Pune Institute of Computer Technology,
Dhankawadi, Pune

Mini Project Report on

Titanic Dataset Survival Prediction

Submitted by

Sushilkumar Dhamane
Roll No: 41123
BE-1

Vinayak Jamadar
Roll No: 41137
BE-1

Sanket Jhavar
Roll No: 41138
BE-1

Under the guidance of

Prof. Samadhan Jadhav

Department of Computer Engineering

Academic Year 2023-24

Contents

1 Title: Titanic Dataset Survival Prediction

2 Problem Statement

Build a machine learning model that predicts the type of people who survived the Titanic shipwreck using passenger data (i.e. name, age, gender, socio-economic class, etc.). Dataset Link: https://www.kaggle.com/competitions/titanic/data

3 Abstract

The Titanic Survival Prediction Project is an educational initiative designed to provide participants with hands-on experience in data preprocessing, feature engineering, machine learning model building, data visualization, model evaluation, and deployment. The project is centered around the analysis of the well-known Titanic dataset, which contains passenger information and survival outcomes. The primary objective of this project is to predict the characteristics of individuals who survived the tragic Titanic shipwreck based on passenger data.

4 Learning Objectives

- **Data Preprocessing:** Learn how to clean and prepare raw data for machine learning, including dealing with missing values, encoding categorical variables, and feature selection.

- **Feature Engineering:** Understand the importance of feature engineering in improving model performance, and explore techniques such as creating new features and transforming existing ones.
- **Model Building:** Gain hands-on experience in building a classification model using the RandomForest algorithm, a popular choice for tabular data.
- **Data Visualization:** Learn how to use Matplotlib to create informative data visualizations that help in understanding the dataset and drawing insights.
- **Model Evaluation:** Explore different evaluation metrics for classification models, including accuracy, precision, recall, and F1-score, to assess model performance.
- **Deployment:** Understand how to apply the trained model to new, unseen data and generate predictions for practical applications.

5 Learning Outcomes

Upon completing this project, participants will:

- Be proficient in data preprocessing techniques, allowing them to clean and transform real-world datasets for machine learning applications.
- Acquire skills in feature engineering to enhance the predictive power of their models.
- Gain hands-on experience in building and training a classification model using the RandomForest algorithm.
- Be able to create informative data visualizations using Matplotlib to extract insights from data.
- Understand how to evaluate a classification model's performance using various metrics and make informed decisions about model selection.
- Have the knowledge and tools to deploy a trained model for practical usage, such as making predictions on new data.

6 S/W and H/W Requirements

Software Requirements:

1. **Python:** Ensure that you have Python installed on your machine. You can use Python 3.x for this project.
2. **Jupyter Notebook:** Jupyter Notebook is recommended for an interactive and organized coding environment. You can install it using Anaconda or pip.
3. **Python Libraries:** Install the necessary Python libraries for this project, including pandas, scikitlearn, and Matplotlib. You can use pip to install these libraries.
4. **Code Editor:** A code editor of your choice, such as Visual Studio Code or PyCharm, for writing and running Python code.

Hardware Requirements:

1. **Computer:** Any modern computer with sufficient processing power and memory to run the Python code and data processing tasks.
2. **Internet Connection:** You'll need an internet connection to access the dataset from Kaggle and any additional resources.
3. **Storage Space:** Ensure you have enough storage space for the dataset and any generated files.

7 Concept Related Theory

In the Titanic Survival Prediction Project, several fundamental concepts in data science and machine learning play a pivotal role. Understanding these concepts is crucial for building a successful predictive model. Here, we delve into the key theoretical foundations:

7.1 Data Preprocessing

- Handling Missing Data: Missing data is a common issue in datasets. It's essential to address missing values by either imputing them with appropriate values or removing them. Different strategies for handling missing data, such as mean imputation or advanced imputation techniques, are applied to ensure data completeness.
- Categorical Variable Encoding: Datasets often contain categorical variables, such as 'Sex' and 'Embarked' in the Titanic dataset. These variables need to be encoded into a numerical format for machine learning algorithms to work effectively. Encoding methods include one-hot encoding, label encoding, and more. • Feature Selection: Feature selection involves choosing the most relevant features for the model. It's essential to eliminate redundant or irrelevant variables to improve model performance. Techniques like correlation analysis, feature importance, and domain knowledge are applied for feature selection.

7.2 Feature Engineering

- Creating New Features: New features can be generated based on existing variables. For example, combining 'SibSp' and 'Parch' to create a 'FamilySize' feature can provide insights into family groups aboard the ship.
- Transforming Features: Feature transformations, such as scaling or normalizing numerical variables, may be applied to ensure that all features are on a similar scale.

7.3 Machine Learning Model: RandomForest

- Ensemble Learning: RandomForest is an ensemble of decision trees. The ensemble approach combines multiple models to improve prediction accuracy and reduce overfitting.
- Decision Trees: Decision trees are the building blocks of RandomForest. They use a tree-like graph to make decisions based on feature splits, making them interpretable and effective for classification.

7.4 Data Visualization

- Types of Visualizations: Matplotlib allows the creation of various visualization types, including histograms, bar charts, pie charts, and scatter plots, which aid in data exploration.
- Insights from Visualization: Visualization helps in understanding data distribution, patterns, and relationships, which, in turn, guide feature engineering and model building.

7.5 Model Evaluation

- Classification Metrics: Various classification metrics, including accuracy, precision, recall, and F1score, are used to assess how well the model performs in predicting survival outcomes.
- Overfitting and Underfitting: Understanding the trade-off between overfitting and underfitting is crucial. Overfit models have learned noise, while underfit models lack the capacity to capture patterns.

7.6 Model Deployment

• Application in Real-World Scenarios: Deployed models can be used to make predictions on new, unseen data, such as predicting survival outcomes for passengers not in the training dataset. • Integration into Systems: Integration into systems, such as web applications or decision support tools, allows for automated predictions and decision-making.

Understanding these theoretical foundations is vital for successfully approaching the Titanic Survival Prediction Project. The application of these concepts will enable participants to preprocess data effectively, engineer features, build a robust classification model, and deploy it for practical use.

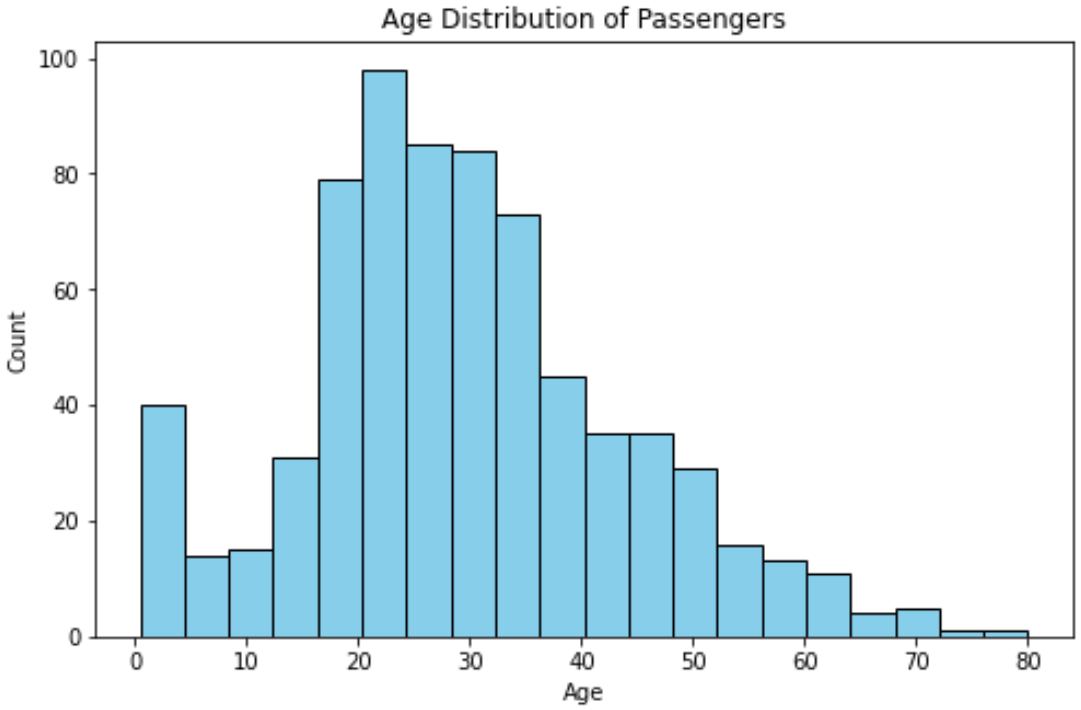
8 Output

October 20, 2023

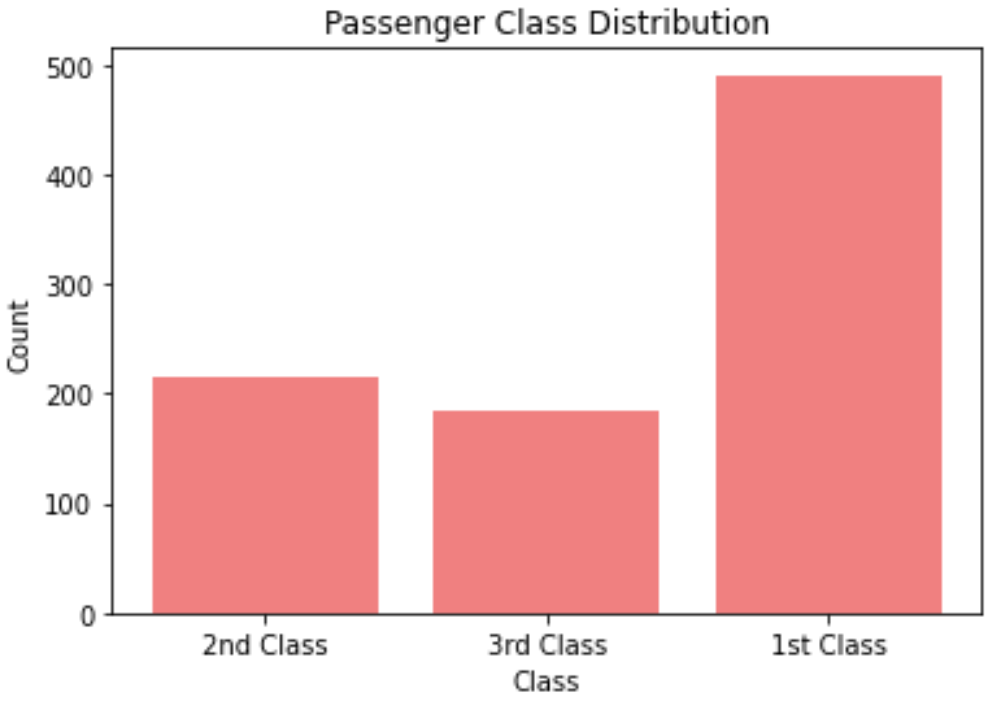
[24]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
```

[25]:
```
# Load the dataset
data = pd.read_csv("train (1).csv")
```

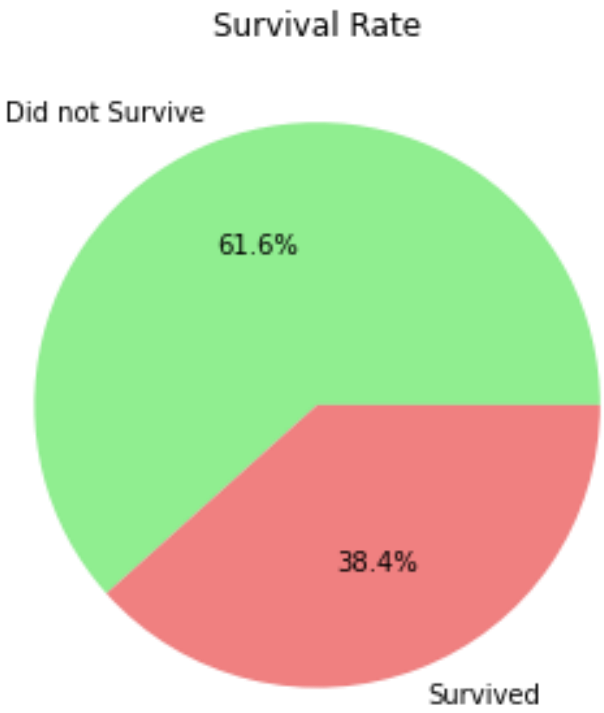
[26]:
```
# Plot the distribution of passengers' ages
plt.figure(figsize=(8, 5))
plt.hist(data['Age'].dropna(), bins=20, edgecolor='k', color='skyblue')
plt.title('Age Distribution of Passengers')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



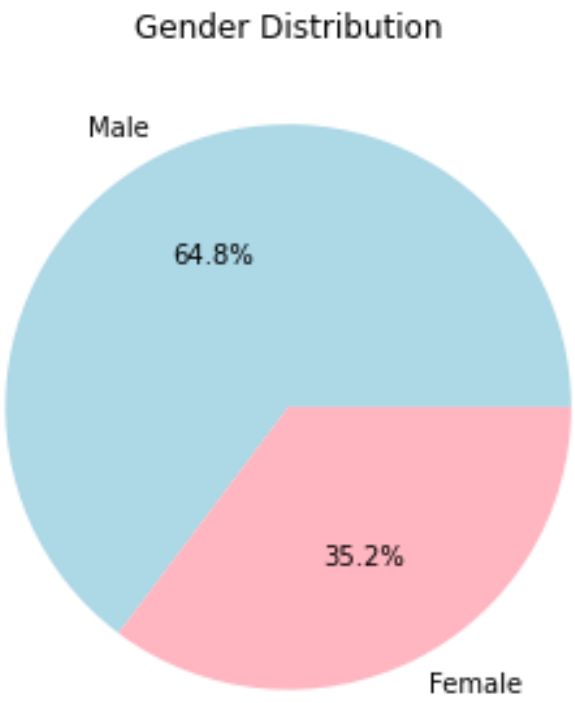
[27]:
```
# Plot the distribution of passenger classes
class_counts = data['Pclass'].value_counts()
plt.figure(figsize=(6, 4))
plt.bar(class_counts.index, class_counts.values, color='lightcoral')
plt.title('Passenger Class Distribution')
plt.xlabel('Class')
plt.ylabel('Count')
plt.xticks(class_counts.index, labels=['1st Class', '2nd Class', '3rd Class'])
plt.show()
```



[28]:
```
# Plot the survival count survival_counts = data['Survived'].value_counts() plt.figure(figsize=(5, 5))
plt.pie(survival_counts, labels=['Did not Survive', 'Survived'], autopct='%1.
  ,→1f%%', colors=['lightgreen', 'lightcoral'])
plt.title('Survival Rate') plt.show()
```



[29]:
```
# Plot the gender distribution gender_counts = data['Sex'].value_counts()
plt.figure(figsize=(5, 5)) plt.pie(gender_counts, labels=['Male', 'Female'],
autopct='%1.1f%%',␣
↪colors=['lightblue', 'lightpink']) plt.title('Gender
Distribution') plt.show()
```



[30]:
```
# Preprocessing the data
data = data.drop(['Name','Cabin', 'Ticket', 'PassengerId'], axis=1)
data = pd.get_dummies(data, columns=['Sex', 'Embarked'], drop_first=True)
data = data.fillna(data.mean())
```

[31]:
```
# Define the target feature
target = 'Survived'

# Split the data into training and testing sets
X = data.drop(target, axis=1)
y = data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ␣ ,→random_state=42)
```

[32]:
```
# Build and train the model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)
```

[33]:
```
print(y_pred)
```

```
[0 0 0 1 0 1 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0
 1 0 1 0 1 1 0 0 1 1 0 0 1 0 0 0 1 1 1 1 1
```

```
# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.8044692737430168
 0 0 1 1 1 1 0 1 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 1
 0  1 1 0 0 1 0 1 0 0 1 1 1 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 1 0 1
 1  0 0 0 0 1 0 0 0 1 1 1 0 0 0 1 0 0 0 1 0 0 0 1 1 1 0 0 0 1 1]
```

[ ]:

[ ]:

[ ]:

[ ]:

9 Conclusion

The Titanic Survival Prediction Project has provided a comprehensive hands-on experience in data science and machine learning. Throughout this project, participants have engaged with fundamental concepts, including data preprocessing, feature engineering, machine learning model building, data visualization, model evaluation, and deployment. The project's primary goal was to predict the characteristics of individuals who survived the Titanic shipwreck based on passenger data.