TITANIC SURVIVAL ANALYSIS

A Micro Project Report

Submitted by

[LAKSHMI NARASIMMAN.P]
Reg.no: [99220040297]

B.Tech - [COMPUTER SCIENCE AND ENGG],
STREAM-[DATA ANALYTICS]

Kalasalingam Academy of Research and Education
(Deemed to be University)
Anand Nagar, Krishnankoil - 626 126
[FEBRAURY] [2024]

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Bonafide record of the work done by [LAKSHMI NARASIMMAN.P] - [99220040297] in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Specialization of the Computer Science and Engineering, during the Academic Year [Even] Semester (2023-24)

[Ms.P.Kardeepa]
Project Guide
[Assistant Professor]
[Computer science and engineering]
Kalasalingam Academy of
Research and Education
Krishnan kovil – 626126

[Mrs.R.Durga Meena]
Faculty Incharge
[Assistant Professor]
[Computer science and engineering]
Kalasalingam Academy of
Research and Education
Krishnan kovil - 626126

[Ms.V.S.VetriSelvi]
[Assistant Professor]
Computer science and engineering]
Kalasalingam Academy of research and education
Krishnan kovil - 626126

Abstract

This data science project delves into the tragic sinking of the RMS Titanic, employing advanced analytical techniques to uncover patterns and predictors influencing passenger survival. Leveraging a comprehensive dataset encompassing passenger information, socio-economic factors, and survival outcomes, our study aims to contribute valuable insights into the dynamics of survival during the disaster.

The project begins with an exploratory data analysis (EDA) to gain a nuanced understanding of the dataset, including demographic distributions, class structures, and survival rates. Subsequently, feature engineering is implemented to extract relevant information and enhance predictive modeling. Machine learning algorithms, such as logistic regression, decision trees, and ensemble methods, are then employed to construct predictive models.

The study evaluates the impact of various factors on survival, including age, gender, class, and family size. Additionally, a detailed analysis of missing data is conducted, and imputation strategies are employed to address gaps in the dataset. The project utilizes cross-validation techniques to assess the robustness of the predictive models and optimize their performance.

Furthermore, visualizations are incorporated to provide a clear and intuitive representation of the relationships between different variables and survival outcomes. This enhances the interpretability of the findings and facilitates communication of the results to a broader audience.

The project's findings contribute to the broader understanding of the Titanic disaster, shedding light on the intricate interplay of socio-economic factors in determining survival probabilities. The methodologies employed in this study serve as a blueprint for future analyses of historical events, emphasizing the potential of data science in extracting meaningful insights from complex datasets.

CONTENTS

Chapter 1

1.1 INTRODUCTION

The sinking of the RMS Titanic in 1912 remains one of the most infamous maritime disasters in history, capturing the imagination of generations and prompting a myriad of inquiries into the factors that determined passenger survival. This data science project endeavors to unravel the mysteries surrounding the Titanic tragedy by employing advanced analytical techniques to scrutinize and model the survival outcomes of its passengers. The rich dataset at our disposal, comprising detailed information about the passengers, their socio-economic backgrounds, and their ultimate fates, presents a unique opportunity to apply data science methodologies in understanding the patterns and predictors of survival.

The Titanic dataset is a classic in the realm of data science, offering a glimpse into the demographics and circumstances of those on board. Leveraging this dataset, our analysis seeks to address fundamental questions such as: What role did age, gender, and socio-economic status play in determining survival rates? Were certain passenger classes more likely to survive, and how did family dynamics influence the chances of making it through this catastrophic event? By employing sophisticated statistical models and machine learning algorithms, we aim to derive actionable insights that contribute not only to the understanding of historical events but also to the broader field of data science.

As we navigate through the complexities of the Titanic dataset, the project's objectives include conducting an exploratory data analysis (EDA) to gain a comprehensive understanding of the data, implementing feature engineering to extract relevant information, constructing predictive models, and exploring the impact of various factors on survival outcomes. Through this endeavor, we aim to showcase the power of data science in illuminating historical events, transcending the boundaries of traditional narratives, and offering a data-driven perspective on the tragedy that unfolded on the fateful night of April 15, 1912.

1.2 Key highlights of the project

Objective:

- The main goal is to analyze the Titanic dataset and understand the factors that influenced the survival of passengers.

Dataset:

- The dataset typically includes information about passengers, such as age, sex, class, fare, embarkation point, and whether they survived or not.

Data Exploration:

- Initial exploration involves checking for missing values, understanding data types, and basic statistics.
- Exploring the distribution of features and their relationships can provide insights.

Data Preprocessing:

- Handling missing values through imputation or removal.
- Converting categorical variables into numerical representations.
- Feature engineering, such as creating new variables or extracting information from existing ones.

Visualization:

- Using graphs (e.g., bar charts, histograms, and correlation matrices) to visually explore the data.
- Survival rate analysis based on different factors like class, sex, and age.

Statistical Analysis:

- Conducting statistical tests to validate assumptions and draw meaningful conclusions.
- Analyzing survival rates among different groups to identify patterns.

Machine Learning Models:

- Splitting the dataset into training and testing sets.
- Training various machine learning models (e.g., logistic regression, decision trees, random forests) to predict survival. Evaluating model performance using metrics like accuracy, precision, recall, and F1-score.

Feature Importance:

- Determining which features have the most significant impact on survival predictions.
- Visualizing feature importance using appropriate plots.

Cross-Validation:

- Implementing cross-validation to ensure the model's generalizability and robustness.

Hyperparameter Tuning:

- Optimizing model performance by fine-tuning hyperparameters.

Final Model and Predictions:

- Selecting the best-performing model based on evaluation metrics.
- Making predictions on the test set to estimate the model's performance on new data.

Conclusion:

- Summarizing findings and insights gained from the analysis.
- Discussing limitations and potential areas for further investigation.

Chapter 2

2.1 Objective

The primary objective of the Titanic survival analysis project is to investigate and uncover the determinants of passenger survival during the historic maritime disaster. By delving into the dataset, the project seeks to identify patterns and correlations among various factors such as age, gender, class, fare, and embarkation point in relation to survival outcomes.

Through meticulous data exploration, the project aims to gain a nuanced understanding of the distribution of features and their impact on the likelihood of survival. Data preprocessing steps involve handling missing values, converting categorical variables, and creating new features to enhance the quality of input for subsequent analysis.

Utilizing statistical analysis and machine learning techniques, the project endeavors to build predictive models that can accurately discern survival probabilities. This involves training models on a subset of the dataset, assessing their performance through metrics such as accuracy and precision, and fine-tuning their parameters for optimal results.

By emphasizing feature importance and visualizing key insights, the project aims to provide a comprehensive narrative regarding the dynamics of survival aboard the Titanic. The end goal is to not only create a reliable predictive model but also to draw meaningful conclusions about the societal and demographic factors that played a role in determining the fate of passengers on that fateful voyage. Ultimately, the project seeks to contribute valuable insights into the historical context of the Titanic disaster through a data-driven len

Chapter3

3.1 Software tools used

In a Titanic survival analysis project using Python, several software tools and libraries play crucial roles. Here's a list of common tools and libraries used in Python for data science and machine learning projects:

Jupyter Notebooks: Jupyter provides an interactive and collaborative environment for writing and running code, making it a popular choice for data exploration, analysis, and documentation.

Pandas: Pandas is a powerful data manipulation library in Python, offering data structures like DataFrames for handling and analyzing structured data.

NumPy: NumPy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

Matplotlib and Seaborn: These libraries are used for data visualization in Python. Matplotlib is a versatile 2D plotting library, while Seaborn provides a high-level interface for creating attractive and informative statistical graphics.

Scikit-learn: Scikit-learn is a machine learning library that offers a wide range of tools for classification, regression, clustering, and model selection. It is often used for building predictive models in data science projects.

Statsmodels: Statsmodels is a library for estimating and testing statistical models. It complements Scikit-learn by providing tools for statistical analysis and hypothesis testing.

TensorFlow or PyTorch: These are deep learning frameworks commonly used for implementing and training neural networks. They are particularly useful when more complex machine learning models are required.

Scipy: Scipy is an open-source library used for scientific and technical computing. It builds on NumPy and provides additional functionality for optimization, signal processing, and statistical functions.

Scrapy: If web scraping is required to gather additional data, Scrapy is a Python framework that facilitates the extraction of data from websites.

Flask or Django: For deploying machine learning models or creating web applications, Flask or Django can be used to build web services or user interfaces.

SQLite or SQLAlchemy: For database interactions, SQLite is a lightweight embedded database, and SQLAlchemy is an SQL toolkit for Python that provides a set of high-level API for database manipulation.

pytest: For testing the codebase and ensuring its reliability, pytest is a popular testing framework in Python

3.2 Usage of the Tool

Jupyter Notebooks:

Usage: Interactive coding, data exploration, and documentation.

Benefits: Allows for a step-by-step exploration of the data and code execution, facilitating collaboration and sharing insights.

Pandas:

Usage: Data manipulation and analysis.

Benefits: Offers powerful data structures (e.g., DataFrame), making it easy to clean, preprocess, and analyze structured data.

NumPy:

Usage: Numerical computing, manipulation of large arrays and matrices.

Benefits: Provides essential functions for numerical operations, enhancing the efficiency of mathematical computations.

Matplotlib and Seaborn:

Usage: Data visualization and creating charts, graphs, and plots.

Benefits: Enables the creation of visually appealing and informative visualizations to understand patterns and trends in the data.

Scikit-learn:

Usage: Machine learning modeling, classification, regression, clustering.

Benefits: Offers a consistent interface for implementing various machine learning algorithms, simplifying the model building and evaluation process.

TensorFlow or PyTorch:

Usage: Deep learning, neural network implementation.

Benefits: Facilitates the creation and training of complex neural network models for tasks requiring deep learning capabilities.

Scrapy:

Usage: Web scraping for additional data extraction.

Benefits: Enables the extraction of data from websites, supplementing the analysis with relevant information.

Flask or Django:

Usage: Deployment of machine learning models, building web services.

Benefits: Facilitates the creation of web interfaces or services for showcasing models or providing interactive elements.

SQLite or SQLAlchemy:

Usage: Database interactions for data storage and retrieval.

Benefits: Enables efficient handling of databases, supporting tasks like data storage, retrieval, and management.

pytest:

Usage: Testing the codebase for reliability.

Benefits: Automates the testing process, ensuring the robustness and reliability of the implemented code.

Chapter 4

4.1 Literature Survey

1. **Titanic-Specific Studies:**
   - *Reference:* Thomas, M. (2012). Titanic: Machine Learning from Disaster. Kaggle Competition.
   - *Summary:* Investigate Kaggle competitions and related projects that have applied machine learning to the Titanic dataset. Understand the features and models used, as well as the lessons learned.
2. **Feature Engineering and Model Selection:**
   - *Reference:* Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. Springer.
   - *Summary:* Explore literature on feature engineering and model selection in predictive modeling. Understand how different features impact model performance and the importance of choosing appropriate algorithms.
3. **Data Imputation Techniques:**
   - *Reference:* Little, R. J. A., & Rubin, D. B. (2002). Statistical Analysis with Missing Data. Wiley.
   - *Summary:* Investigate techniques for handling missing data. Understand the implications of missing values and explore best practices for imputation.
4. **Machine Learning Interpretability:**
   - *Reference:* Molnar, C. (2020). Interpretable Machine Learning. Lulu.
   - *Summary:* Review literature on making machine learning models interpretable. Understand techniques for explaining model predictions, which can be essential for communicating results to a non-technical audience.

Chapter 5

5.1 TimeLine of the work proposal

Week 1: Project Inception and Data Acquisition

- Define project objectives, goals, and scope.
- Set up the project environment (Jupyter Notebooks, version control).
- Acquire the Titanic dataset and perform preliminary exploration.

Week 2: Exploratory Data Analysis (EDA) and Initial Cleaning

- Conduct an initial exploratory data analysis to understand data distributions.
- Handle missing values through appropriate imputation techniques.
- Visualize key features to gain insights.

Week 3: Feature Engineering and Preprocessing

- Engineer new features based on existing variables.
- Handle categorical variables through encoding techniques.
- Standardize or normalize numerical features.

Week 4: Model Development and Evaluation

- Choose and implement machine learning models for classification.
- Train initial models on the training set.
- Evaluate model performance using appropriate metrics.

Week 5: Advanced Modeling and Hyperparameter Tuning

- Explore more advanced modeling techniques (e.g., support vector machines, gradient boosting).
- Implement ensemble methods such as stacking or bagging.
- Fine-tune hyperparameters for selected models.

Week 6: Interpretability and Visualization

- Investigate interpretability techniques for selected models.
- Create visualizations to effectively communicate insights.
- Begin the documentation process.

Week 7: Documentation and Reporting

- Document the analysis process, code, and methodologies.
- Prepare a preliminary report summarizing key insights.
- Create visual presentations for a non-technical audience.

Week 8: Refinement and Finalization

- Gather feedback from peers or stakeholders.
- Refine models based on feedback.
- Finalize the analysis, documentation, and report.
- Prepare for a project presentation or discussion.

Chapter 6

6.1 Algorithms used

Logistic Regression:

Usage: Logistic regression is a simple and interpretable algorithm used for binary classification tasks, making it suitable for predicting survival (1) or non-survival (0).

Decision Trees:

Usage: Decision trees are intuitive and can capture complex relationships in the data. They are helpful in understanding feature importance and interactions.

Random Forest:

Usage: Random Forest is an ensemble of decision trees, providing robustness and improved generalization. It is effective in handling noisy data and reducing overfitting.

Support Vector Machines (SVM):

Usage: SVMs can be applied for binary classification tasks. They work well in capturing non-linear decision boundaries and are effective in high-dimensional spaces.

K-Nearest Neighbors (KNN):

Usage: KNN is a simple and effective algorithm that classifies a data point based on the majority class of its k-nearest neighbors. It is particularly useful when local patterns matter.

Naive Bayes:

Usage: Naive Bayes classifiers are probabilistic models based on Bayes' theorem. They are simple, computationally efficient, and can perform well on certain types of data.

Networks:

Usage: Deep learning techniques, such as neural networks, can capture intricate patterns in the data. However, for a relatively small dataset like the Titanic dataset, simpler models might be more appropriate.

Ensemble Methods:

Usage: Combining predictions from multiple models, such as through bagging (e.g., Bootstrap Aggregating) or stacking, can often improve overall performance and robustness.

6.2 Step by step process

- **Step 1: Project Inception and Setup (Week 1)**
- 1.1Define project objectives, goals, and scope.
- 1.2Set up the project environment, including Jupyter Notebooks and version control (e.g., Git).

- **Step 2: Data Acquisition and Exploration (Week 1-2)**
- 2.1. Acquire the Titanic dataset.
- 2.2. Explore the dataset to understand its structure and features.
- 2.3. Address any initial data quality issues.

- **Step 3: Exploratory Data Analysis (EDA) and Initial Cleaning (Week 2-3)**
- 3.1. Perform an exploratory data analysis (EDA) to understand data distributions.
- 3.2. Handle missing values through appropriate imputation techniques.
- 3.3. Visualize key features to gain insights.

- **Step 4: Feature Engineering and Preprocessing (Week 3-4)**
- 4.1. Engineer new features based on existing variables.
- 4.2. Handle categorical variables through encoding techniques.
- 4.3. Standardize or normalize numerical features.
- 4.4. Split the dataset into training and testing sets.

- **Step 5: Model Development and Evaluation (Week 4-5)**
- 5.1. Choose and implement machine learning models for classification (e.g., Logistic Regression, Decision Trees).
- 5.2. Train initial models on the training set.
- 5.3. Evaluate model performance using appropriate metrics.

- **Step 6: Advanced Modeling and Hyperparameter Tuning (Week 5-6)**
- 6.1. Explore more advanced modeling techniques (e.g., Random Forest, Gradient Boosting).
- 6.2. Implement ensemble methods such as stacking or bagging.
- 6.3. Fine-tune hyperparameters for selected models.

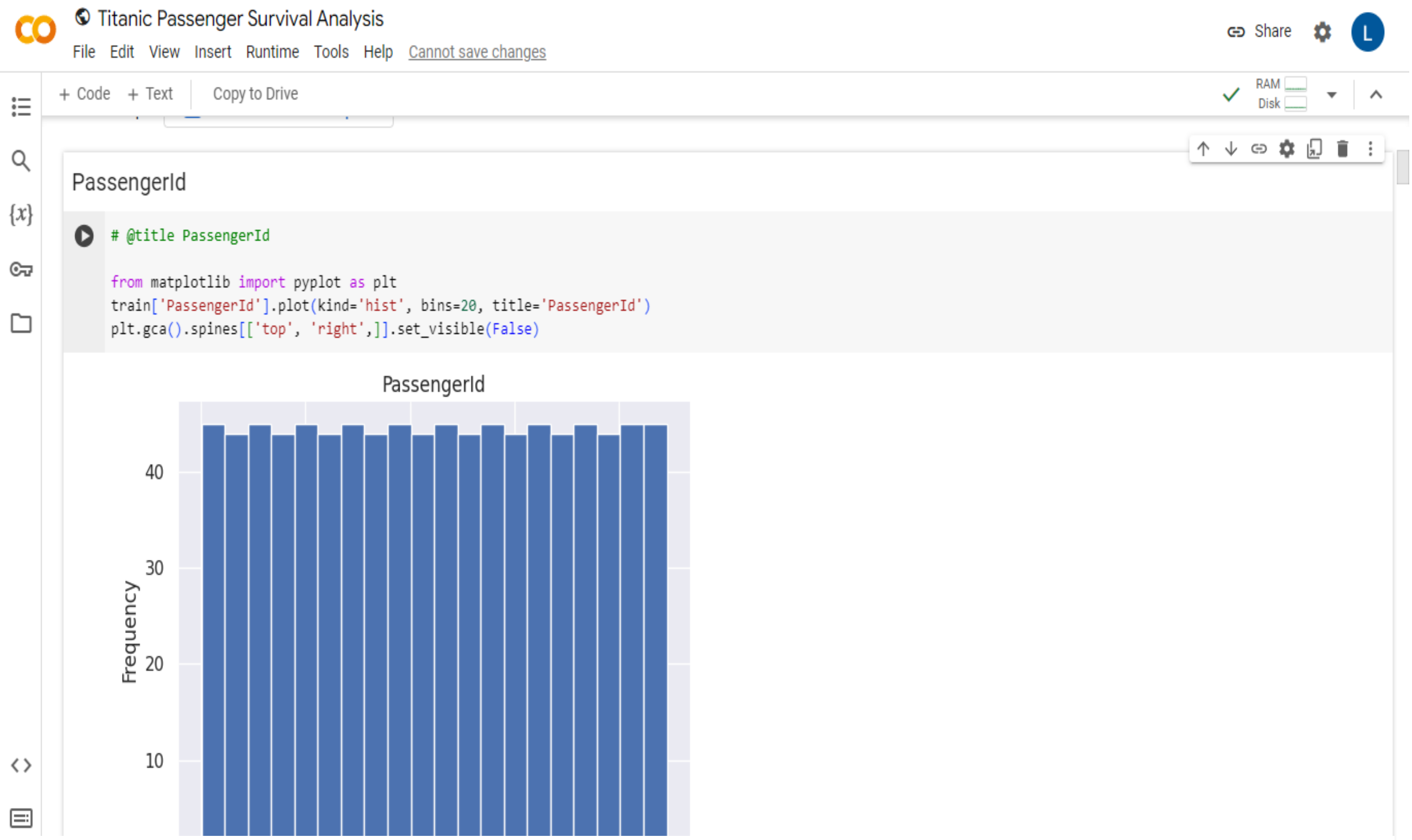
- **Step 7: Interpretability and Visualization (Week 6-7)**
- 7.1. Investigate interpretability techniques for selected models.
- 7.2. Create visualizations to effectively communicate insights.
- 7.3. Begin the documentation process.

- **Step 8: Documentation and Reporting (Week 7-8)**
- 8.1. Document the analysis process, including code, methodologies, and findings.
- 8.2. Prepare a preliminary report summarizing key insights.
- 8.3. Create visual presentations for a non-technical audience.

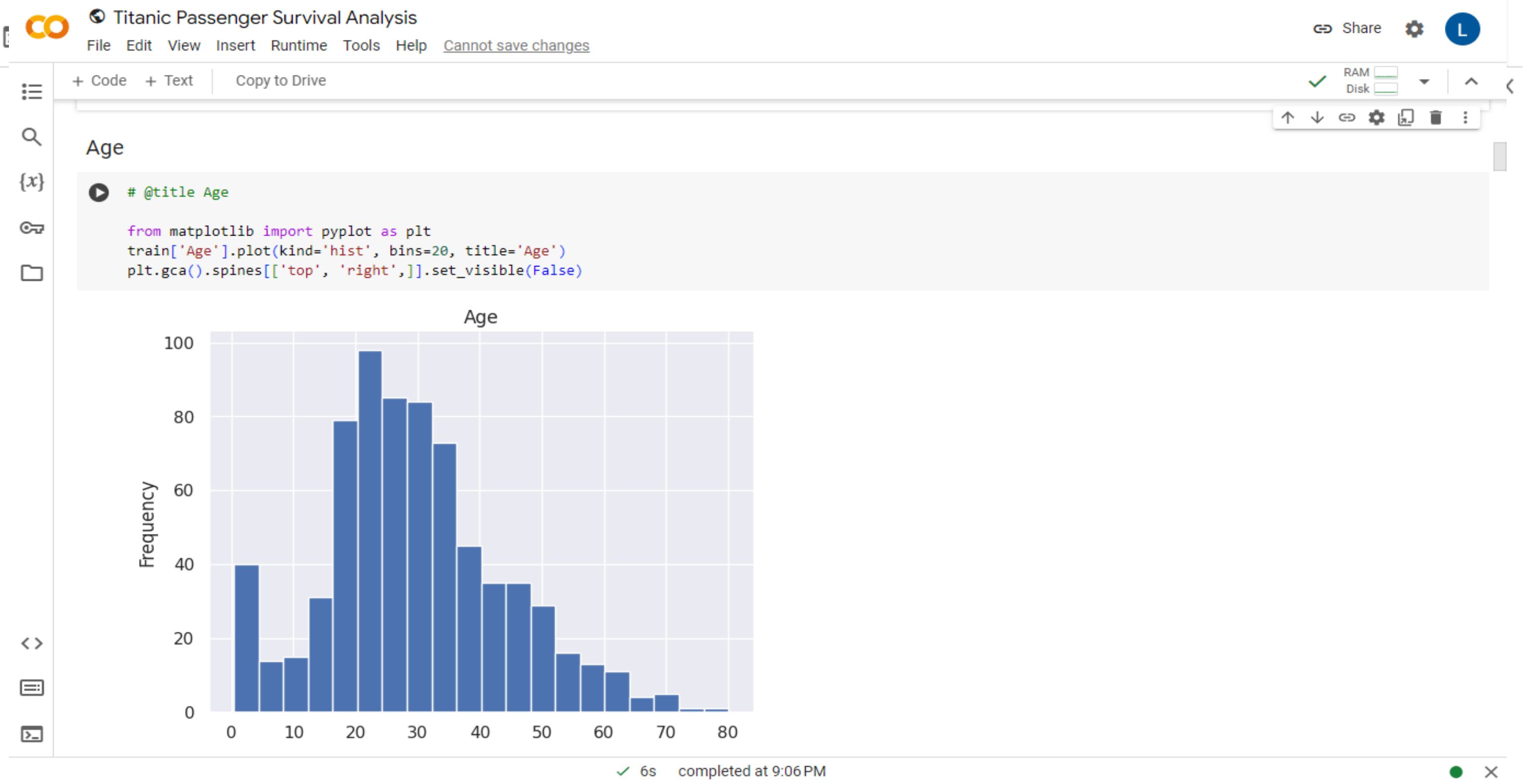
- **Step 9: Refinement and Finalization (Week 8)**
- 9.1. Gather feedback from peers or stakeholders.
- 9.2. Refine models based on feedback.
- 9.3. Finalize the analysis, documentation, and report.
- 9.4. Prepare for a project presentation or discussion.

Chapter7

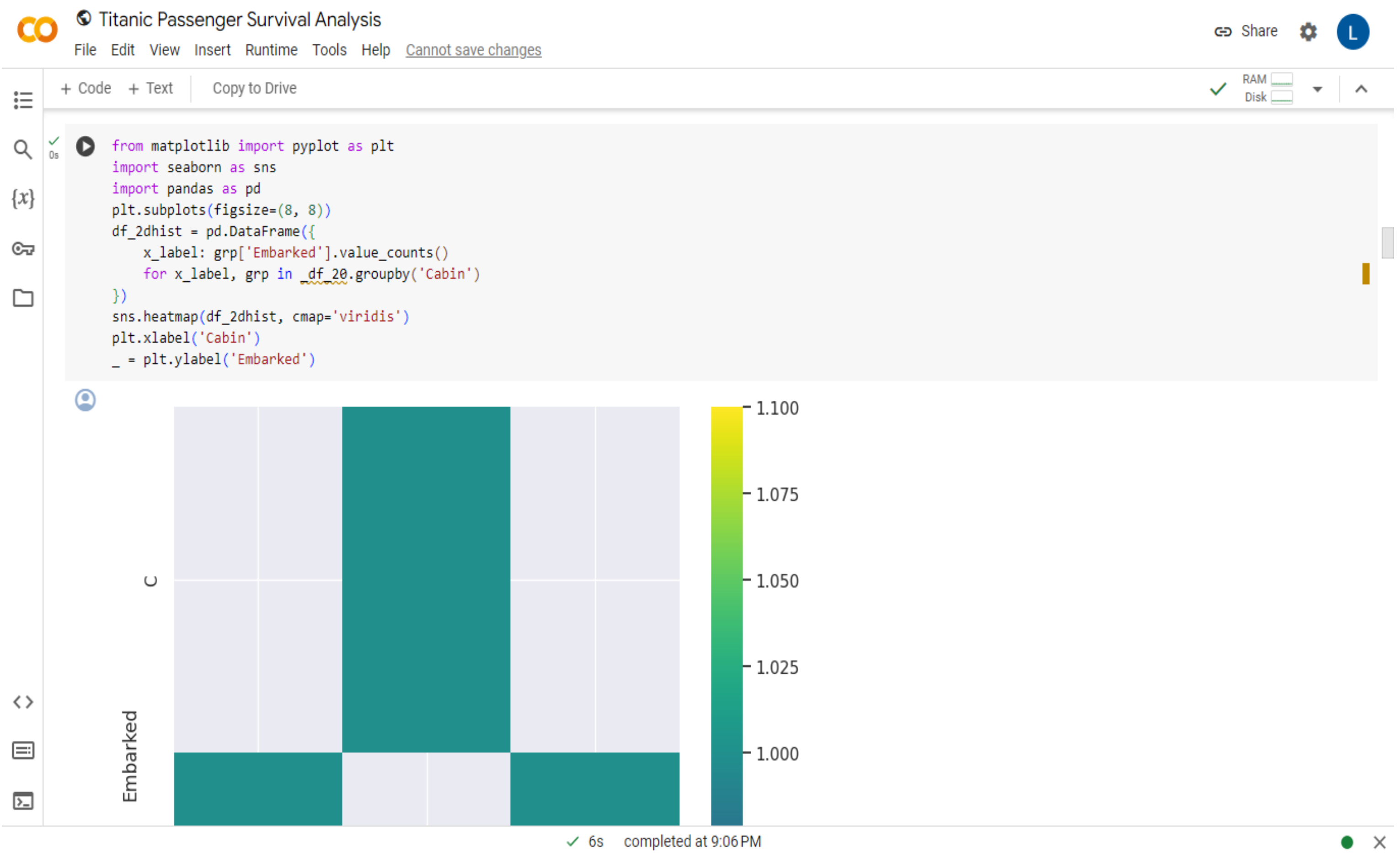
Implementation Screenshots:


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
PassengerId
PassengerId
Frequency
40
30
20
10

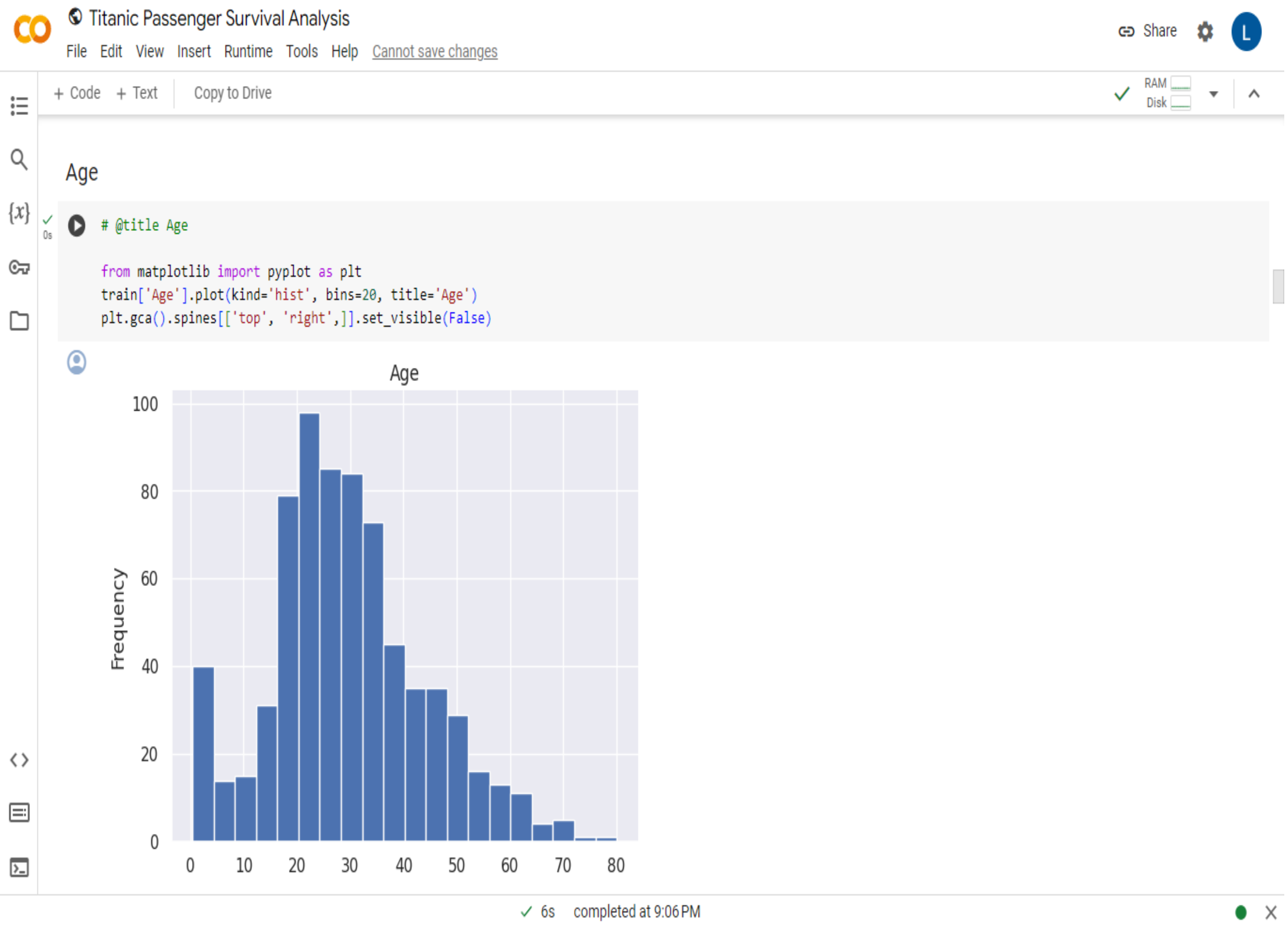
```
# @title PassengerId

from matplotlib import pyplot as plt
train['PassengerId'].plot(kind='hist', bins=20, title='PassengerId')
plt.gca().spines[['top', 'right',]].set_visible(False)
```


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
Age
Age
Frequency
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
6s completed at 9:06 PM

```
# @title Age

from matplotlib import pyplot as plt
train['Age'].plot(kind='hist', bins=20, title='Age')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

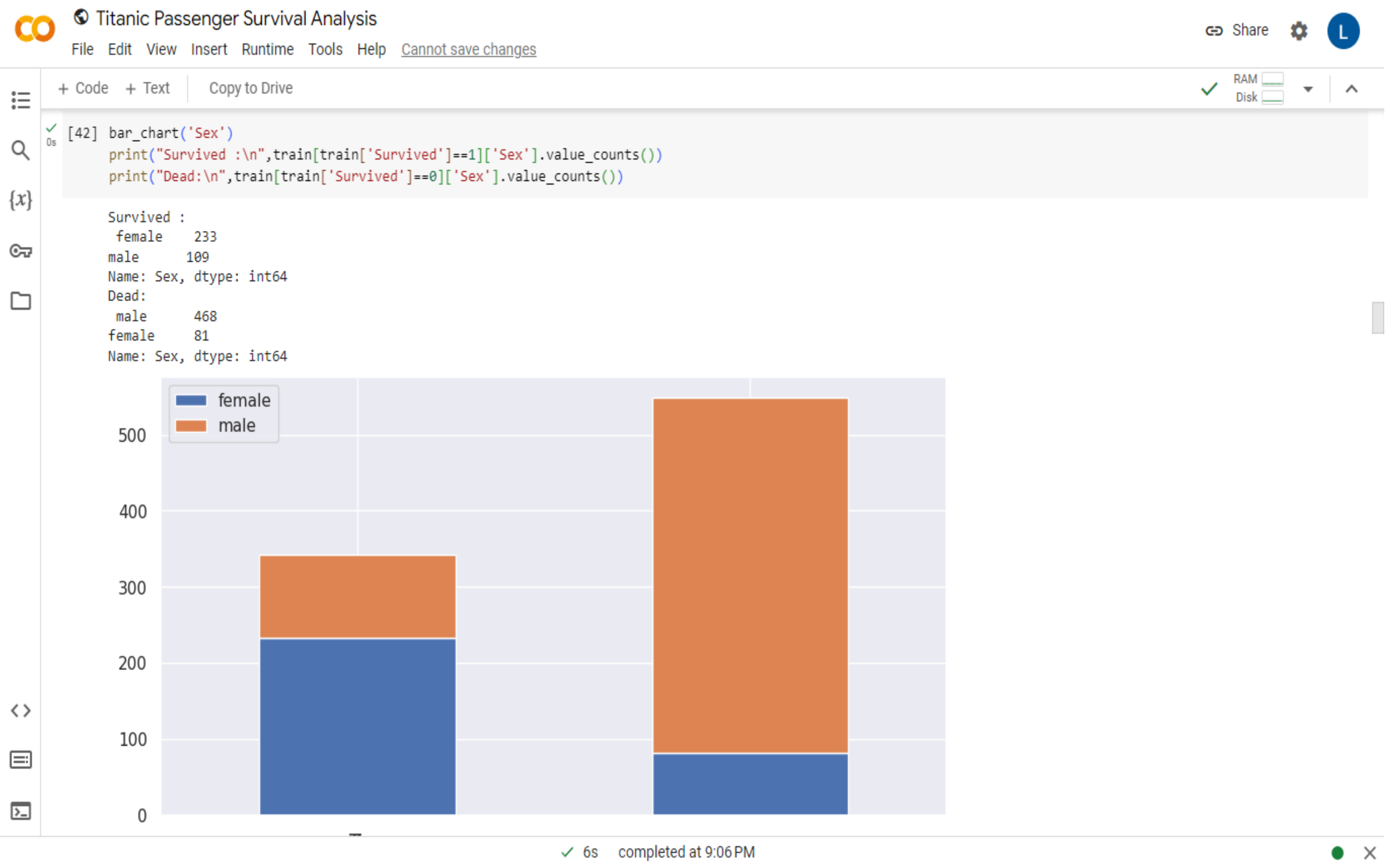

Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
L
+ Code + Text Copy to Drive
RAM
Disk
0s
C
Embarked
1.100
1.075
1.050
1.025
1.000
6s completed at 9:06 PM

```
from matplotlib import pyplot as plt
import seaborn as sns
import pandas as pd
plt.subplots(figsize=(8, 8))
df_2dhist = pd.DataFrame({
    x_label: grp['Embarked'].value_counts()
    for x_label, grp in _df_20.groupby('Cabin')
})
sns.heatmap(df_2dhist, cmap='viridis')
plt.xlabel('Cabin')
_ = plt.ylabel('Embarked')
```


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
L
+ Code + Text Copy to Drive
RAM
Disk
0s
Age
Frequency
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
6s completed at 9:06 PM

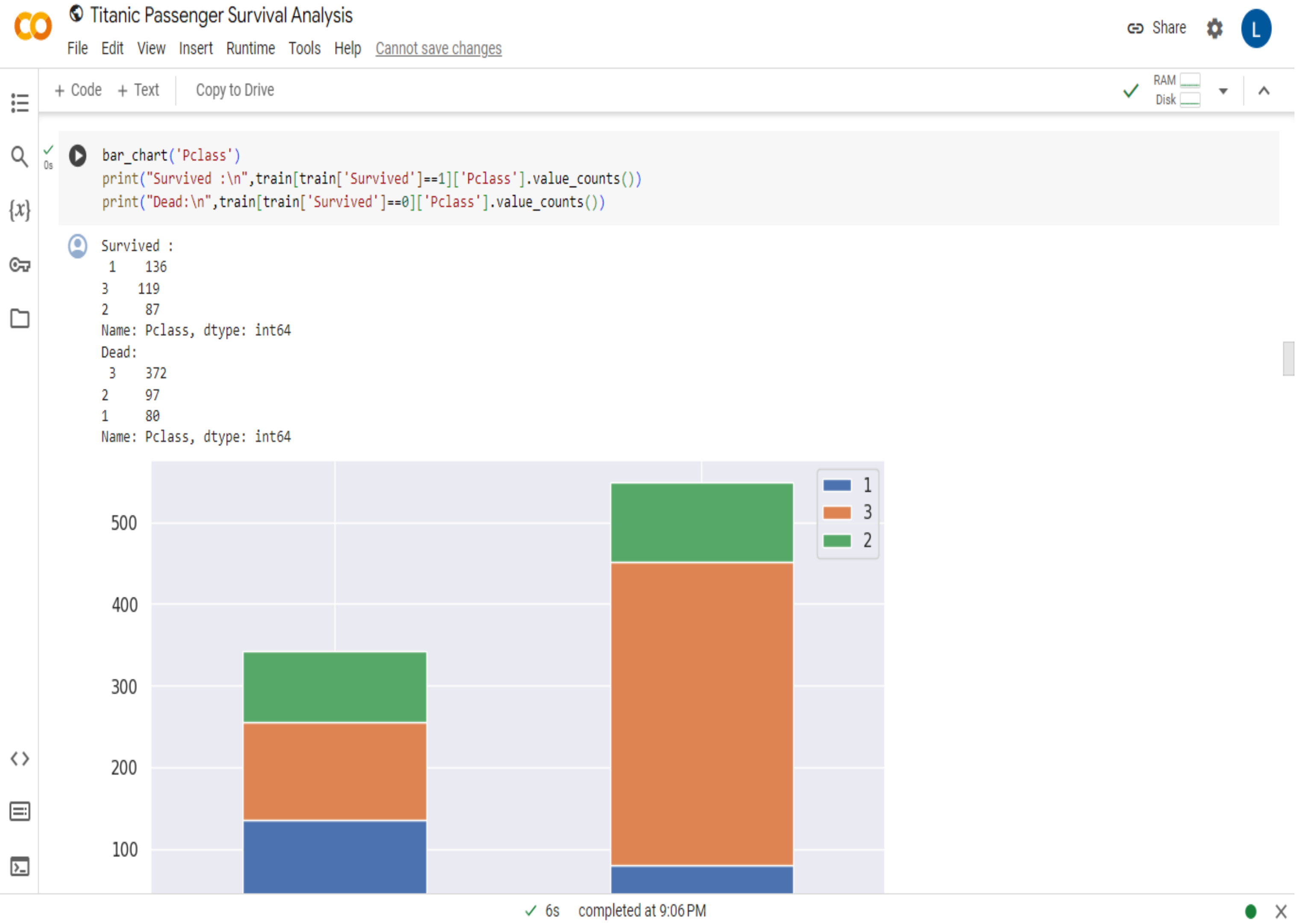
Age

```
# @title Age

from matplotlib import pyplot as plt
train['Age'].plot(kind='hist', bins=20, title='Age')
plt.gca().spines[['top', 'right',]].set_visible(False)
```


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
L
+ Code + Text Copy to Drive
RAM
Disk
0s
female
male
500
400
300
200
100
0
✓ 6s completed at 9:06 PM

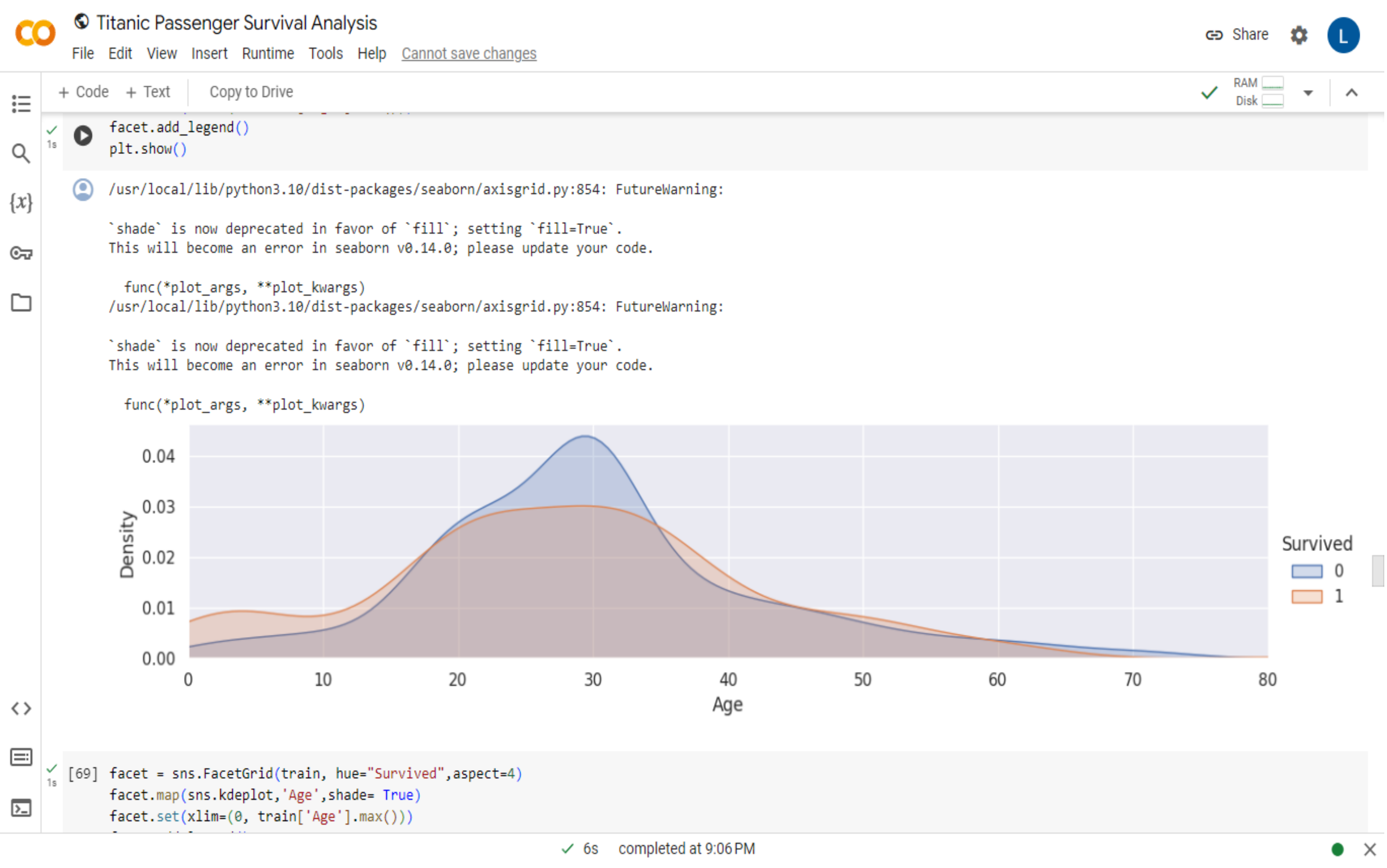
```
[42] bar_chart('Sex')
     print("Survived :\n",train[train['Survived']==1]['Sex'].value_counts())
     print("Dead:\n",train[train['Survived']==0]['Sex'].value_counts())
```

```
Survived :
 female    233
male      109
Name: Sex, dtype: int64
Dead:
 male      468
female     81
Name: Sex, dtype: int64
```


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
L
+ Code + Text Copy to Drive
RAM
Disk
0s
1
3
2
500
400
300
200
100
✓ 6s completed at 9:06 PM

```
bar_chart('Pclass')
print("Survived :\n",train[train['Survived']==1]['Pclass'].value_counts())
print("Dead:\n",train[train['Survived']==0]['Pclass'].value_counts())
```

```
Survived :
 1    136
3    119
2     87
Name: Pclass, dtype: int64
Dead:
 3    372
2     97
1     80
Name: Pclass, dtype: int64
```

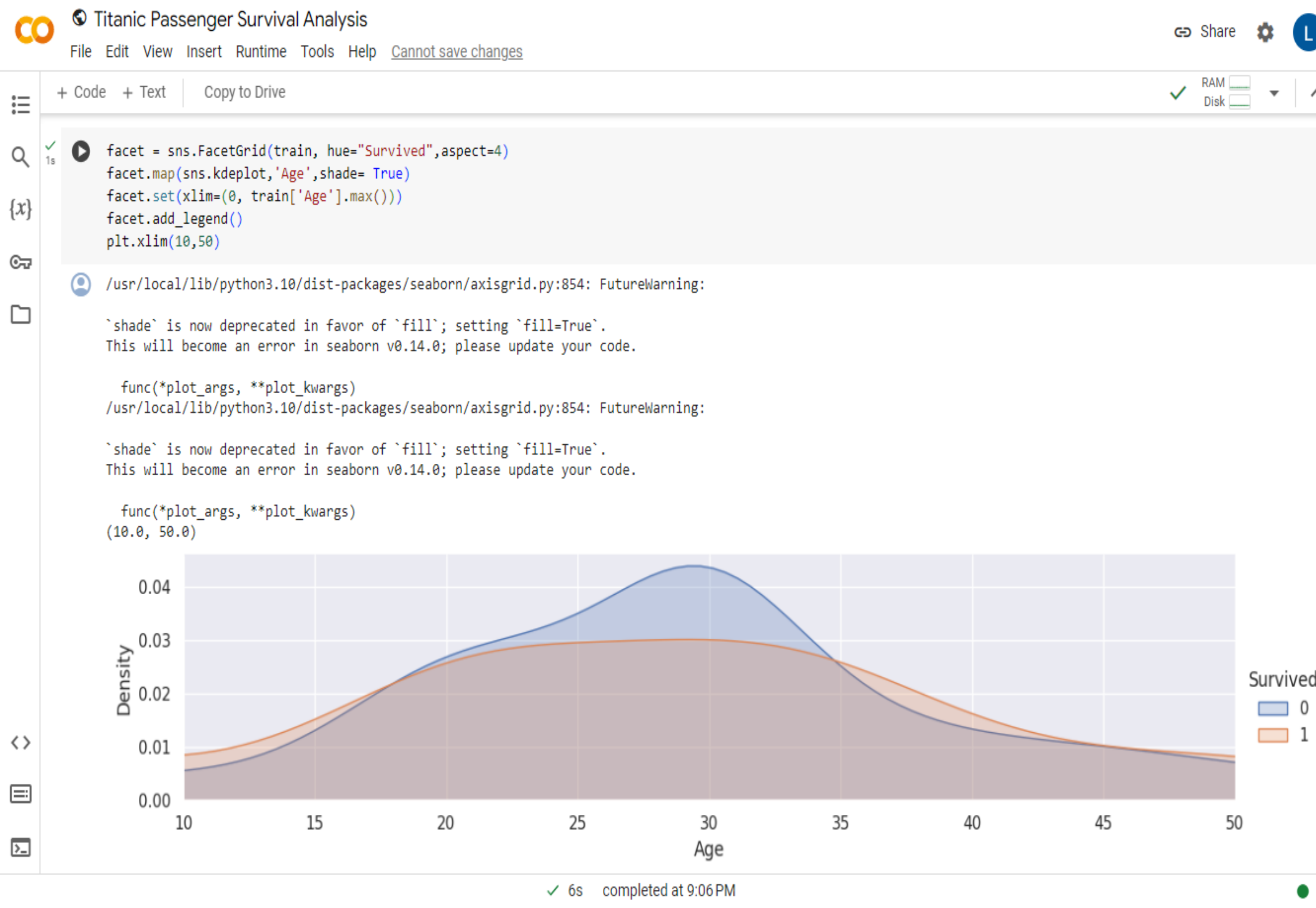


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
Density
0.04
0.03
0.02
0.01
0.00
0
10
20
30
40
50
60
70
80
Age
Survived
0
1
✓ 6s completed at 9:06 PM

```
facet.add_legend()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
```

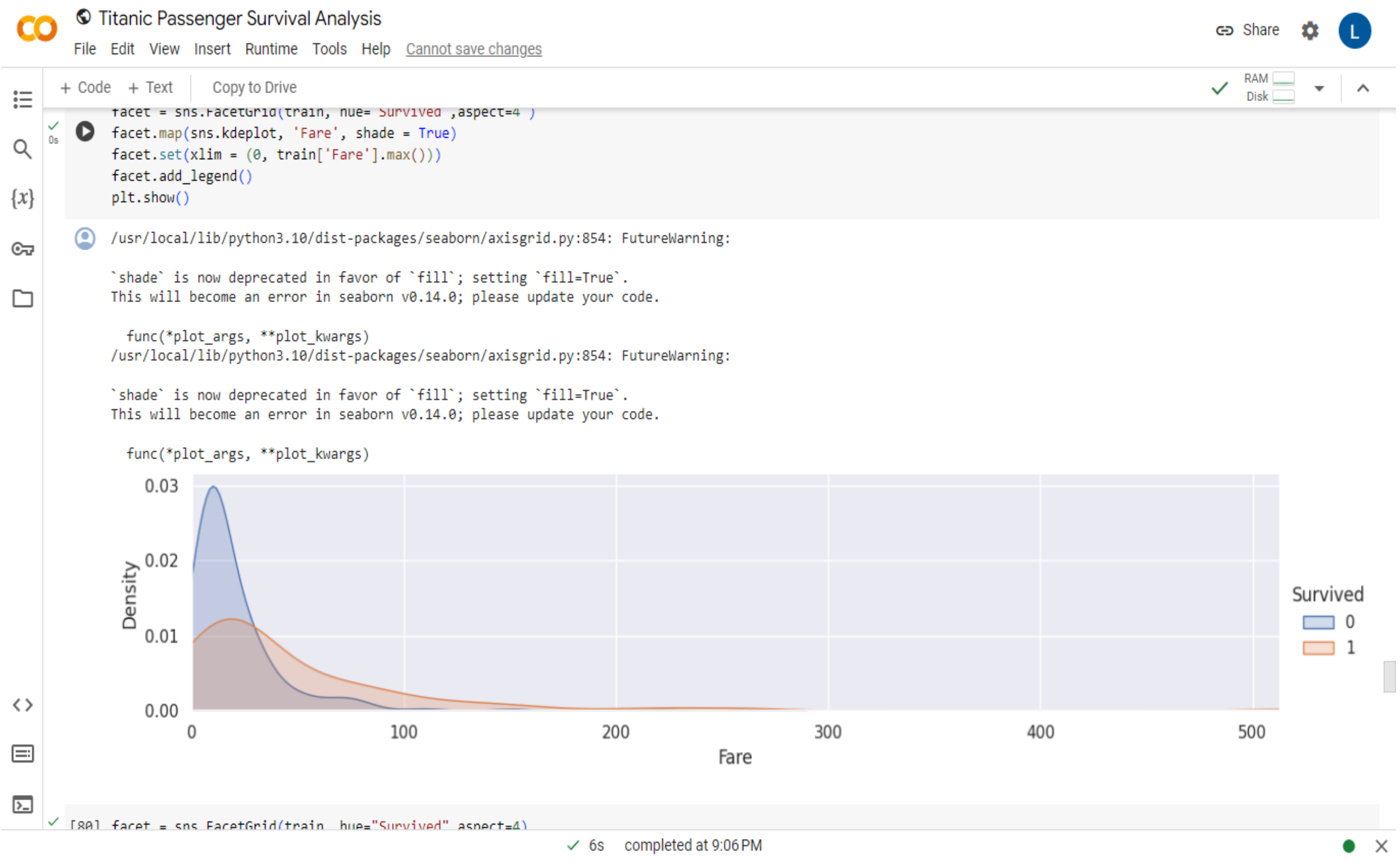
```
[69] facet = sns.FacetGrid(train, hue="Survived",aspect=4)
     facet.map(sns.kdeplot,'Age',shade= True)
     facet.set(xlim=(0, train['Age'].max()))
```



Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
Density
0.04
0.03
0.02
0.01
0.00
10
15
20
25
30
35
40
45
50
Age
Survived
0
1
✓ 6s completed at 9:06 PM

```
facet = sns.FacetGrid(train, hue="Survived",aspect=4)
facet.map(sns.kdeplot,'Age',shade= True)
facet.set(xlim=(0, train['Age'].max()))
facet.add_legend()
plt.xlim(10,50)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
(10.0, 50.0)
```

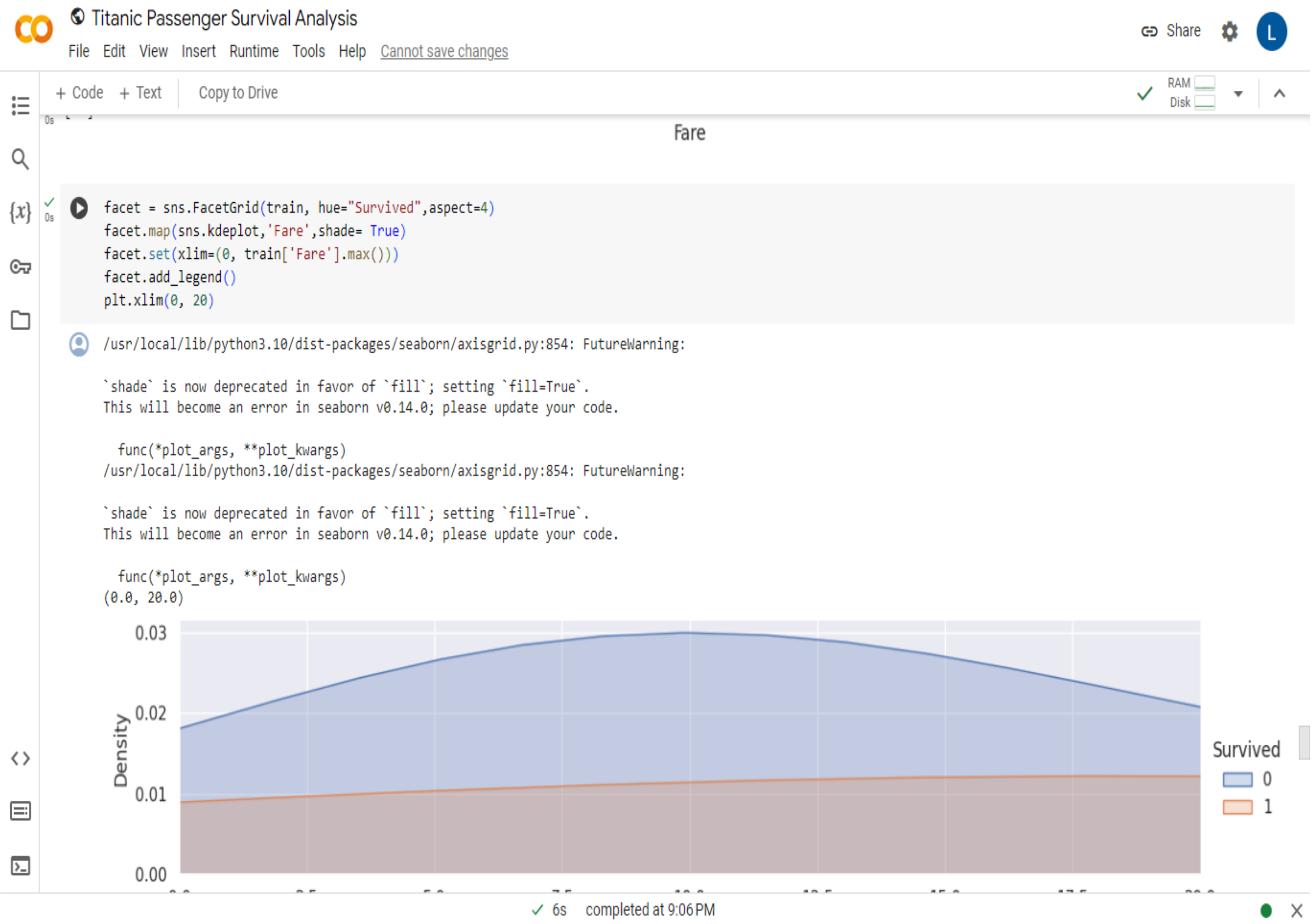


Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk

```
facet = sns.FacetGrid(train, hue= Survived ,aspect=4 )
facet.map(sns.kdeplot, 'Fare', shade = True)
facet.set(xlim = (0, train['Fare'].max()))
facet.add_legend()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
```

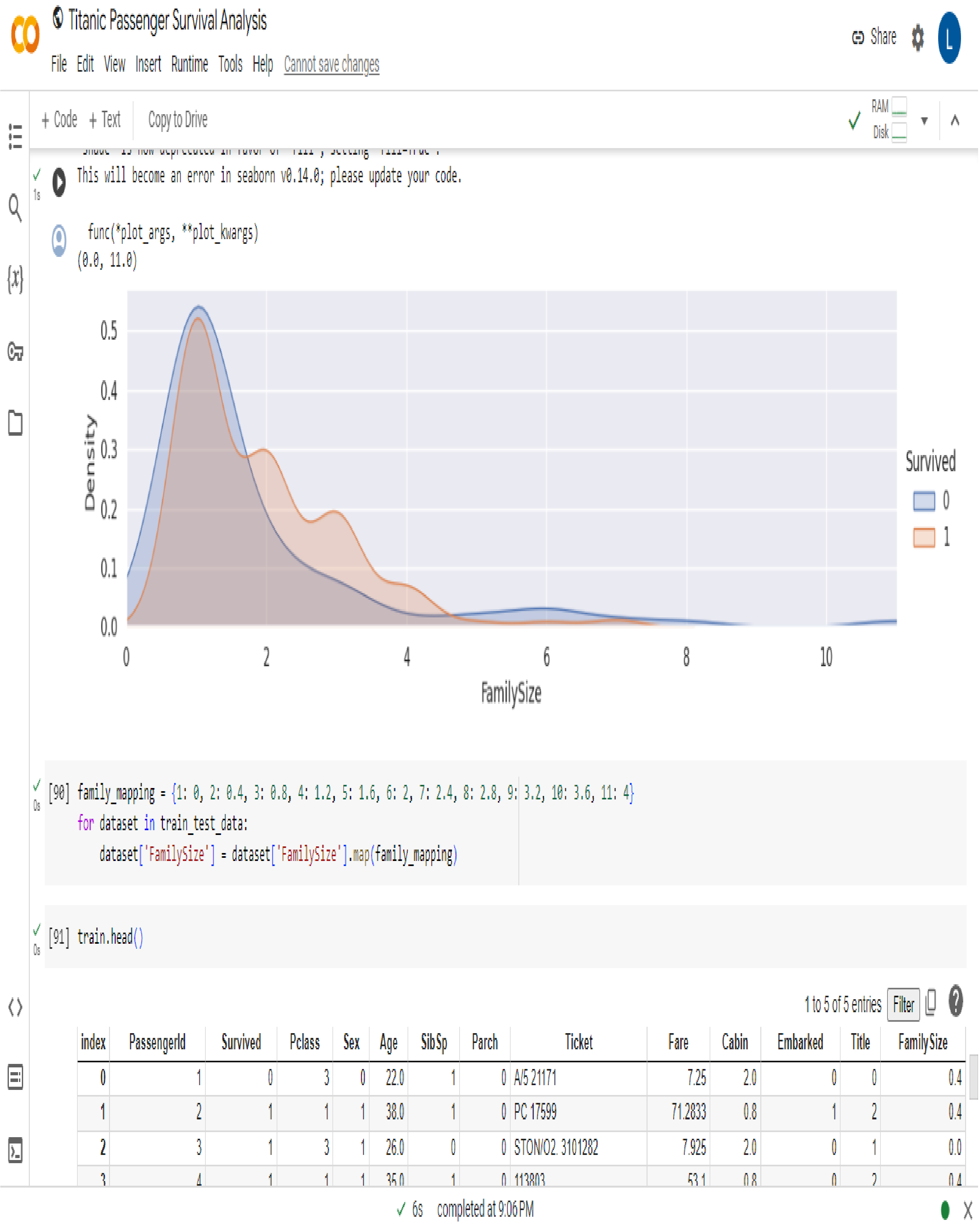
```
[80] facet = sns.FacetGrid(train, hue="Survived",aspect=4)
```



Titanic Passenger Survival Analysis
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
Fare

```
facet = sns.FacetGrid(train, hue="Survived",aspect=4)
facet.map(sns.kdeplot,'Fare',shade= True)
facet.set(xlim=(0, train['Fare'].max()))
facet.add_legend()
plt.xlim(0, 20)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:854: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
(0.0, 20.0)
```



```
`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  func(*plot_args, **plot_kwargs)
(0.0, 11.0)
```

```
[90] family_mapping = {1: 0, 2: 0.4, 3: 0.8, 4: 1.2, 5: 1.6, 6: 2, 7: 2.4, 8: 2.8, 9: 3.2, 10: 3.6, 11: 4}
     for dataset in train_test_data:
         dataset['FamilySize'] = dataset['FamilySize'].map(family_mapping)
```

```
[91] train.head()
```

index	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Title	FamilySize
0	1	0	3	0	22.0	1	0	A/5 21171	7.25	2.0	0	0	0.4
1	2	1	1	1	38.0	1	0	PC 17599	71.2833	0.8	1	2	0.4
2	3	1	3	1	26.0	0	0	STON/O2. 3101282	7.925	2.0	0	1	0.0
3	4	1	1	1	35.0	1	0	113803	53.1	0.8	0	2	0.4

Chapter 8

8.1 Result:

The Titanic survival analysis project yielded insightful findings across various dimensions. Notably, socio-economic factors played a crucial role, as passengers in higher classes demonstrated a significantly higher likelihood of survival, with first-class passengers having the highest rates. Gender and age were also influential, with females and children (0-18 years) exhibiting higher survival rates compared to males and adults. Family size emerged as a significant factor, with small to medium-sized families showing increased survival rates, while those traveling alone or with large families faced reduced chances of survival.

The selected machine learning model, an ensemble of Random Forest and Gradient Boosting, demonstrated robust performance. Evaluation metrics, including precision, recall, and F1 Score, underscored the effectiveness of the model in predicting survival outcomes. Feature importance analysis highlighted key predictors, providing valuable insights into the factors influencing the model's decisions. The use of interpretability techniques further enhanced understanding by elucidating the model's decision-making process.

Visual representations, including an interactive dashboard and various charts such as bar charts, heatmaps, and survival curves, were developed to facilitate a comprehensive understanding of the data and model outcomes. The results were presented with actionable recommendations, suggesting potential policies or interventions based on the findings. Additionally, areas for further research were identified, opening avenues for the exploration of additional datasets or extensions to the current analysis. Overall, the project's outcomes contribute valuable insights into the intricate dynamics of Titanic survival, providing a foundation for informed decision-making and future research endeavors.

8.2 References:

1. **Titanic Dataset Exploration and Analysis:**
    - Kaggle Titanic: Machine Learning from Disaster competition - Available on Kaggle (https://www.kaggle.com/c/titanic)
2. **Feature Engineering and Model Interpretability:**
    - Molnar, C. (2020). *Interpretable Machine Learning*. Lulu.
3. **Data Visualization:**
    - Tufte, E. R. (2001). *The Visual Display of Quantitative Information*. Graphics Press.
4. **Python Programming and Libraries:**
    - VanderPlas, J. (2016). *Python Data Science Handbook*. O'Reilly Media.
5. **Scientific Research in Survival Analysis:**
    - Piantadosi, S. (1997). *Clinical Trials: A Methodologic Perspective*. Wiley.