

# Data and Artificial Intelligence

## Cyber Shujaa Program

### Week 3 Assignment

### Exploratory Data Analysis

**Student Name:** Faith Jeptoo

**Student ID:** CS-DA02-25005

## Table of Contents

1. **Cover Page**
2. **Introduction**
3. **Importing Libraries and Loading the Dataset**
4. **Initial Data Exploration**
  - 4.1. Dataset Overview
  - 4.2. Feature Profiling
5. **Checking for Missing Values and Duplicates**
6. **Univariate Analysis**
  - 6.1. Age Distribution
  - 6.2. Fare Distribution
  - 6.3. Passenger Class, Gender, and Embarked
7. **Bivariate Analysis**
  - 7.1. Fare vs Passenger Class
  - 7.2. Survival by Gender and Embarked
  - 7.3. Age and Survival
8. **Multivariate Analysis**
  - 8.1. Class, Gender, and Survival
  - 8.2. Age, Fare, and Survival
9. **Outlier Detection and Handling**
  - 9.1. Age Outliers
  - 9.2. Fare Outliers
10. **Target Variable (Survived) Analysis**
11. **Conclusion**
12. **Notebook Link**

# Introduction

This assignment focuses on performing **Exploratory Data Analysis (EDA)** on the famous **Titanic dataset** from Kaggle.

The goal is to gain hands-on experience in exploring, cleaning, and visualizing data to uncover meaningful patterns and insights.

EDA involves:

- Understanding the structure and quality of the data
- Handling missing values and outliers
- Performing univariate, bivariate, and multivariate analyses
- Exploring the target variable (**Survived**) to identify influencing factors

## Tasks Completed

### Step 1: Importing Libraries and Loading the Dataset

```
import pandas as pd          # Data manipulation and analysis
```

```
import numpy as np          # Numerical computations
```

```
import matplotlib.pyplot as plt # Data visualization
```

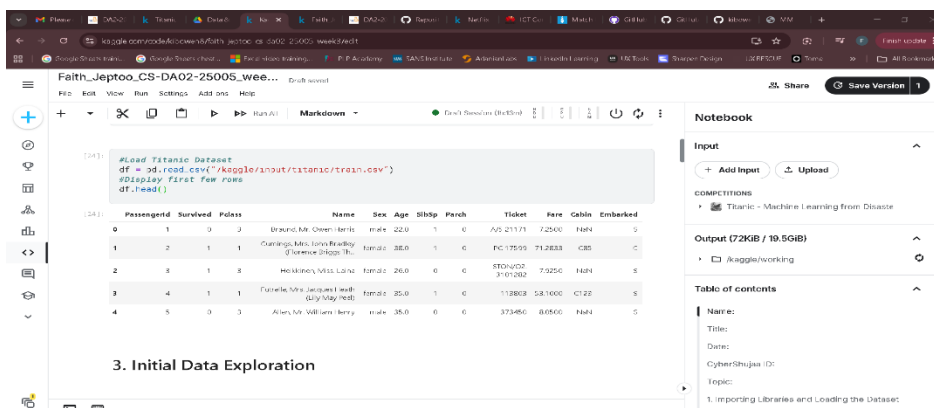
```
import seaborn as sns       # Advanced plots
```

```
sns.set(style="whitegrid")
```

```
# Load the Titanic dataset
```

```
df = pd.read_csv("/kaggle/input/titanic/train.csv") # Display first few rows
```

```
df.head()
```



The screenshot shows a Jupyter Notebook with the following code and output:

```
[24]: #Load Titanic Dataset
df = pd.read_csv("/kaggle/input/titanic/train.csv")
#Display first few rows
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	ParCh	Ticket	Fare	Cabin	Embarked
0	1	0	3	Brands Hatch Owen /berts	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Quarles, Mrs. Louis Bradley	titanic	35.0	1	0	PC-17599	71.2833	C85	C
2	3	1	3	Hickman, Miss Lland	titanic	26.0	0	0	STON/OOD	7.9250	NaN	S
3	4	1	1	Casselle, Mrs. Jacques Heath	titanic	35.0	1	0	113803	54.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

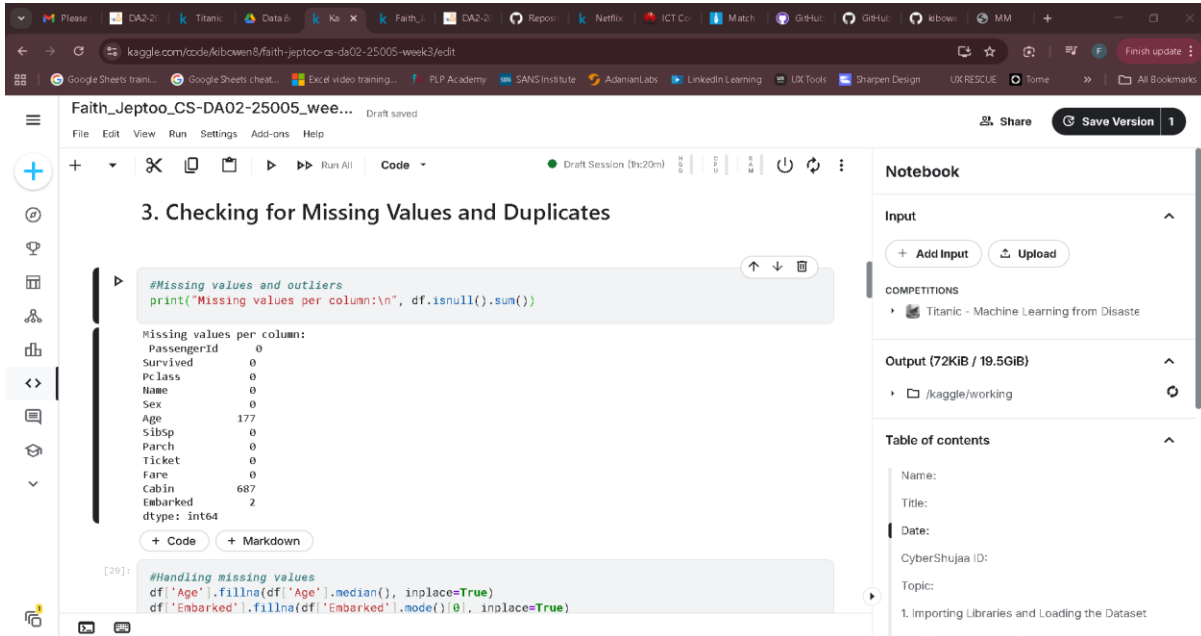
3. Initial Data Exploration



## Step 3: Checking for Missing Values and Duplicates

*#Missing values and outliers*

`print("Missing values per column:\n", df.isnull().sum())`



The screenshot shows a Jupyter Notebook interface with the following content:

```
#Missing values and outliers
print("Missing values per column:\n", df.isnull().sum())
```

The output of the code is displayed as follows:

```
Missing values per column:
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
```

The notebook also shows a table of contents on the right side, listing the following sections:

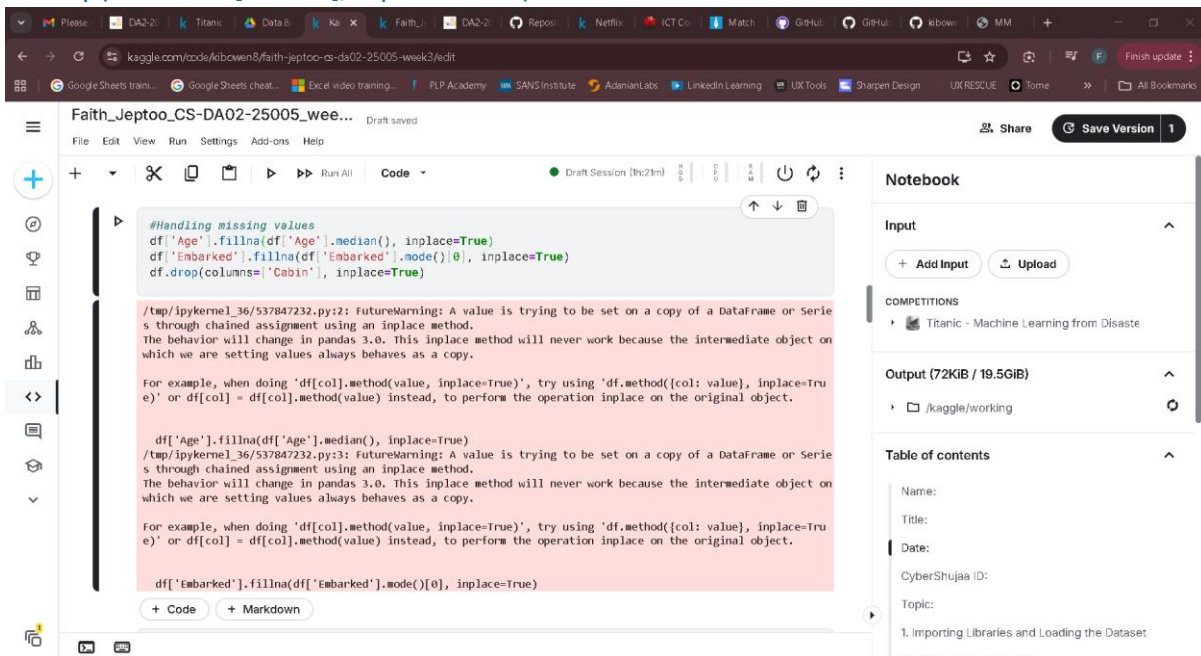
- 1. Importing Libraries and Loading the Dataset
- 2. Initial Data Exploration

*#Handling missing values*

`df['Age'].fillna(df['Age'].median(), inplace=True)`

`df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)`

`df.drop(columns=['Cabin'], inplace=True)`



The screenshot shows a Jupyter Notebook interface with the following content:

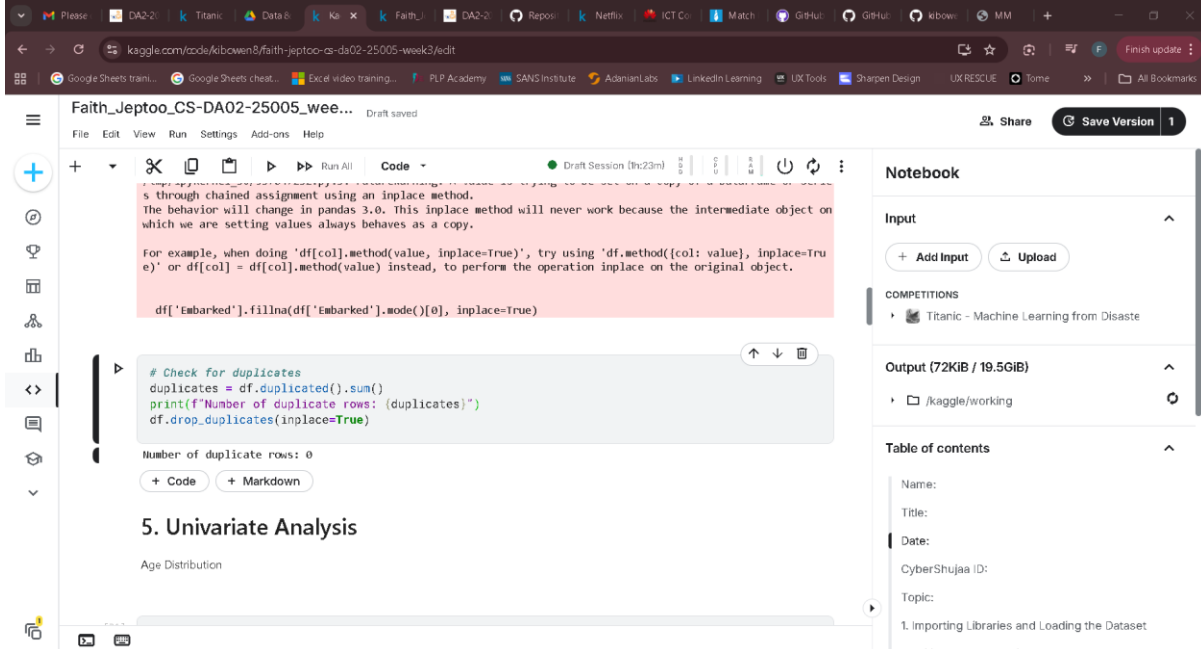
```
#Handling missing values
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
df.drop(columns=['Cabin'], inplace=True)
```

The notebook also shows a table of contents on the right side, listing the following sections:

- 1. Importing Libraries and Loading the Dataset
- 2. Initial Data Exploration

# Check for duplicates

```
duplicates = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicates}")
df.drop_duplicates(inplace=True)
```



The screenshot shows a Kaggle notebook titled "Faith\_Jepto\_CS-DA02-25005\_wee...". The code cell contains the following Python code:

```
# Check for duplicates
duplicates = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicates}")
df.drop_duplicates(inplace=True)
```

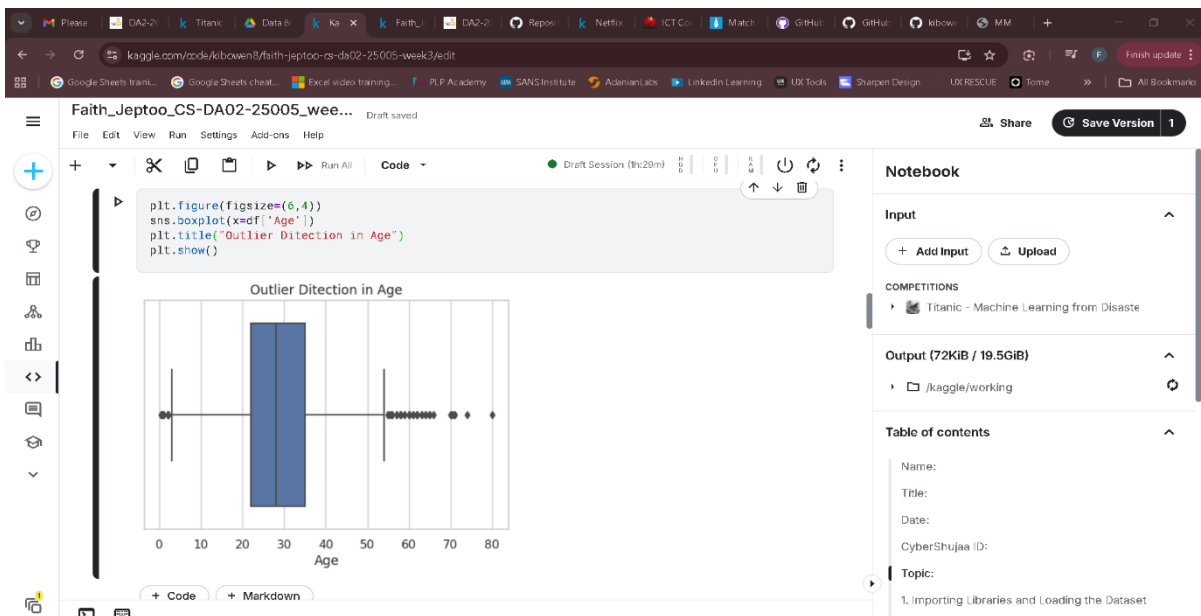
The output shows "Number of duplicate rows: 0". A warning message is displayed above the code cell, explaining that the behavior of `inplace=True` will change in pandas 3.0 and that the intermediate object on which we are setting values always behaves as a copy. The warning suggests using `df[col].method(value, inplace=True)` or `df[col] = df[col].method(value)` instead.

The notebook interface includes a sidebar with icons for file, code, output, and other features. The right sidebar shows the "Notebook" section with "Input" and "Output" tabs, and a "Table of contents" section.

## Step 4: Univariate Analysis

### Age Distribution

```
plt.figure(figsize=(6,4))
sns.boxplot(x=df['Age'])
plt.title("Outlier Ditection in Age")
plt.show()
```



The screenshot shows a Kaggle notebook titled "Faith\_Jepto\_CS-DA02-25005\_wee...". The code cell contains the following Python code:

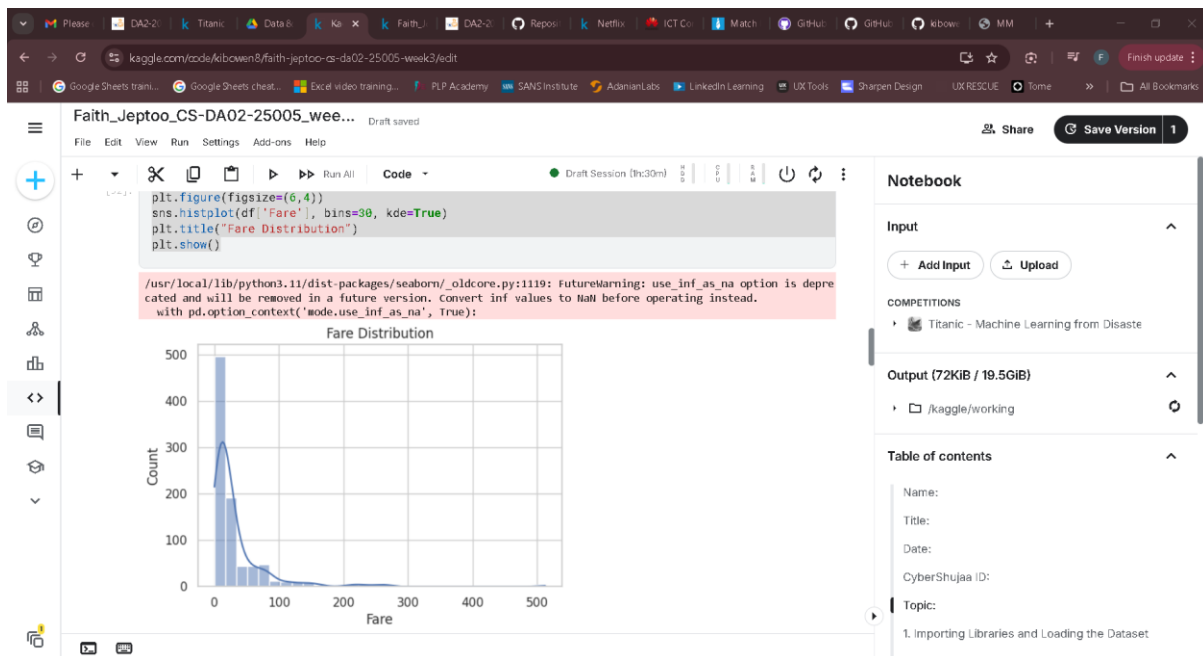
```
plt.figure(figsize=(6,4))
sns.boxplot(x=df['Age'])
plt.title("Outlier Ditection in Age")
plt.show()
```

The output is a boxplot titled "Outlier Ditection in Age". The x-axis is labeled "Age" and ranges from 0 to 80. The boxplot shows the distribution of ages, with a median around 30. There are several outliers plotted as individual points to the right of the whiskers, around 60-70.

The notebook interface includes a sidebar with icons for file, code, output, and other features. The right sidebar shows the "Notebook" section with "Input" and "Output" tabs, and a "Table of contents" section.

## Fare Distribution

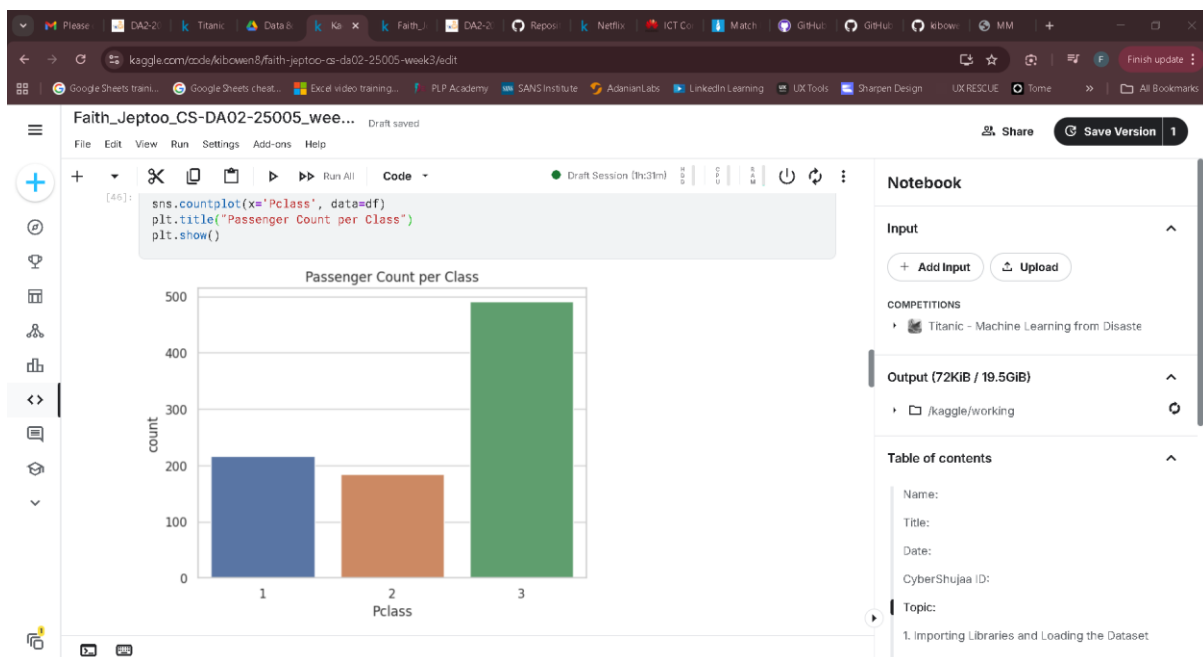
```
plt.figure(figsize=(6,4))
sns.histplot(df['Fare'], bins=30, kde=True)
plt.title("Fare Distribution")
plt.show()
```



## Passenger Class, Gender, and Embarked

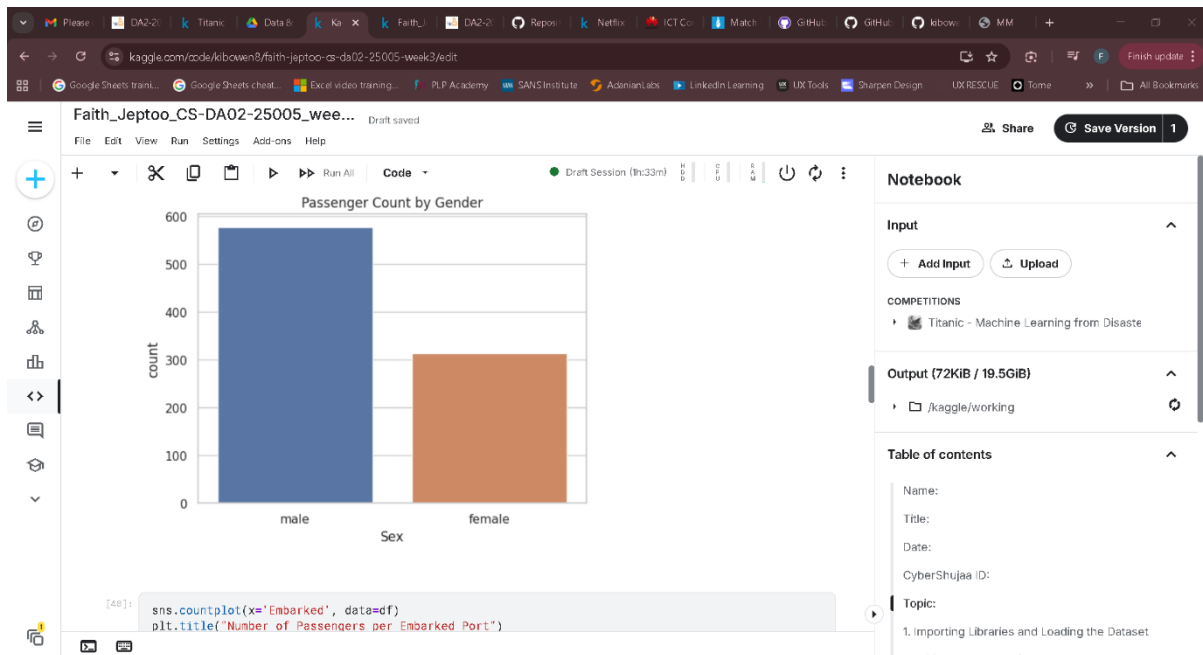
Class

```
sns.countplot(x='Pclass', data=df)
plt.title("Passenger Count per Class")
plt.show()
```



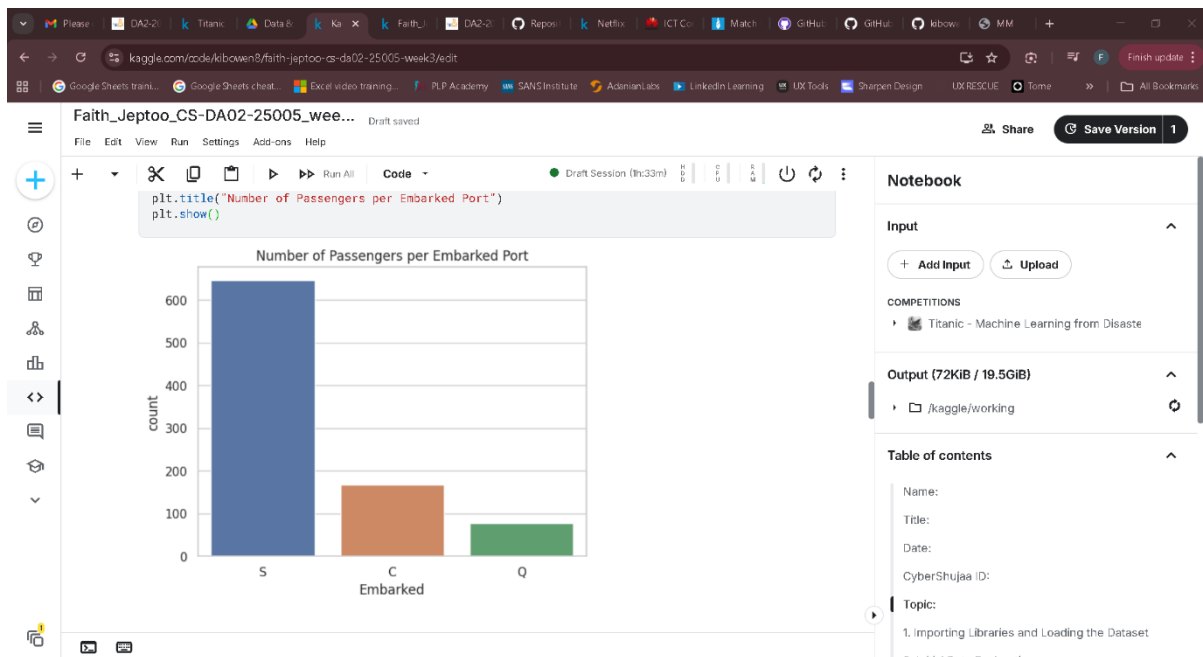
## Gender

```
sns.countplot(x='Sex', data=df)
plt.title("Passenger Count by Gender")
plt.show()
```



## Embarked

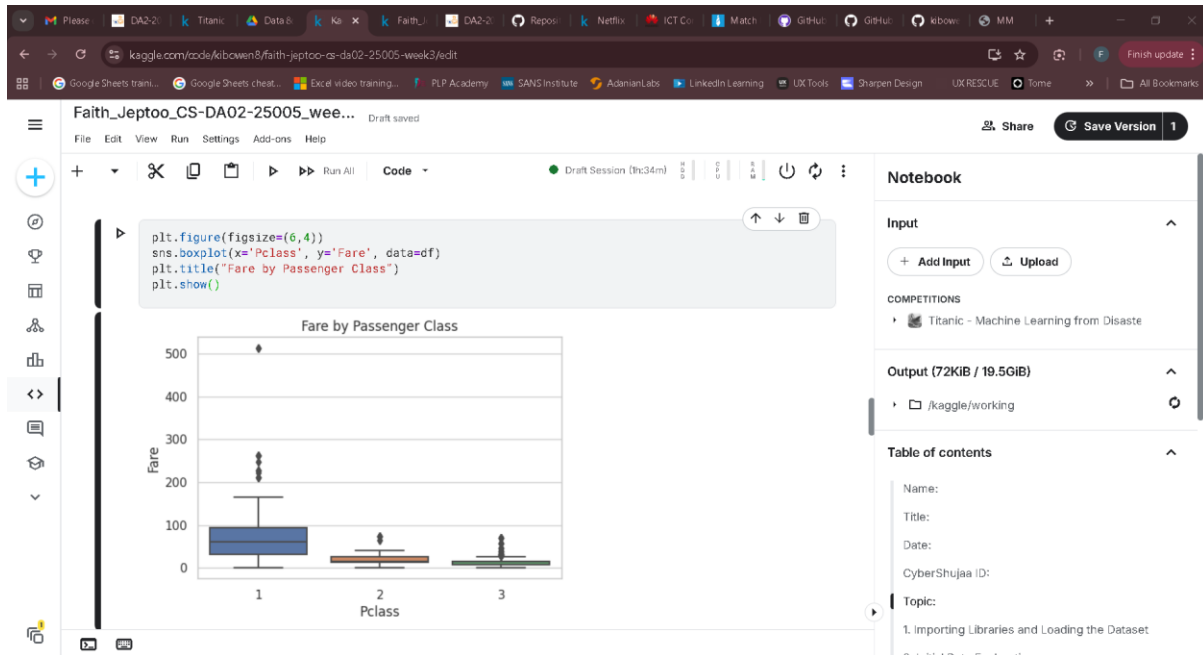
```
sns.countplot(x='Embarked', data=df)
plt.title("Number of Passengers per Embarked Port")
plt.show()
```



## Step 5: Bivariate Analysis

### Fare vs Pclass

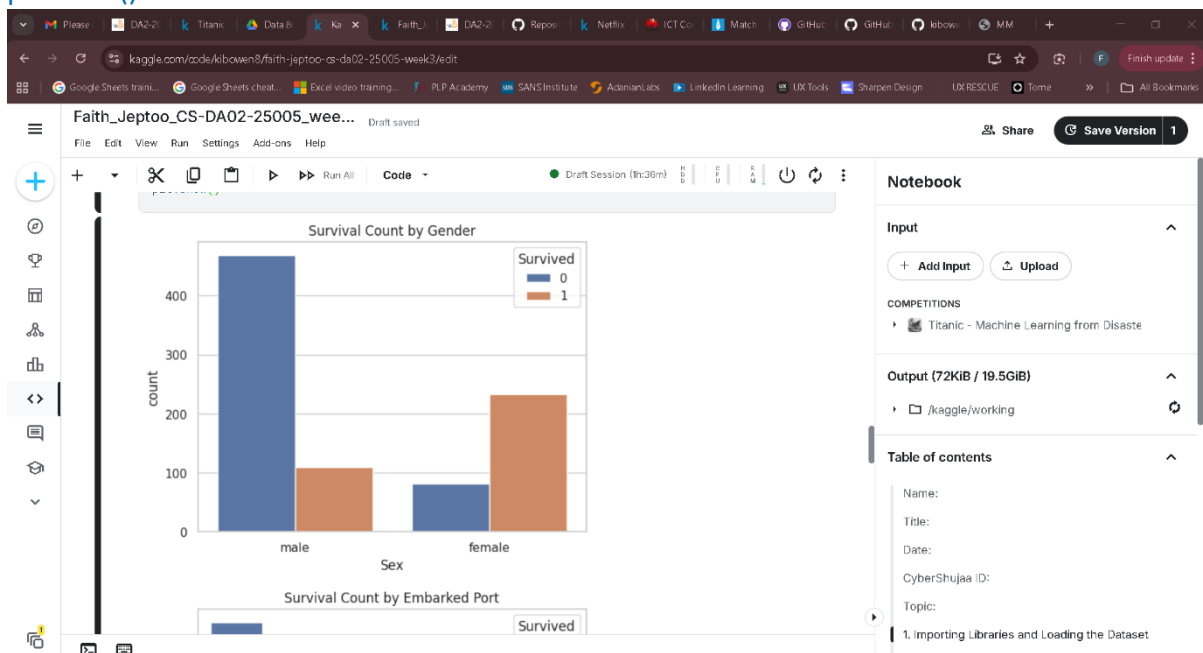
```
plt.figure(figsize=(6,4))
sns.boxplot(x='Pclass', y='Fare', data=df)
plt.title("Fare by Passenger Class")
plt.show()
```



### Survival by Gender and Embarked

Gender

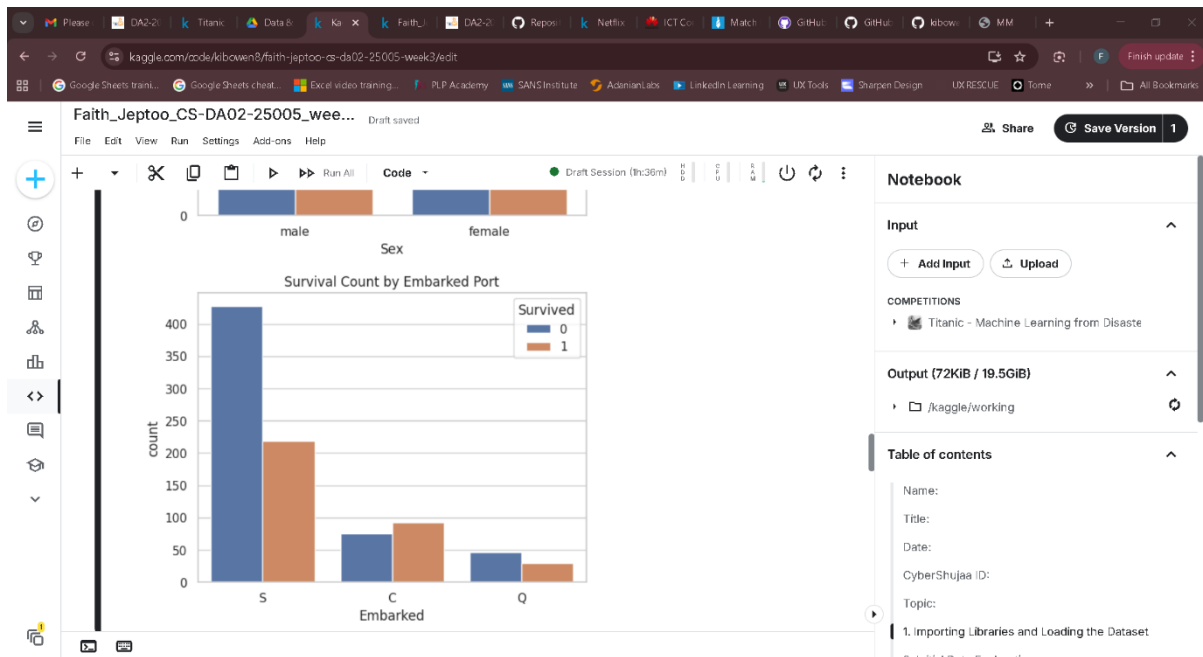
```
sns.countplot(x='Sex', hue='Survived', data=df)
plt.title("Survival Count by Gender")
plt.show()
```





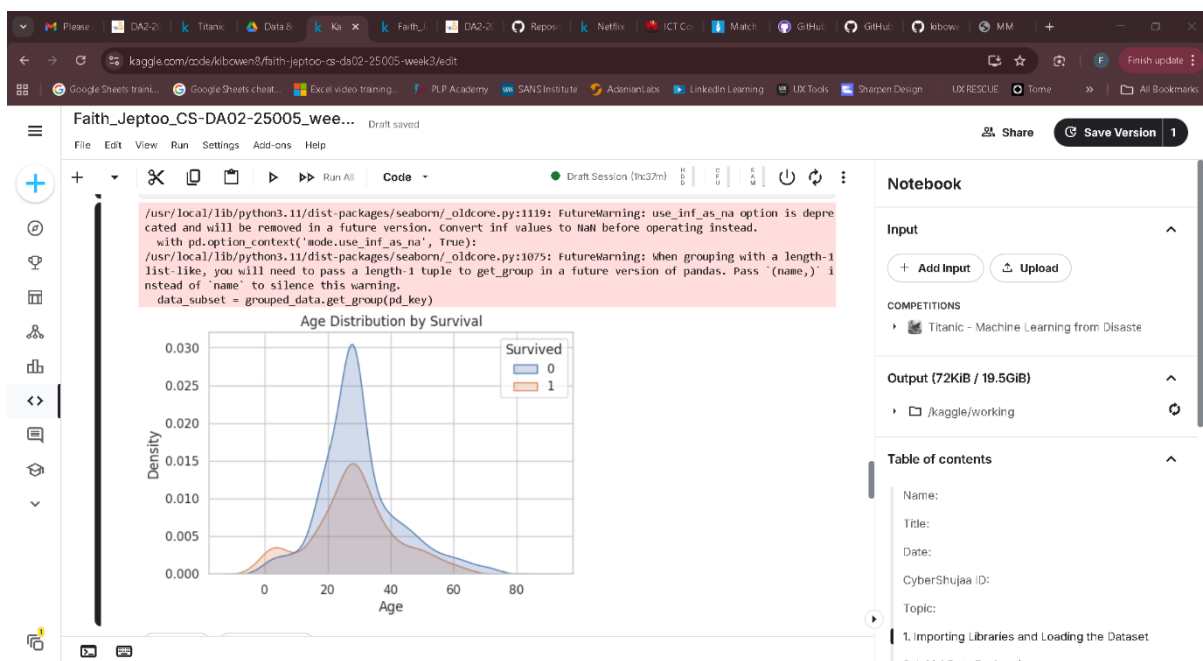
Emberked

```
sns.countplot(x='Embarked', hue='Survived', data=df)
plt.title("Survival Count by Embarked Port")
plt.show()
```



## Age and Survival

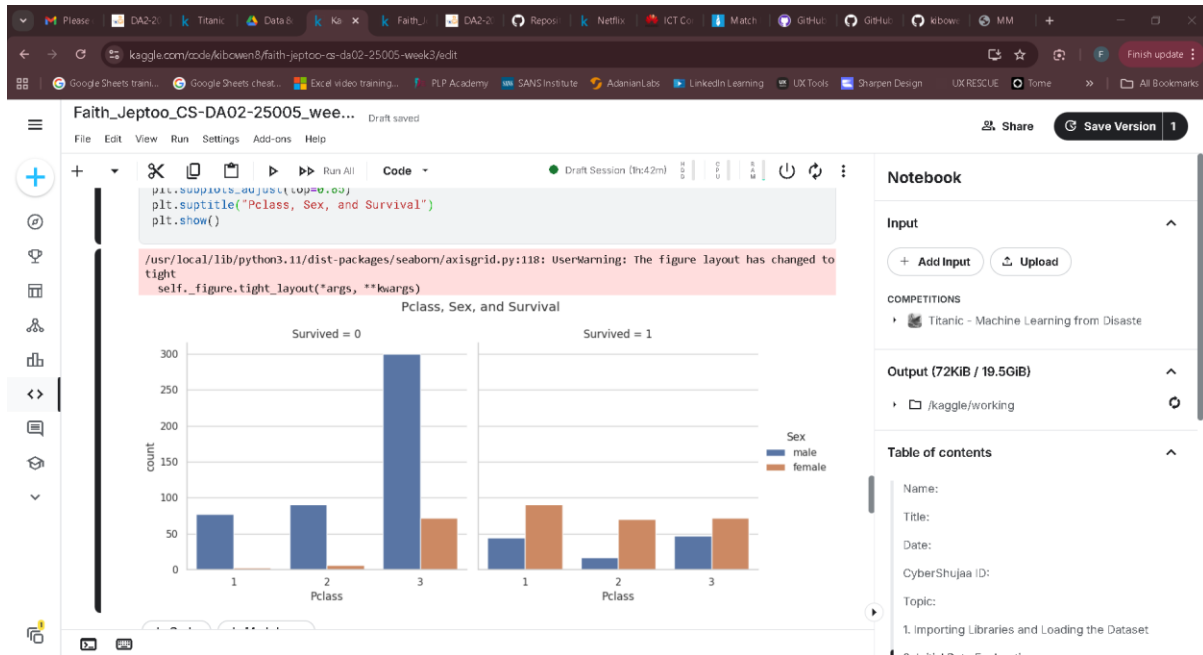
```
plt.figure(figsize=(6,4))
sns.kdeplot(data=df, x='Age', hue='Survived', fill=True)
plt.title("Age Distribution by Survival")
plt.show()
```



## Step 6: Multivariate Analysis

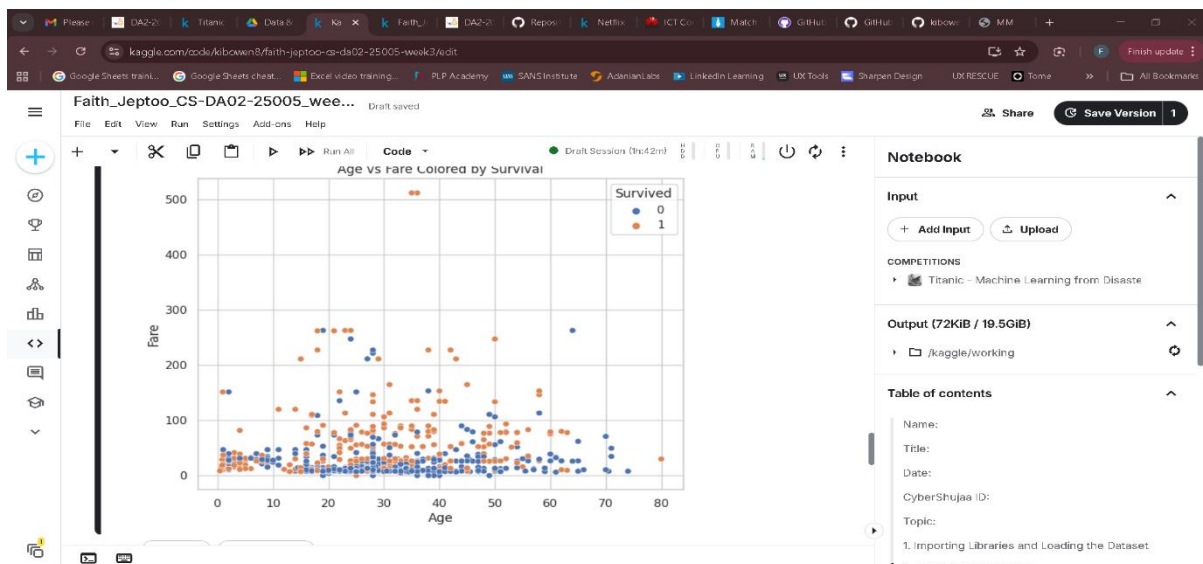
### Class, Gender, and Survival

```
sns.catplot(x='Pclass', hue='Sex', col='Survived', data=df, kind='count')
plt.subplots_adjust(top=0.85)
plt.suptitle("Pclass, Sex, and Survival")
plt.show()
```



### Age, Fare, and Survival

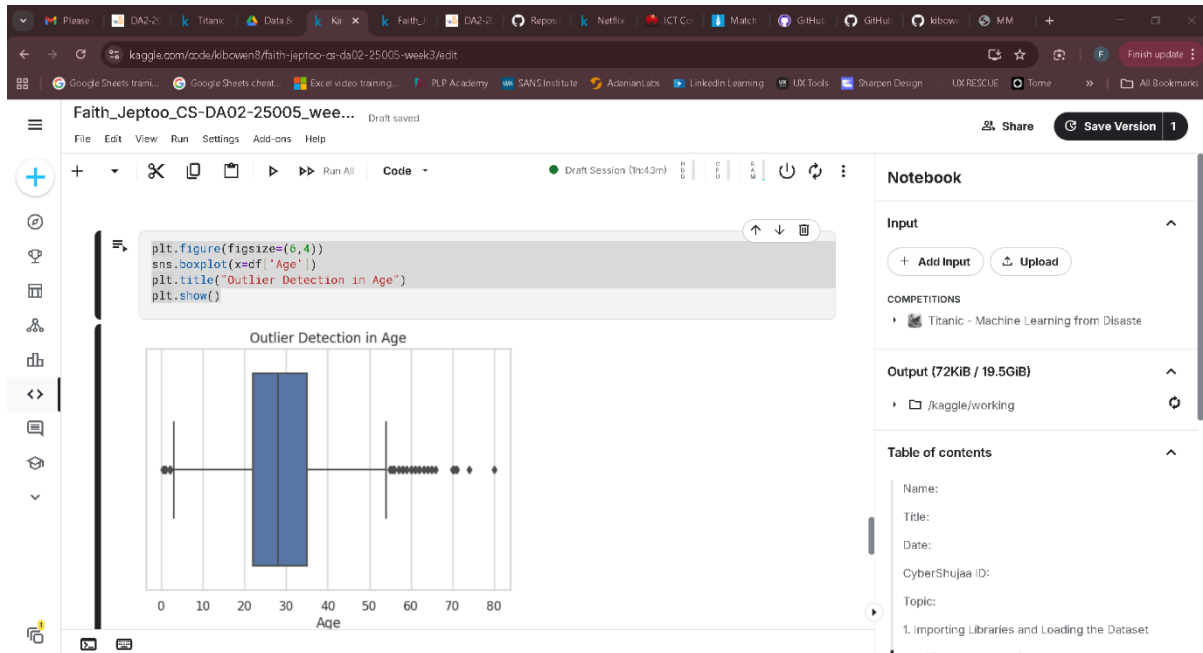
```
plt.figure(figsize=(8,6))
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title("Age vs Fare Colored by Survival")
plt.show()
```



## Step 7: Outlier Detection and Handling

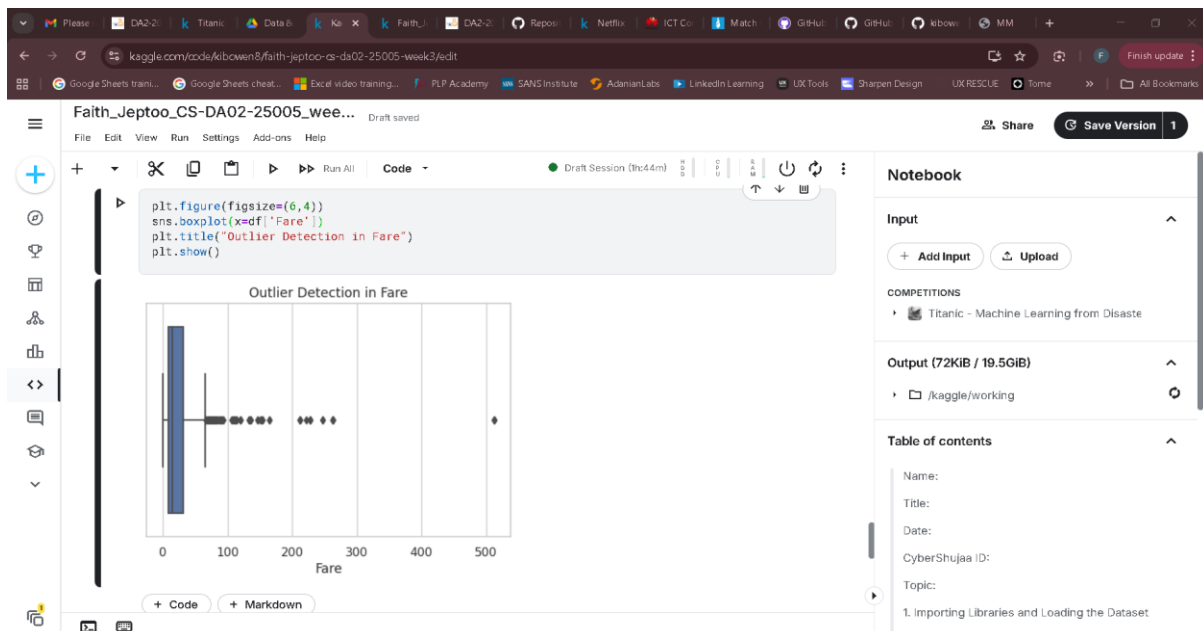
### Age Outliers

```
plt.figure(figsize=(6,4))
sns.boxplot(x=df['Age'])
plt.title("Outlier Detection in Age")
plt.show()
```



### Fare Outliers

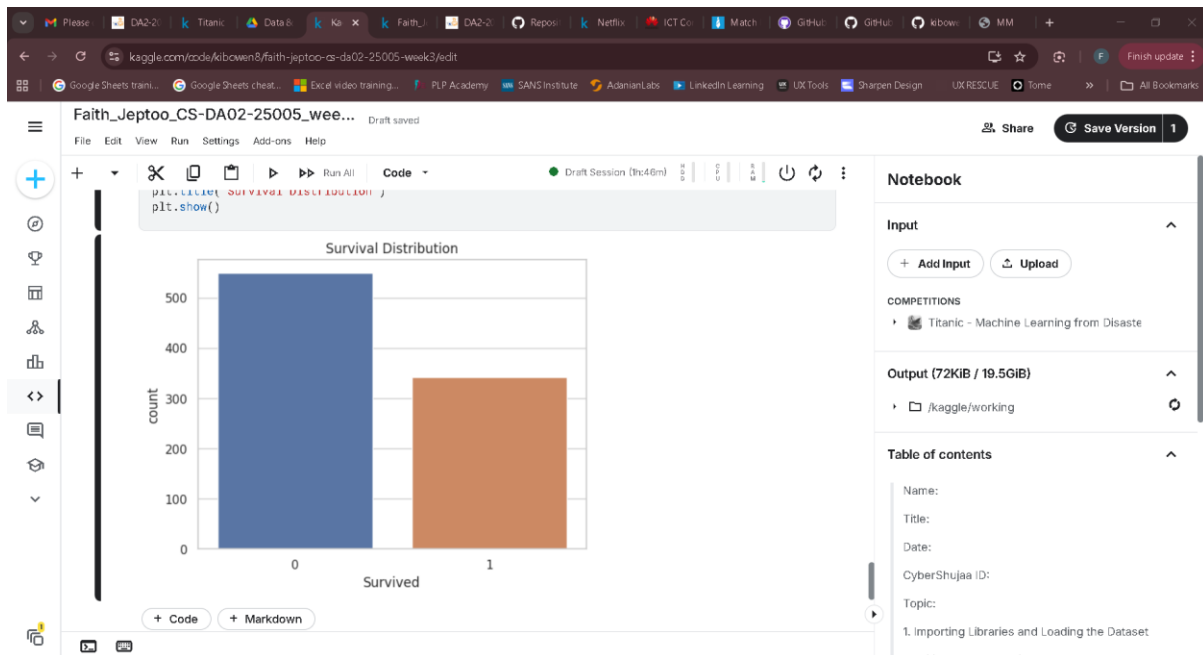
```
plt.figure(figsize=(6,4))
sns.boxplot(x=df['Fare'])
plt.title("Outlier Detection in Fare")
plt.show()
```



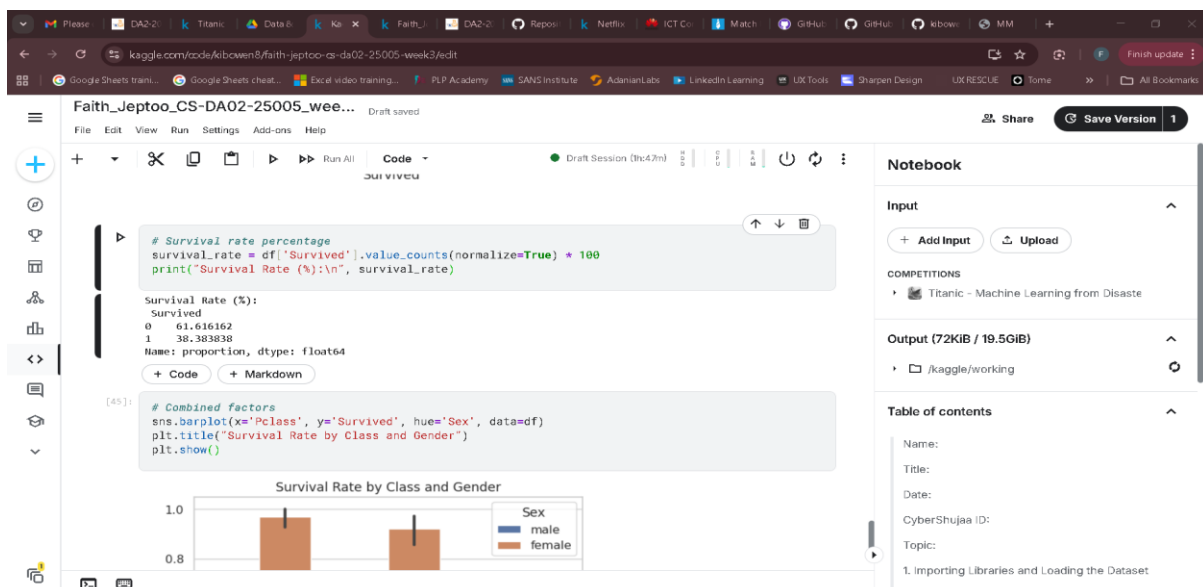
```
# Cap extreme fares at 99th percentile
fare_cap = df['Fare'].quantile(0.99)
df['Fare'] = np.where(df['Fare'] > fare_cap, fare_cap, df['Fare'])
```

## Step 8: Target Variable (Survived) Analysis

```
sns.countplot(x='Survived', data=df)
plt.title("Survival Distribution")
plt.show()
```



```
# Survival rate percentage
survival_rate = df['Survived'].value_counts(normalize=True) * 100
print("Survival Rate (%):\n", survival_rate)
```

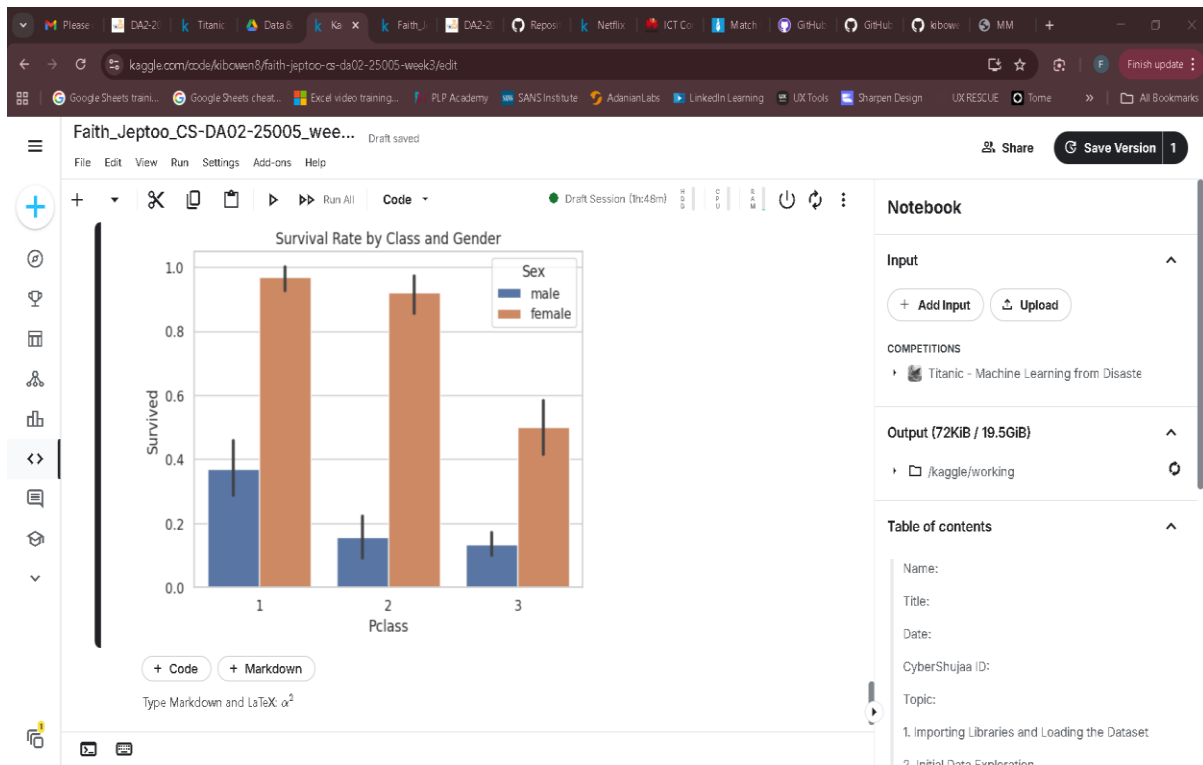


```
# Combined factors
```

```
sns.barplot(x='Pclass', y='Survived', hue='Sex', data=df)
```

```
plt.title("Survival Rate by Class and Gender")
```

```
plt.show()
```



## Conclusion

The Exploratory Data Analysis on the Titanic dataset revealed the following insights:

- The dataset contained missing values in *Age*, *Cabin*, and *Embarked* which were handled using imputation and removal.
- Most passengers were male and from 3rd class.
- Fare and Age were both right-skewed distributions.
- Survival was highly influenced by **Gender**, **Class**, and **Fare**.
- Outliers in *Fare* were capped to minimize skewness.
- The target variable (*Survived*) showed moderate imbalance, which is important for modeling

## Link To Notebook

<https://www.kaggle.com/code/kibowen8/faith-jeptoo-cs-da02-25005-week-3>