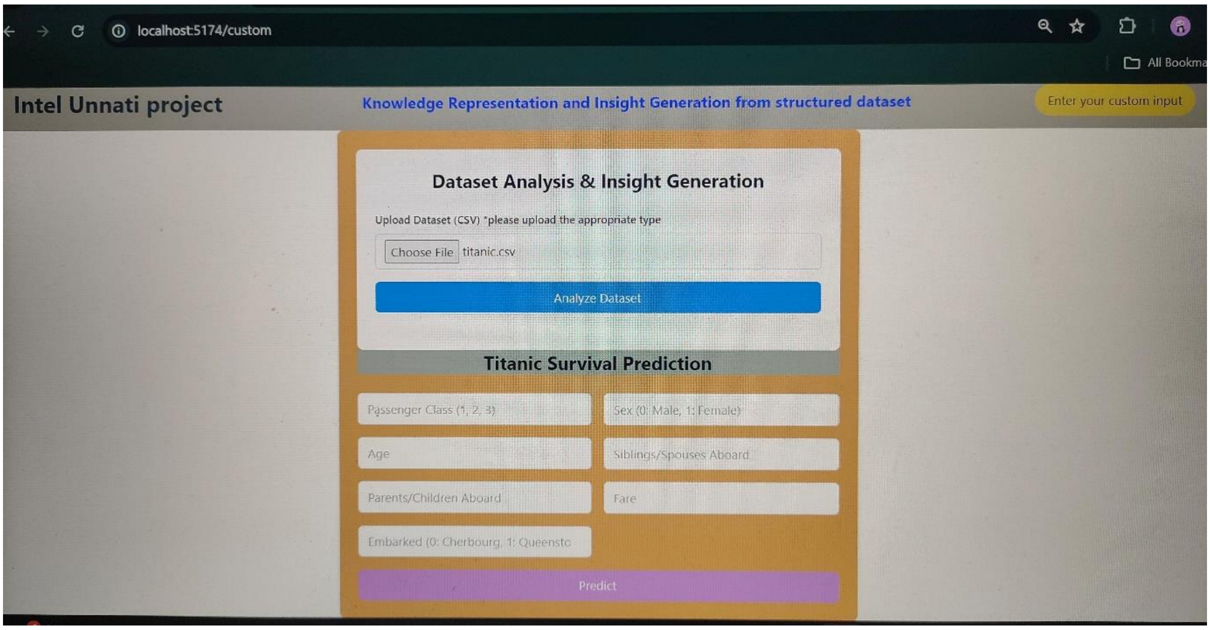
Accuracies of different classification models

```
In [28… for name, model in models.items():
            model.fit(X_train, Y_train)
            y_pred = model.predict(X_test)
            accuracy = accuracy_score(Y_test, y_pred)
            print(f"{name} Accuracy: {accuracy:.2f}")

        Logistic Regression Accuracy: 0.73
        Decision Tree Accuracy: 0.68
        Random Forest Accuracy: 0.71
        SVM Accuracy: 0.75
        KNN Accuracy: 0.72
        Gradient Boosting Accuracy: 0.77
        XGBoost Accuracy: 0.66
        AdaBoost Accuracy: 0.74
        Naive Bayes Accuracy: 0.71
        MLP Neural Network Accuracy: 0.73

In [24… accuracies = {}
```

In collab file we worked on different models and we checked the Accuracies of Different classification models.



localhost:5174/custom

Intel Unnati project

Knowledge Representation and Insight Generation from structured dataset

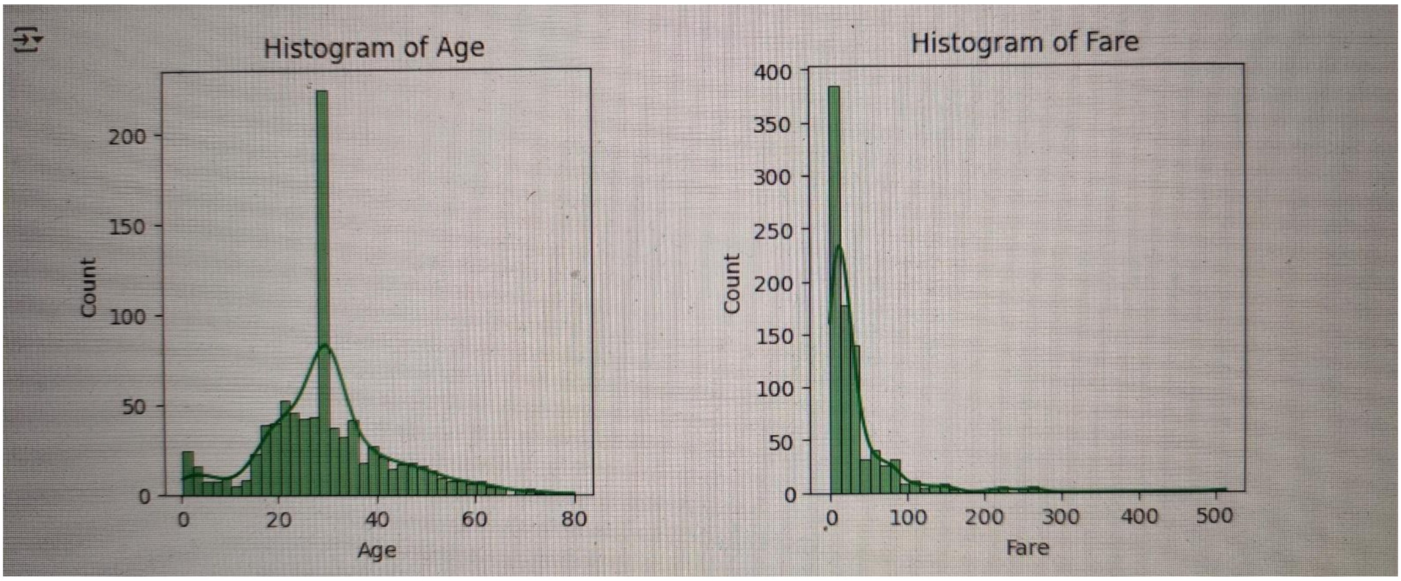
Dataset Analysis & Insight Generation

Upload Dataset (CSV) *please upload the appropriate type

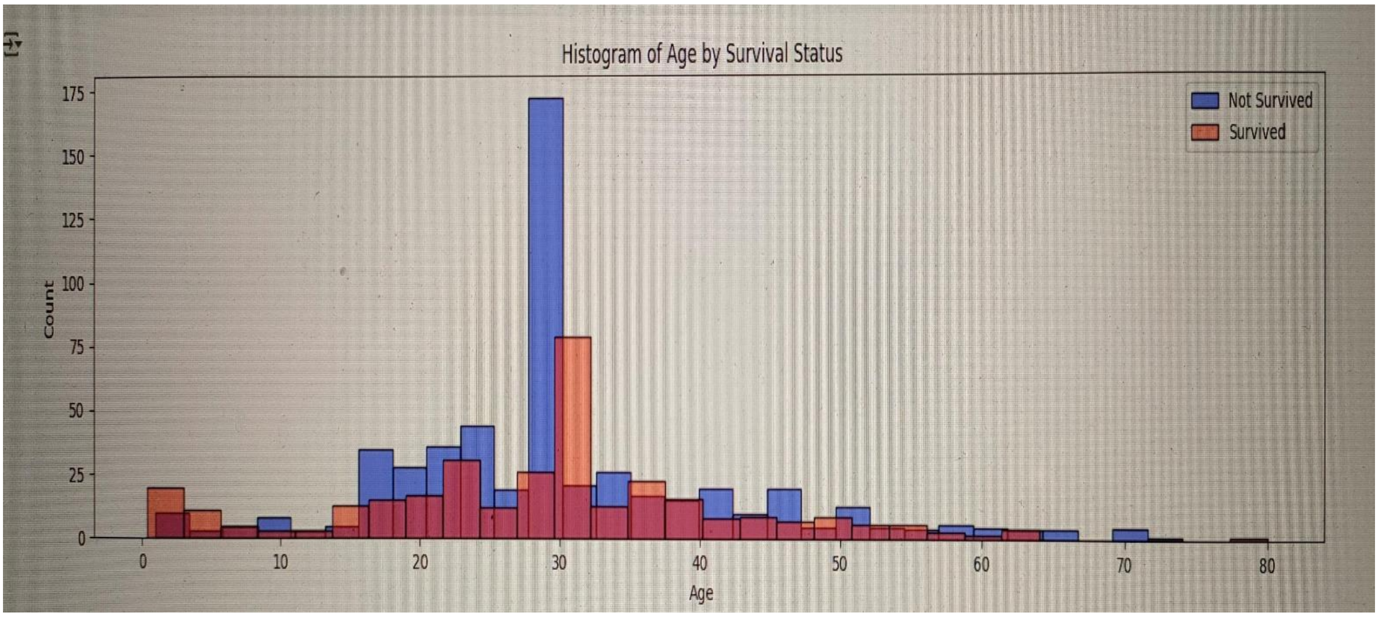
Choose File titanic.csv

Titanic Survival Prediction

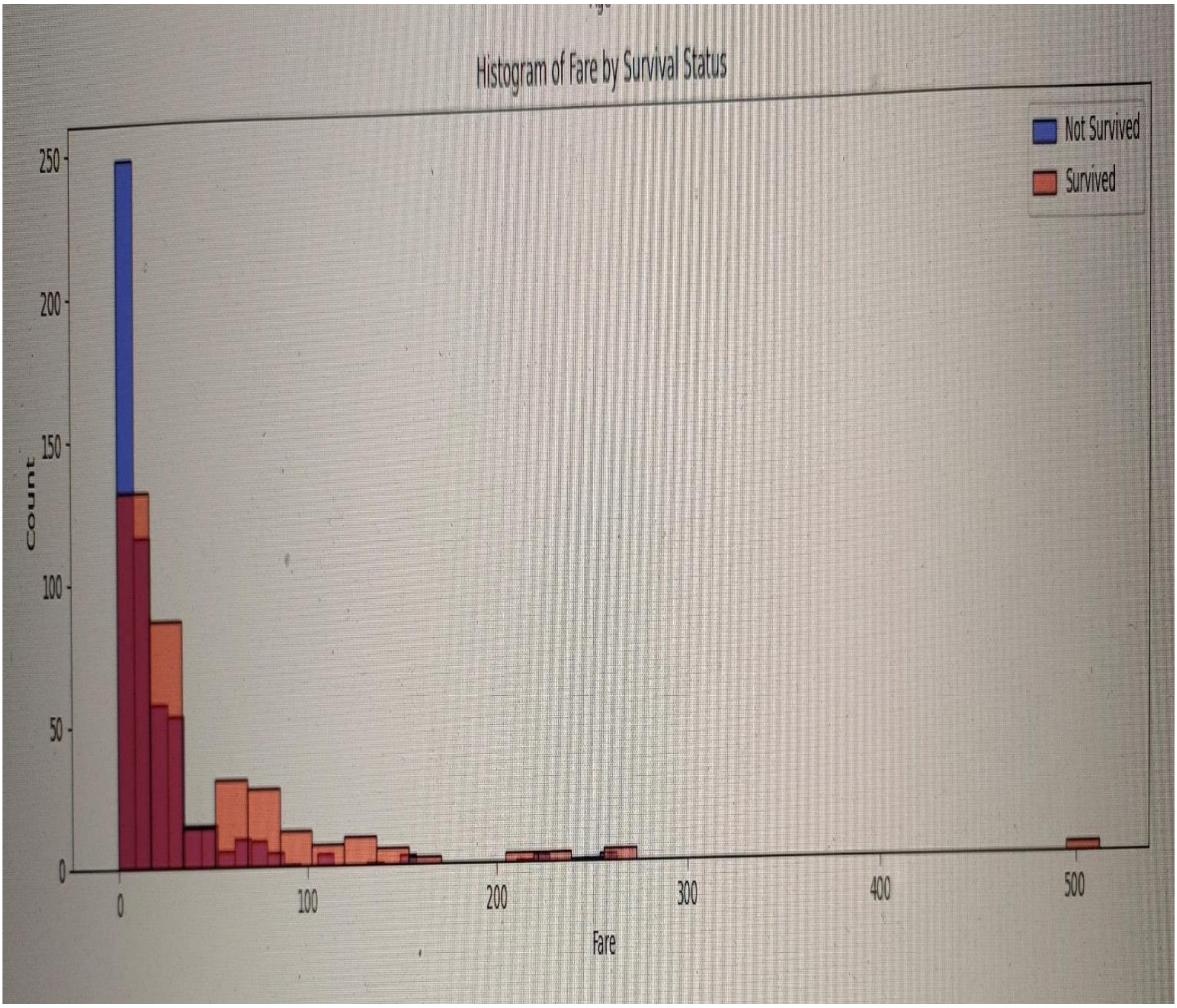
We can custom input when we click on right side button and we get this screen to predict.



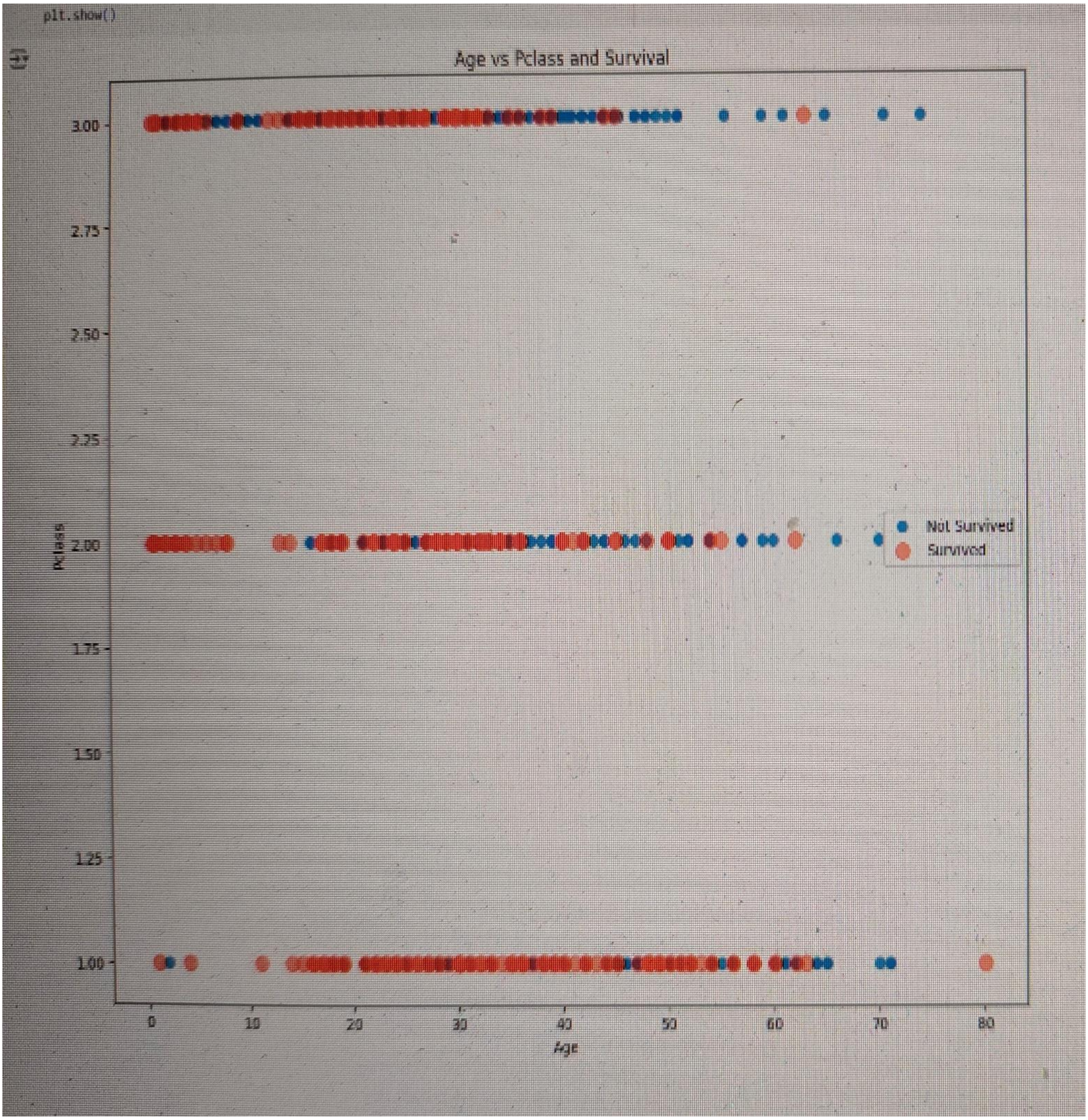
Here we visualized continuous variables:Age and fare using histogram to understand the data better.



Here we visualized continuous variable Age using histogram to understand the survival status based on age.

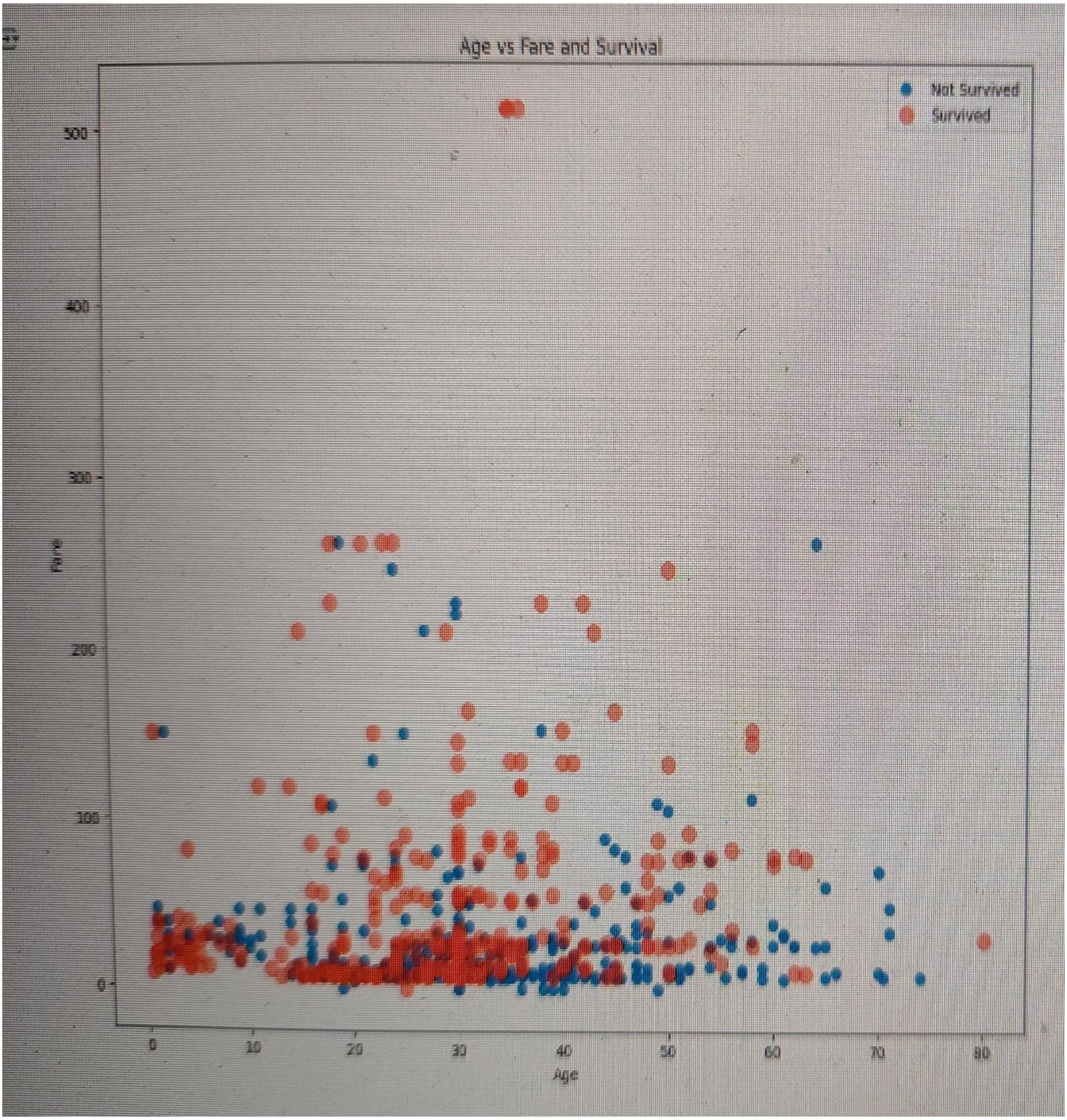


Here we visualized continuous variable Fare using histogram to understand the survival status based on Fare.



```
plt.show()
```

Here we visualized Age, Pclass using Scatterplot to understand the survival status based on Age and Pclass



Here we visualized Age, Fare using Scatterplot to understand the survival status based on Age and Fare.

5.CONCLUSION: (3 marks)

Findings:

Summary of Insights Generated:

Survival Analysis on Titanic Dataset:

Overall Survival Rate: Approximately 38% of passengers survived.

Passenger Class: First-class passengers had the highest survival rate (~62%), followed by second-class (~47%) and third-class (~24%).

Gender: Females had a significantly higher survival rate (~74%) compared to males (~19%).

Age: Younger passengers, especially children under 16, had a higher survival rate (~55%).

Fare: Higher fare-paying passengers had better survival chances, indicating a correlation with ticket class.

Embarkation Port: Passengers who embarked at Cherbourg had the highest survival rate (~55%), followed by Queenstown (~39%) and Southampton (~34%).

Predictive Modeling: Models such as Logistic Regression, Decision Trees, and Random Forests were employed to predict passenger survival.

Features like Pclass, Sex, Age, SibSp, Parch, Fare, and Embarked were key predictors. Future Work:

Future work:

To Improve Model Accuracy:

Feature Engineering: Derive new features from existing ones to capture more complex relationships (e.g., family size from SibSp and Parch).

Use domain knowledge to create meaningful features (e.g., combining fare and class to create a socio-economic status feature).

Advanced Modeling Techniques: Explore more complex models like Gradient Boosting Machines (GBM), XGBoost, or ensemble methods to capture non-linear relationships.

Implement hyperparameter tuning using grid search or random search to optimize model performance.

Handling Missing Data: Use advanced imputation techniques to handle missing values, ensuring that the imputed values maintain the integrity of the data distribution.

Suggestions for Improving the Model:

Data Augmentation: Augment the dataset by generating synthetic data points that follow the same distribution as the original dataset. This can help in improving model robustness, especially with smaller datasets.

Cross-validation: Use k-fold cross-validation to ensure that the model performance is consistent across different subsets of the data and to avoid overfitting.

Ensemble Learning: Combine multiple models to form an ensemble that can provide more accurate predictions by leveraging the strengths of each individual model.

Domain-Specific Adjustments: Incorporate domain knowledge into the model design and feature selection process to ensure that the model captures relevant patterns specific to the problem context.

By implementing these suggestions and continuously refining the approach, the accuracy and reliability of the insights generated from structured datasets can be significantly enhanced. This, in turn, will contribute to better decision-making and a deeper understanding of the underlying data.

Team Members and Contributions

1.Podilapu Keerthi (Team Lead)

Leadership and Strategy: Directed the overall project, focusing on delivering valuable insights and robust machine learning models.

Exploratory Data Analysis (EDA): Utilized Colab notebooks to perform in-depth data analysis, identifying key patterns and trends.

API Development: Played a crucial role in developing a Flask-based API to seamlessly integrate machine learning models into the website.

2.Velikanti Mythili

UI Design and Implementation: Designed and implemented an intuitive, responsive user interface (UI) using Svelte and Tailwind CSS, ensuring a seamless user experience.

API Development: Played a crucial role in developing a Flask-based API to seamlessly integrate machine learning models into the website.

3.Arisinapalli Sumanth

Data Analysis and Preprocessing: Conducted comprehensive exploratory data analysis (EDA) to understand and preprocess the data effectively.

Frontend Enhancement: Contributed significantly to frontend development by enhancing UI/UX and integrating insightful data visualizations.

4. Kelige Vinayak

Frontend Feature Implementation: Key contributor in developing essential frontend features, such as user input forms.

API Development: Played a crucial role in developing a Flask-based API to seamlessly integrate machine learning models into the website.

5.Syed Shoaibuddin

Backend Interaction: Ensured smooth interaction between frontend elements and backend services, contributing to a cohesive and efficient user experience.