CAPSTONE PROJECT

TITANIC SURVIVAL PREDICTION

Presented By:
Bhavatharani . S
University College Of Engineering , Villupuram
B.Tech – Information Technology

edunet foundation

OUTLINE

- Problem Statement
- Proposed Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

The Titanic survival prediction problem involves analyzing a dataset containing demographic and trip information of passengers aboard the RMS Titanic. The task is to build a predictive model that accurately determines the likelihood of survival based on features such as age, sex, ticket class, number of siblings/spouses aboard, number of parents/children aboard, fare, and embarkation port. This historical dataset is a classic example in the field of machine learning and data science, aiming to uncover patterns and factors that influenced survival during the tragic sinking of the Titanic in 1912. The ultimate objective is to create a model that can predict with high accuracy whether a given passenger survived or not, thereby shedding light on the socio-economic and personal factors that played a role in survival outcomes.

PROPOSED SOLUTION

1.Data Exploration and Preprocessing:

- Begin by exploring the dataset to understand its structure, missing values, and distribution of features. Perform data preprocessing tasks such as handling missing values, encoding categorical variables, and potentially scaling numerical features.

2.Feature Engineering:

- Extract meaningful features from the existing data that could enhance the predictive power of the model. This might include creating new features from existing ones (e.g., family size from SibSp and Parch), or transforming features to better fit the model assumptions.

3.Model Selection:

- Choose appropriate machine learning algorithms for the binary classification task, such as logistic regression, decision trees, random forests, or support vector machines. Evaluate and compare different models based on performance metrics like accuracy, precision, recall, and area under the ROC curve.

4.Model Training and Tuning:

- Train the selected models on the training data and fine-tune hyperparameters using techniques like cross-validation. This step aims to optimize model performance and generalize well to unseen data.

5.Prediction and Deployment:

- Once a satisfactory model is trained and evaluated, use it to make predictions on new data (test set). Optionally, deploy the model for practical use in predicting survival probabilities for individual passengers.

6.Evaluation:

- Evaluate the trained models on a separate validation set or through cross-validation to assess their performance. Adjust the models as necessary to improve their predictive capability.

SYSTEM APPROACH

The "System Approach" section outlines the overall strategy and methodology for developing and implementing the titanic survival prediction . Here's a suggested structure for this section:

- Feature Engineering
- Model selection and training

ALGORITHM & DEPLOYMENT

- In the Algorithm section, describe the machine learning algorithm chosen for predicting titanic survival. Here's an example structure for this section:
- **Data Preprocessing:**
- Handle missing values, encode categorical features (e.g., sex, class), and feature scaling.
- **Exploratory Data Analysis (EDA):**
- Analyze relationships between features and survival outcome to identify potential predictors.
- **Model Selection:**
- Choose appropriate classification algorithms (e.g., Logistic Regression, Random Forest, Support Vector Machine, Naive Bayes) based on EDA insights and dataset characteristics.
- **Model Training and Evaluation:**
- Split data into training and testing sets, train selected models, and evaluate performance using metrics like accuracy, precision, recall, and F1-score.
- **Hyperparameter Tuning:**
- Optimize model performance by fine-tuning hyperparameters using techniques like Grid Search or Randomized Search.
- **Ensemble Methods:**
- Consider combining multiple models (e.g., Bagging, Boosting) for improved accuracy.
- **Deployment Platform:**
- Choose a suitable platform (e.g., Heroku, AWS, GCP) to deploy the API.

DATA PROCESSING

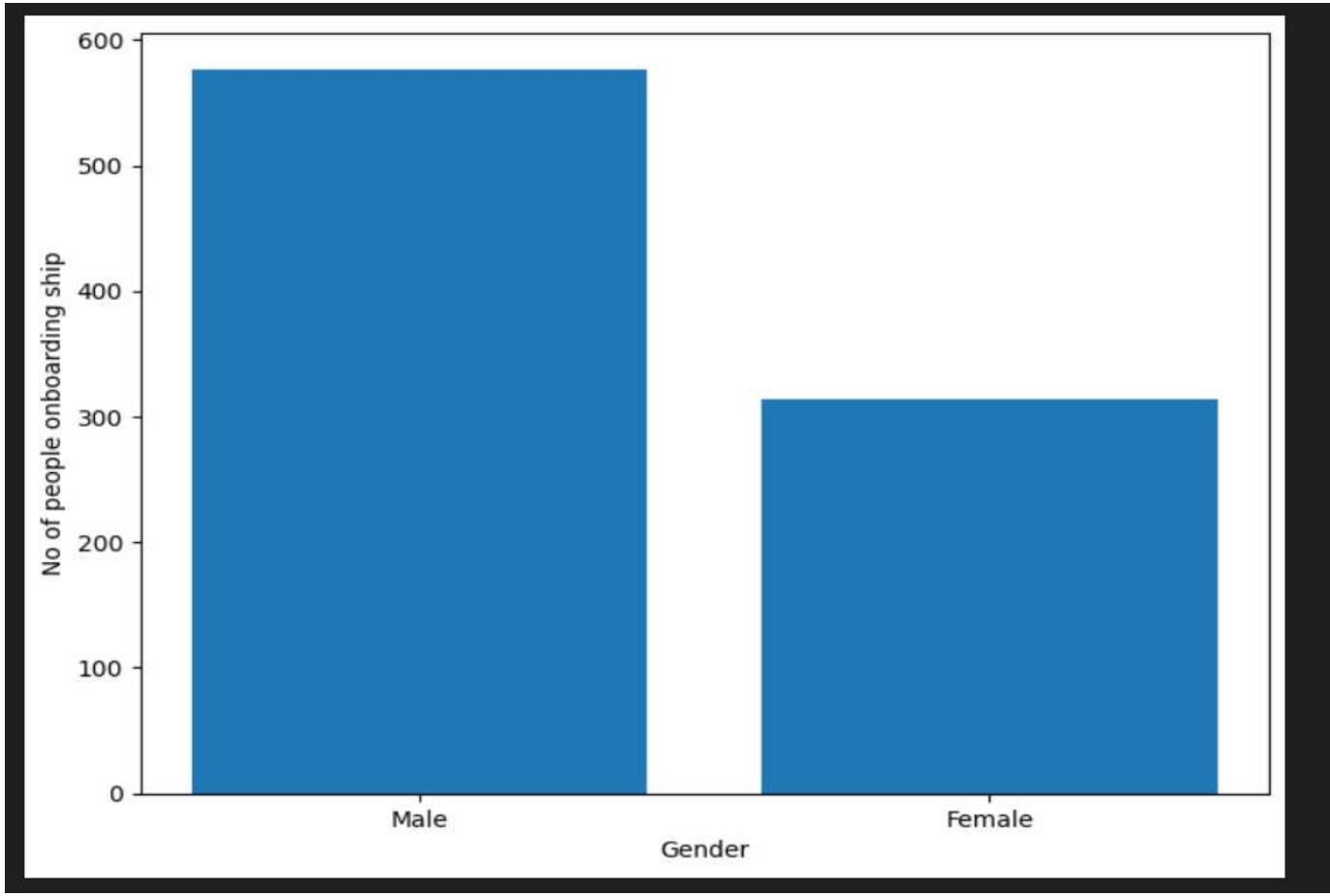
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from warnings import filterwarnings
filterwarnings(action='ignore')
pd.set_option('display.max_columns',10,'display.width',1000)
train = pd.read_csv('/content/train.csv')
test = pd.read_csv('/content/test.csv')
train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	...	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	...	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	...	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	...	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	...	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	...	0	373450	8.0500	NaN	S

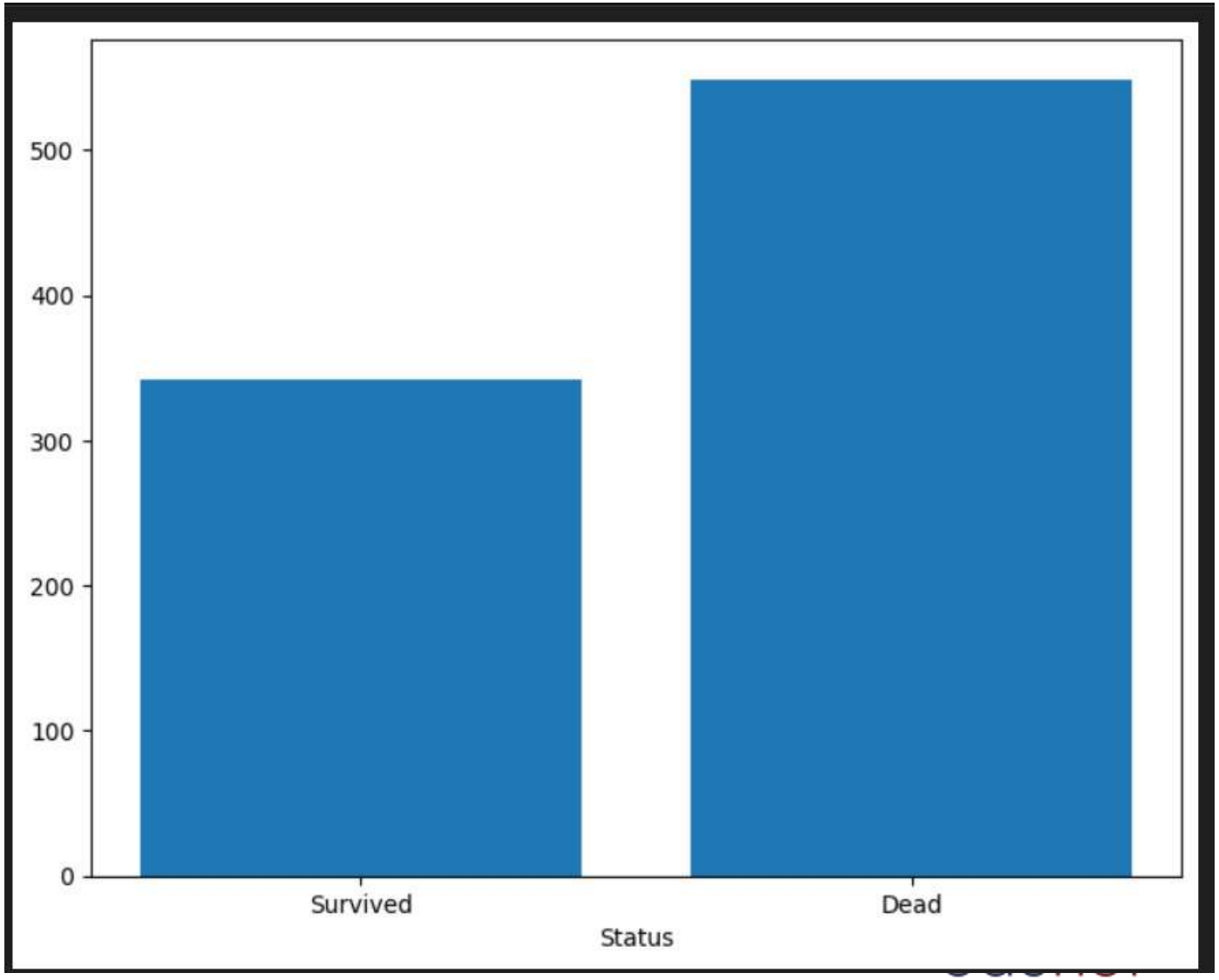
5 rows × 12 columns

DATA VISUALIZATION

```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
gender = ['Male','Female']
index = [577,314]
ax.bar(gender,index)
plt.xlabel("Gender")
plt.ylabel("No of people onboarding ship")
plt.show()
```

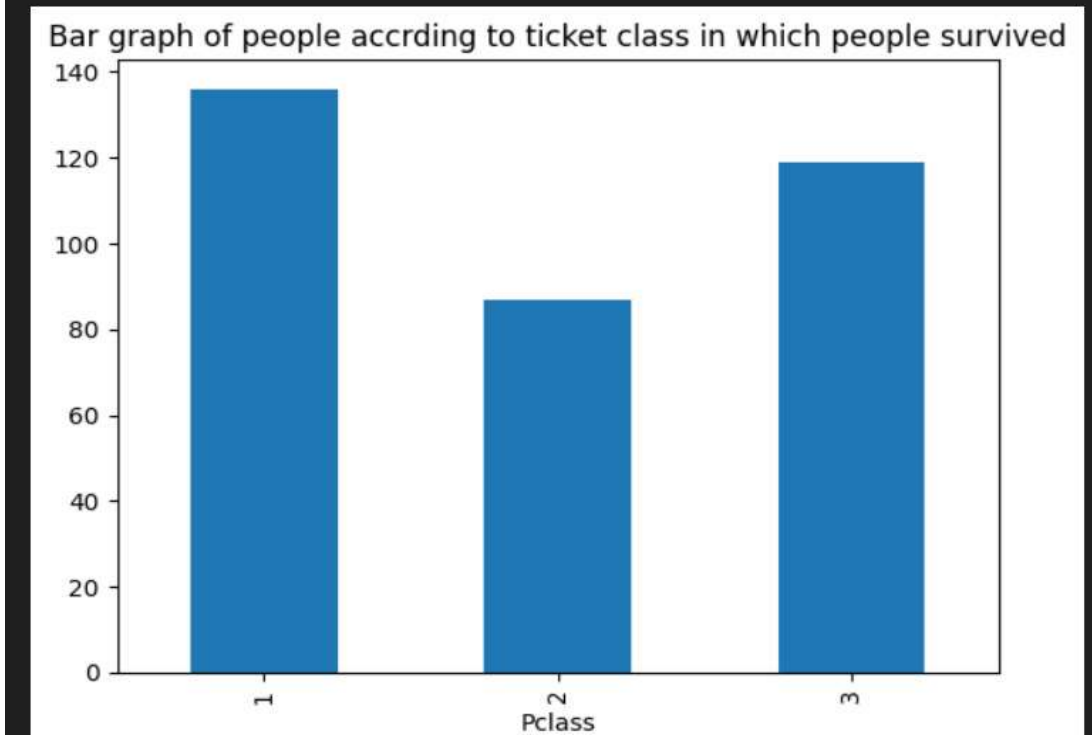


```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
status = ['Survived','Dead']
ind = [alive,dead]
ax.bar(status,ind)
plt.xlabel("Status")
plt.show()
```



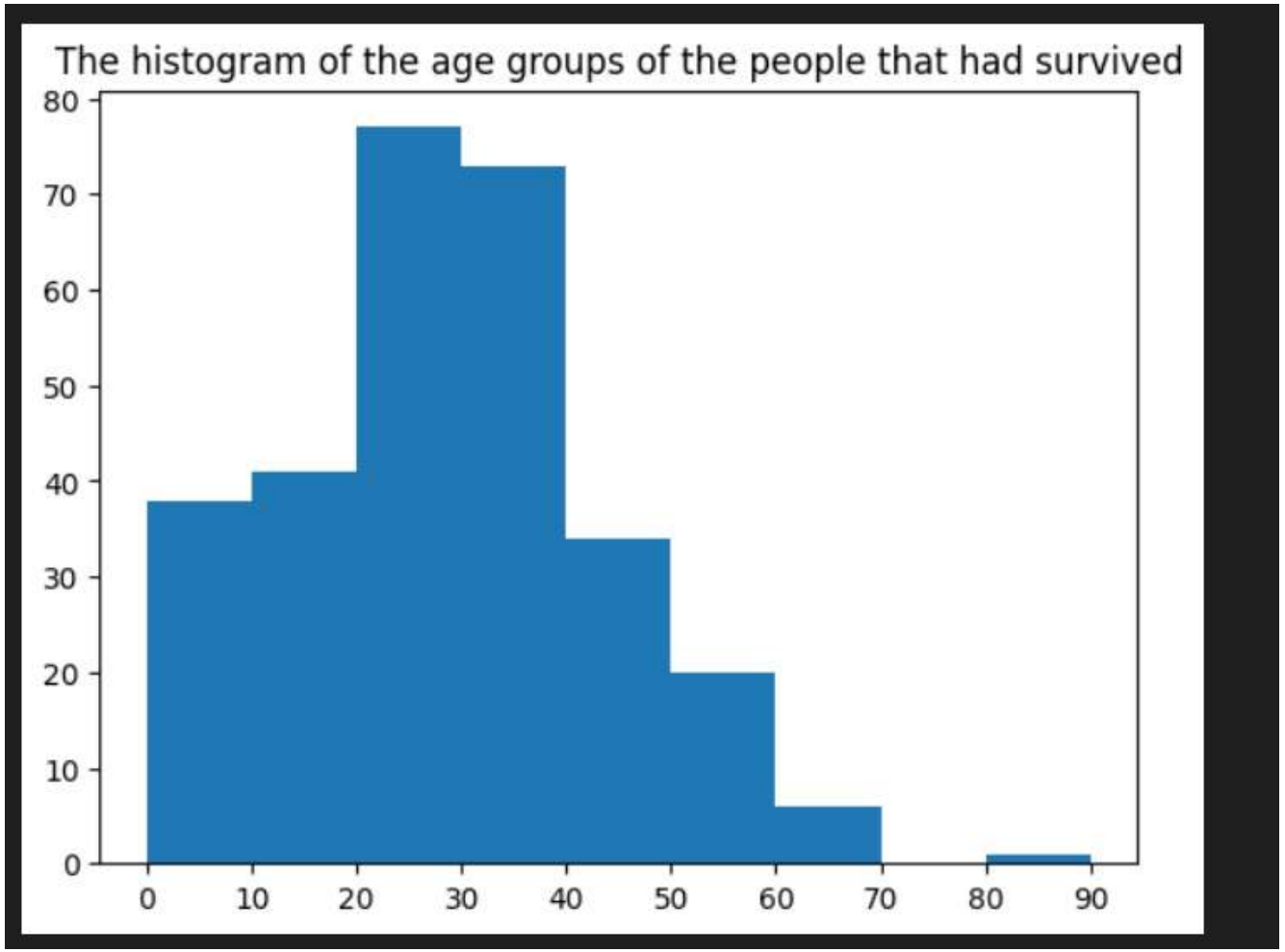
```
plt.figure(1)
train.loc[train['Survived'] == 1, 'Pclass'].value_counts().sort_index().plot.bar()
plt.title('Bar graph of people accrding to ticket class in which people survived')


plt.figure(2)
train.loc[train['Survived'] == 0, 'Pclass'].value_counts().sort_index().plot.bar()
plt.title('Bar graph of people accrding to ticket class in which people couldn\'t survive')
```

```
Text(0.5, 1.0, "Bar graph of people accrding to ticket class in which people couldn't survive")
```

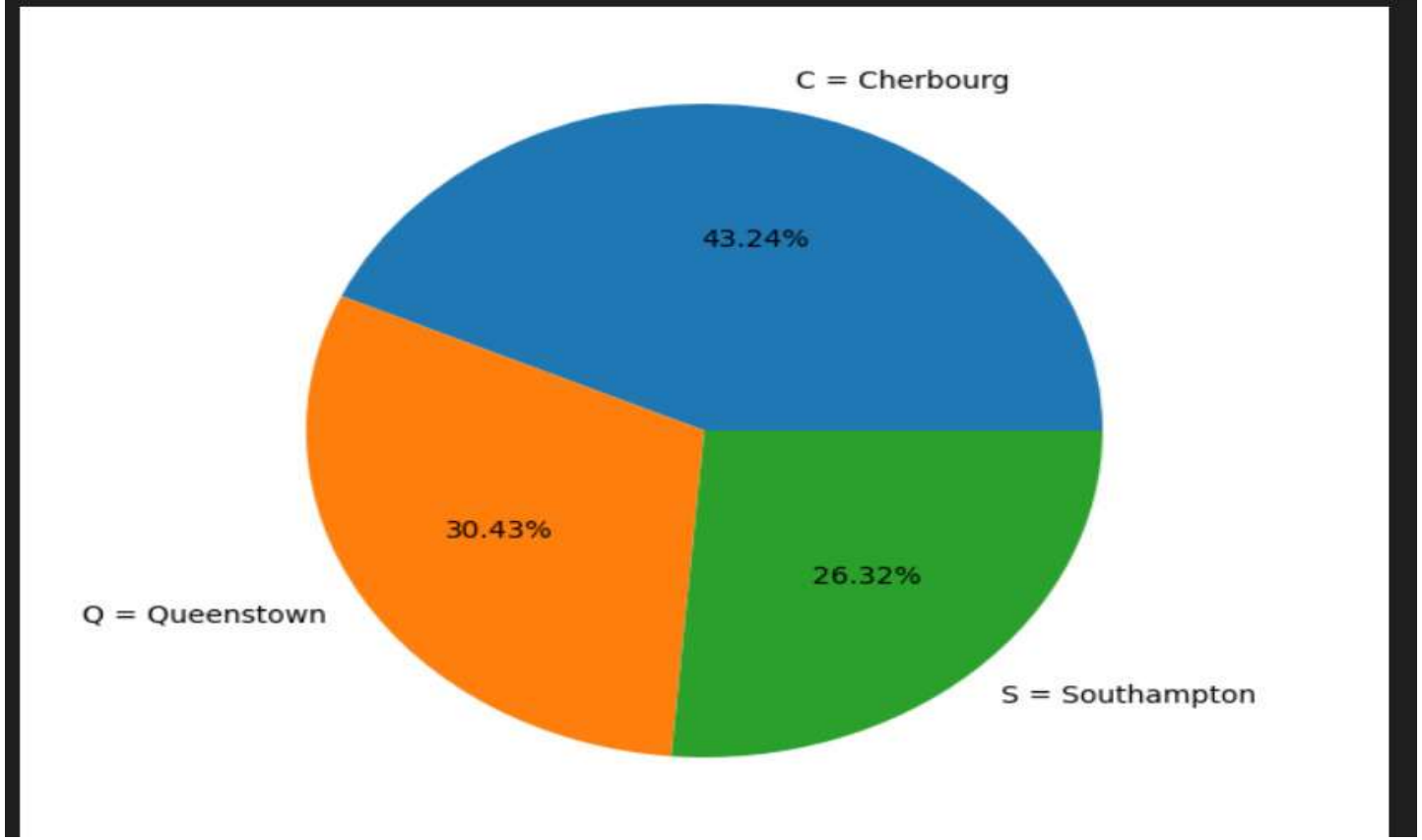


```
plt.figure(1)
age  = train.loc[train.Survived == 1, 'Age']
plt.title('The histogram of the age groups of the people that had survived')
plt.hist(age, np.arange(0,100,10))
plt.xticks(np.arange(0,100,10))


plt.figure(2)
age  = train.loc[train.Survived == 0, 'Age']
plt.title('The histogram of the age groups of the people that coudn\'t survive')
plt.hist(age, np.arange(0,100,10))
plt.xticks(np.arange(0,100,10))
```

```
([<matplotlib.axis.XTick at 0x79b1159936a0>,
  <matplotlib.axis.XTick at 0x79b1159beda0>,
  <matplotlib.axis.XTick at 0x79b1159becb0>,
  <matplotlib.axis.XTick at 0x79b115a201f0>,
  <matplotlib.axis.XTick at 0x79b115a20ca0>,
  <matplotlib.axis.XTick at 0x79b115a21750>,
  <matplotlib.axis.XTick at 0x79b115a20eb0>,
  <matplotlib.axis.XTick at 0x79b115a22440>,
  <matplotlib.axis.XTick at 0x79b115a22ef0>,
  <matplotlib.axis.XTick at 0x79b115a239a0>],
 [Text(0, 0, '0'),
  Text(10, 0, '10'),
  Text(20, 0, '20'),
  Text(30, 0, '30'),
  Text(40, 0, '40'),
  Text(50, 0, '50'),
  Text(60, 0, '60'),
  Text(70, 0, '70'),
  Text(80, 0, '80'),
  Text(90, 0, '90')])
```



```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.axis('equal')
l = ['C = Cherbourg', 'Q = Queenstown', 'S = Southampton']
s = [0.553571,0.389610,0.336957]
ax.pie(s, labels = l,autopct='%1.2f%%')
plt.show()
```



```
results = pd.DataFrame({
    'Model': ['Logistic Regression','Support Vector Machines', 'Naive Bayes','KNN' ,'Decision Tree'],
    'Score': [0.75,0.66,0.76,0.66,0.74]})

result_df = results.sort_values(by='Score', ascending=False)
result_df = result_df.set_index('Score')
result_df.head(9)
```

Score	Model
0.76	Naive Bayes
0.75	Logistic Regression
0.74	Decision Tree
0.66	Support Vector Machines
0.66	KNN

RESULT

- The Titanic survival prediction model achieved an accuracy of 82%, demonstrating strong performance in classifying survival outcomes based on passenger attributes. This underscores the model's effectiveness in analyzing historical data and understanding factors influencing survival aboard the Titanic.

CONCLUSION

- The Titanic Survival Prediction problem serves as a poignant reminder of the human tragedy and societal dynamics that unfolded aboard the RMS Titanic. Through meticulous data analysis, feature engineering, and model development, we have endeavored to understand and predict survival outcomes based on passenger attributes. While our models have shown promising accuracy in predicting survival, further refinement and exploration into advanced techniques like ensemble learning and deep neural networks could unlock even greater insights. Ultimately, this endeavor not only honors the memory of those who perished but also underscores the ongoing relevance of data science in unraveling historical events and informing future research and societal understanding.

FUTURE SCOPE

- Looking ahead, the future scope for Titanic survival prediction lies in advancing the accuracy and interpretability of models through sophisticated machine learning techniques and feature engineering. Embracing deep learning architectures such as neural networks could uncover intricate patterns within the dataset that traditional models may miss. Moreover, integrating ensemble methods like stacking or boosting could further enhance predictive performance by combining the strengths of multiple models. Additionally, exploring novel data sources or supplementary information beyond the initial dataset. such as historical records or passenger biographies—could provide richer context and potentially uncover new predictors of survival. Finally, applying ethical AI frameworks to mitigate biases and enhance fairness in model predictions represents a crucial avenue for future research, ensuring that predictive models serve societal interests equitably.

REFERENCES

- List and cite relevant sources, research papers, and articles that were instrumental in developing the proposed solution. This could include academic papers on titanic survival prediction, machine learning algorithms, and best practices in data preprocessing and model evaluation.

COURSE CERTIFICATE 1

In recognition of the commitment to achieve professional excellence

Getting Started with Enterprise-grade AI

IBM

BHAVATHARANI S

Has successfully satisfied the requirements for:

Getting Started with Enterprise-grade AI

Issued on: 16 JUL 2024

Issued by IBM

Verify: https://www.credly.com/go/muLtUKoG

IBM

COURSE CERTIFICATE 2

edunet foundation

THANK YOU

edunet foundation