

Pune Institute of Computer Technology



Department of Computer Engineering

(2022- 2023)

“Titanic Survival Prediction using Machine Learning”

Submitted to the

Savitribai Phule Pune University

In partial fulfilment for the award of the Degree of

Bachelor of Engineering

in

Computer Engineering

By

- | | | |
|----|----------------------------|--------------|
| 1) | Anuj Mahendra Mutha | 41443 |
| 2) | Nidhi Patil | 41447 |
| 3) | Amit Purohit | 41450 |

Abstract

This project is based on the Titanic dataset given on Kaggle. The sinking of the Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, the widely considered “unsinkable” Titanic sank after colliding with an iceberg. Unfortunately, there weren’t enough lifeboats for everyone on board, resulting in the **death**. In this project, we see how we can use machine-learning techniques to predict survivors of the Titanic. With a dataset of 891 individuals containing features like sex, age, and class, we attempt to predict the survivors of a small test group of 418. We are using Logistic Regression Model for the same.

Introduction

Machine learning means the application of any computer-enabled algorithm that can be applied against a data set to find a pattern in the data. This encompasses basically all types of data science algorithms, supervised, unsupervised, segmentation, classification, or regression". few important areas where machine learning can be applied are Handwriting Recognition, Language Translation, Speech Recognition, Image Classification, Autonomous Driving. Some features of machine learning algorithms can be observations that are used to form predictions for image classification, the pixels are the features, for voice recognition, the pitch and volume of the sound samples are the features and for autonomous cars, data from the cameras, range sensors, and GPS.

Using data provided by www.kaggle.com, our goal is to apply machine-learning techniques to successfully predict which passengers survived the sinking of the Titanic. Features like ticket price, age, sex, and class will be used to make the predictions. Using Logistic Regression methods, we try to predict the survival of passengers using different combinations of features. The challenge boils down to a classification problem given a set of features.

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>

Motivation

To predict what type of people survived the Titanic Shipwreck using passenger data and build its prediction model is the main motive to study this mini project.

Objective

Goal: Build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data like age, gender, class, etc.

Theory

Data Set:

The data we used for our project was provided on the Kaggle website. We were given 891 passenger samples for our training set and their associated labels of whether the passenger survived. For each passenger, we were given his/her passenger class, name, sex, age, number of siblings/spouses aboard, number of parents/children aboard, ticket number, fare, cabin embarked, and port of embarkation.

For the test data, we had 418 samples in the same format. The dataset is not complete, meaning that for several samples, one or many of fields were not available and marked empty (especially in the latter fields – age, fare, cabin, and port). However, all sample points contained at least information about gender and passenger class.

To normalize the data, we replace missing values with the mean of the remaining data set or other values.

Understanding the Titanic Dataset

So first we will understand our titanic dataset. This is a dataset of Titanic ship passengers & here

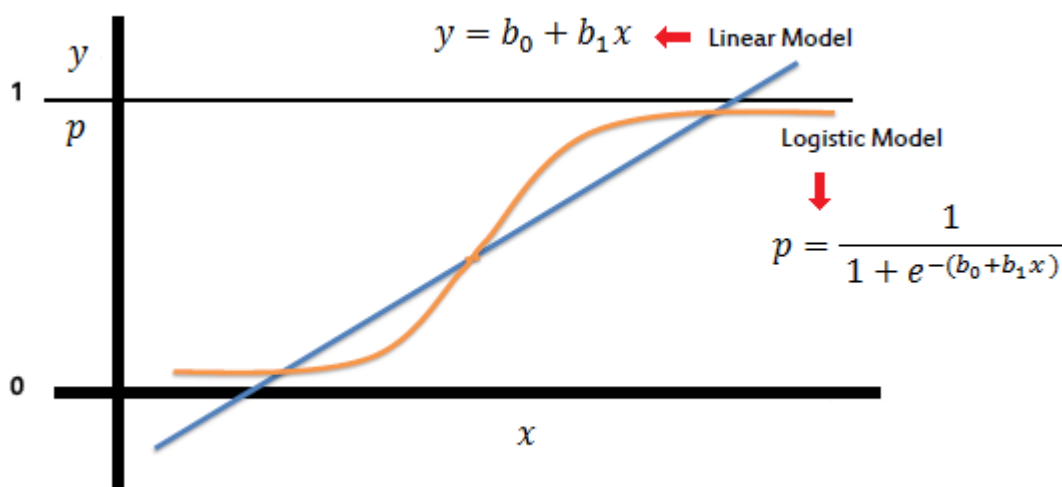
- Each row represents the data of 1 passenger.
 - Columns represent the features. We have 10 features/ variables in this dataset.
1. **Survival:** This variable shows whether the person survived or not. This is our target variable & we must predict its value. It's a binary variable. *0 means not survived and 1 means survived.*
 2. **pclass:** The ticket class of passengers. 1st (upper class), 2nd (middle), or 3rd (lower).

3. **Sex:** Gender of passenger
4. **Age:** Age (in years) of a passenger
5. **sibsp:** The no. of siblings/spouses of a particular passenger who were there on the ship.
6. **parch:** The no. of parents/children of a particular passenger who were there on the ship.
7. **ticket:** Ticket Number
8. **fare:** Passenger fare (like 1st class ticket fare must be greater than 2nd pr 3rd class ticket right)
9. **cabin:** Cabin Number
10. **embarked:** Port of Embarkation; From where that passenger took the ship. (C = Cherbourg, Q = Queenstown, S = Southampton)

Logistic Regression:

A simple yet crisp description of Logistic Description would be, “it is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.” as stated in the tutorial points article.

The graph of logistic regression is as shown below:



What is Training Dataset?

The *training data is the biggest (in -size) subset of the original dataset, which is used to train or fit the machine learning model*. Firstly, the training data is fed to the ML algorithms, which lets them learn how to make predictions for the given task.

What is Test Dataset?

Once we train the model with the training dataset, it's time to test the model with the test dataset. This dataset evaluates the performance of the model and ensures that the model can generalize well with the new or unseen dataset. *The test dataset is another subset of original data, which is independent of the training dataset*. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

Accuracy

To find the accuracy of model in confusion matrix the formula is:

$$accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ positives + false\ negatives}$$

Workflow

Work Flow



Data



Data pre processing



Data Analysis



Train Test split



Logistic Regression model



Evaluation

CODE & RESULT:

CO

Project 1 · Titanic Survival Prediction using Machine Learning.ipynb

☆

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

Importing the Dependencies

```
[ ] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Data Collection & Processing

```
[ ] # load the data from csv file to Pandas DataFrame
titanic_data = pd.read_csv('/content/train.csv')
```

```
[ ] # printing the first 5 rows of the dataframe
titanic_data.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

CO

Project 1 · Titanic Survival Prediction using Machine Learning.ipynb

☆

File Edit View Insert Runtime Tools Help Changes will not be saved

+ Code + Text Copy to Drive

```
[ ] 3      4      1      1      Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1      0      113803  53.1000  C123      S
4      5      0      3      Allen, Mr. William Henry      male    35.0      0      0      373450   8.0500   NaN      S
```

```
[ ] # number of rows and columns
titanic_data.shape

(891, 12)
```

```
[ ] # getting some informations about the data
titanic_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   PassengerId  891 non-null    int64
1   Survived     891 non-null    int64
2   Pclass       891 non-null    int64
3   Name         891 non-null    object
4   Sex          891 non-null    object
5   Age          714 non-null    float64
6   SibSp        891 non-null    int64
7   Parch        891 non-null    int64
8   Ticket       891 non-null    object
9   Fare         891 non-null    float64
10  Cabin        204 non-null    object
11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```



+ Code + Text Copy to Drive

```
[ ] # check the number of missing values in each column
titanic_data.isnull().sum()
```

```
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
```

Handling the Missing values

```
[ ] # drop the "Cabin" column from the dataframe
titanic_data = titanic_data.drop(columns='Cabin', axis=1)
```

```
[ ] # replacing the missing values in "Age" column with mean value
titanic_data['Age'].fillna(titanic_data['Age'].mean(), inplace=True)
```

```
[ ] # finding the mode value of "Embarked" column
print(titanic_data['Embarked'].mode())
```

```
0    S
dtype: object
```




Project 1: Titanic Survival Prediction using Machine Learning.ipynb ☆

File Edit View Insert Runtime Tools Help [Changes will not be saved](#)



+ Code + Text Copy to Drive



dtype: object

```
[ ] print(titanic_data['Embarked'].mode()[0])
```

S

```
[ ] # replacing the missing values in "Embarked" column with mode value
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=True)
```

```
[ ] # check the number of missing values in each column
titanic_data.isnull().sum()
```

```
PassengerId    0
Survived        0
Pclass          0
Name            0
Sex             0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked        0
dtype: int64
```



Data Analysis



```
[ ] # getting some statistical measures about the data
titanic_data.describe()
```



Project 1 . Titanic Survival Prediction using Machine Learning.ipynb



File Edit View Insert Runtime Tools Help [Changes will not be saved](#)

+ Code + Text Copy to Drive

Data Analysis

```
[ ] # getting some statistical measures about the data
titanic_data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
# finding the number of people survived and not survived
titanic_data['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```



Project 1 . Titanic Survival Prediction using Machine Learning.ipynb

Share

File Edit View Insert Runtime Tools Help [Changes will not be saved](#)

+ Code + Text Copy to Drive

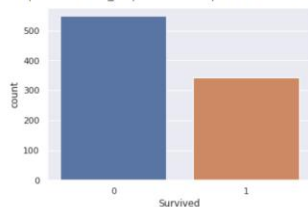
Reconnect Editing

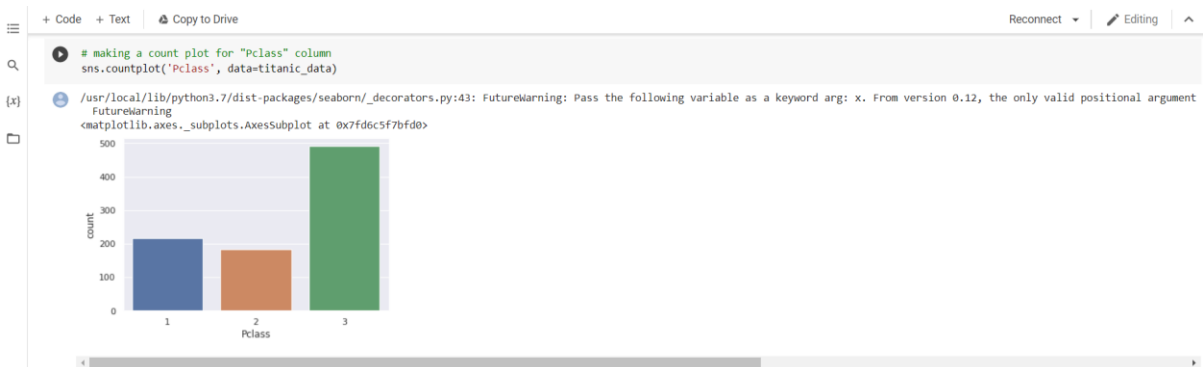
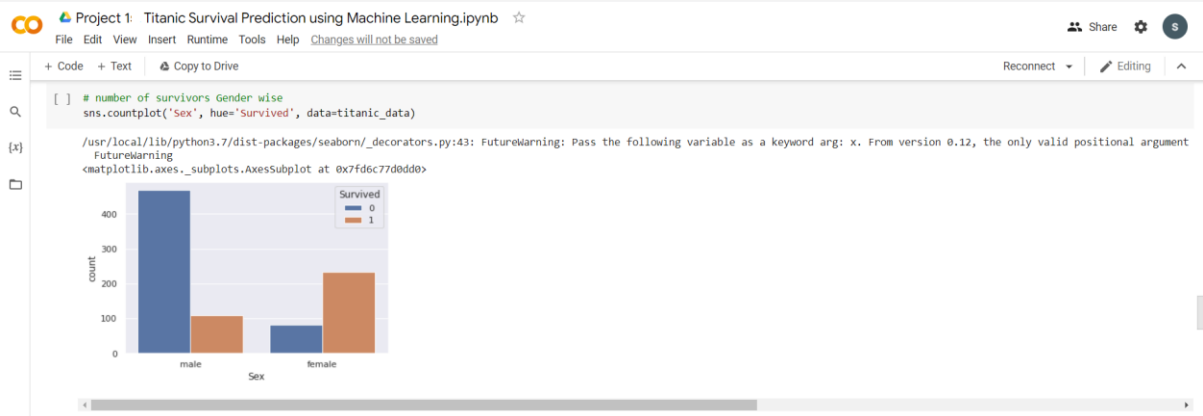
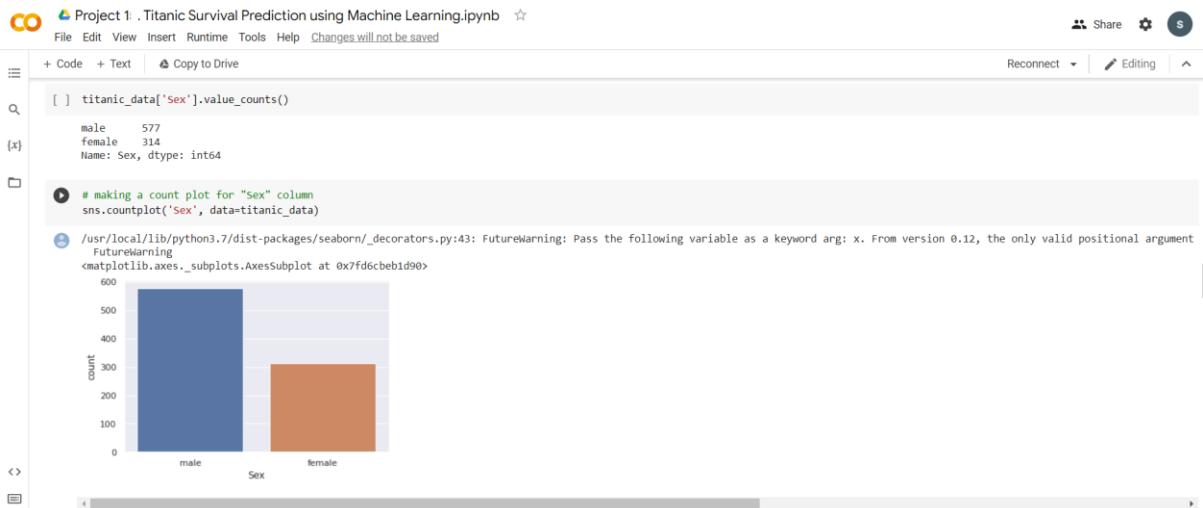
Data Visualization

```
[ ] sns.set()
```

```
# making a count plot for "Survived" column
sns.countplot('Survived', data=titanic_data)
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument is x
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fd6c77f16d0>







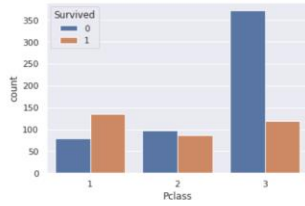
+ Code + Text Copy to Drive

Reconnect Editing ^



```
[ ] sns.countplot('Pclass', hue='Survived', data=titanic_data)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument is: x
<matplotlib.axes._subplots.AxesSubplot at 0x7fd6c7286a90>
```





Project 1 . Titanic Survival Prediction using Machine Learning.ipynb ☆

File Edit View Insert Runtime Tools Help [Changes will not be saved](#)

+ Code + Text Copy to Drive

Data Analysis

```
[ ] # getting some statistical measures about the data
titanic_data.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	13.002015	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	22.000000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	29.699118	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	35.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
# finding the number of people survived and not survived
titanic_data['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```



Project 1 . Titanic Survival Prediction using Machine Learning.ipynb ☆

File Edit View Insert Runtime Tools Help [Changes will not be saved](#)

Share ⚙️ S

+ Code + Text Copy to Drive

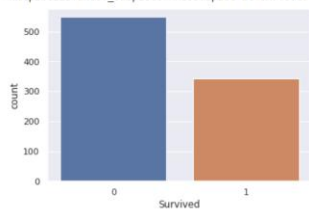
Reconnect Editing ^

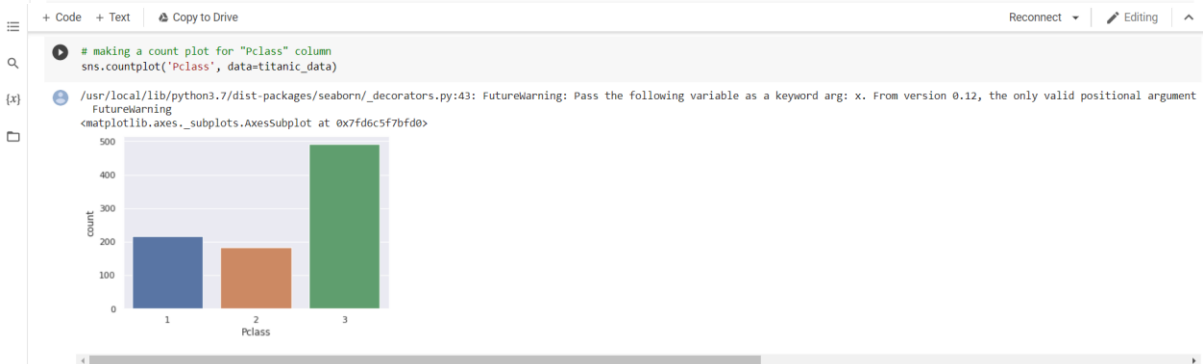
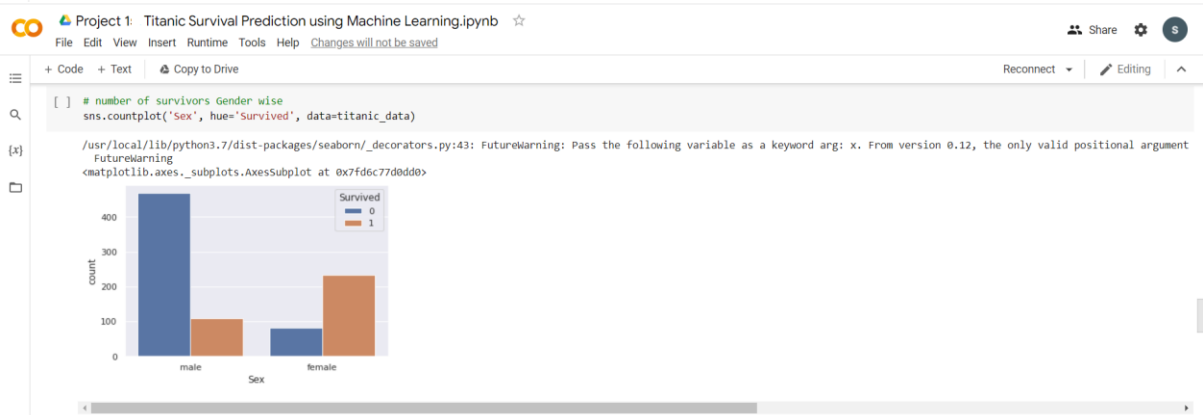
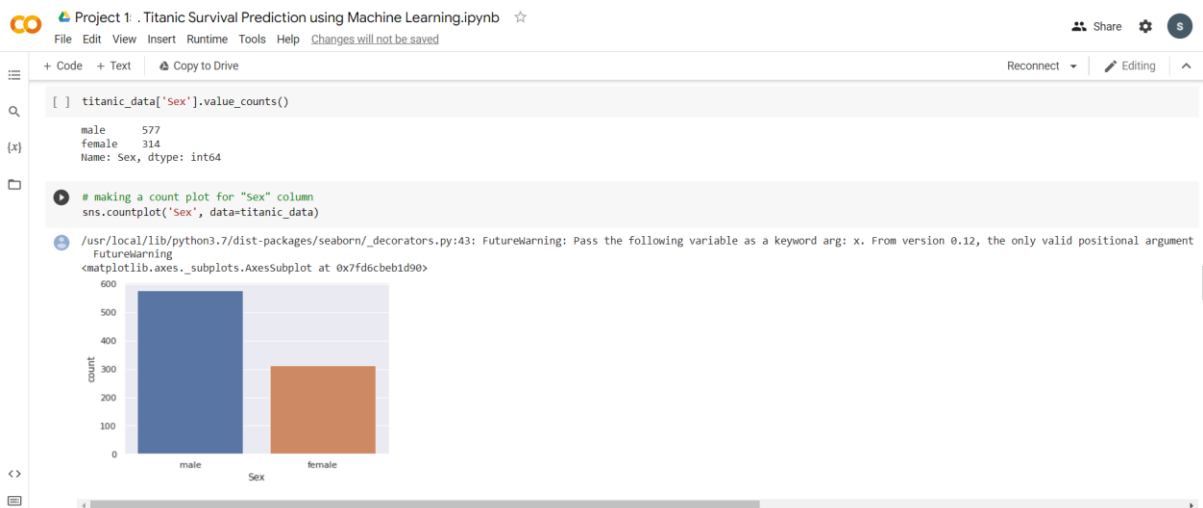
Data Visualization

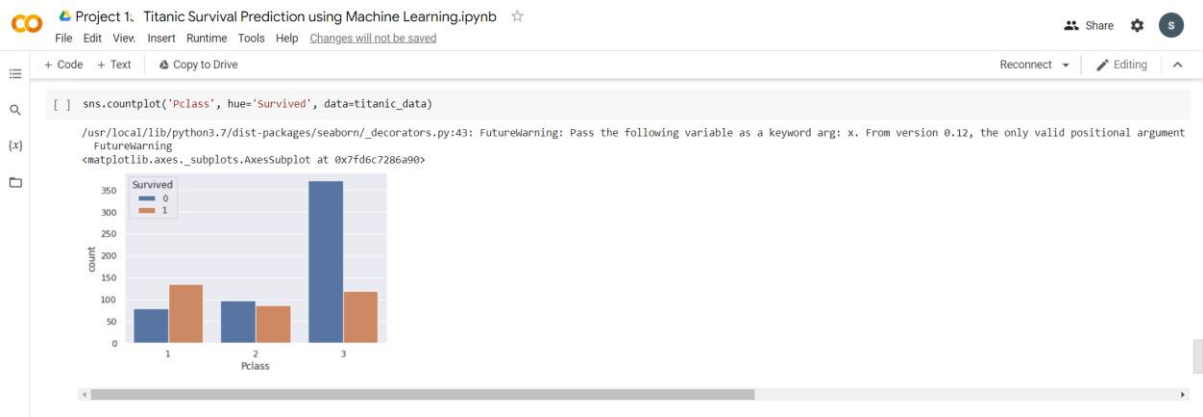
```
[ ] sns.set()
```

```
# making a count plot for "Survived" column
sns.countplot('Survived', data=titanic_data)
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument is x.
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fd6c77f16d0>







Project 1

Titanic Survival Prediction using Machine Learning.ipynb

☆

File

Edit

View

Insert

Runtime

Tools

Help

Changes will not be saved

+ Code

+ Text

Copy to Drive

Encoding the Categorical Columns

[] titanic_data['Sex'].value_counts()

male 577
female 314
Name: Sex, dtype: int64

[] titanic_data['Embarked'].value_counts()

S 646
C 168
Q 77
Name: Embarked, dtype: int64

[] # converting categorical Columns

titanic_data.replace({'Sex':{'male':0,'female':1}, 'Embarked':{'S':0,'C':1,'Q':2}}, inplace=True)

[] titanic_data.head()

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	1
2	3	1	3	Helkkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	0

+

 Code

+

 Text

📄

 Copy to Drive

0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	1
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	0
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	0

Separating features & Target

```
[ ] X = titanic_data.drop(columns = ['PassengerId', 'Name', 'Ticket', 'Survived'], axis=1)
Y = titanic_data['Survived']
```

```
[ ] print(X)
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	22.000000	1	0	7.2500	0
1	1	1	38.000000	1	0	71.2833	1
2	3	1	26.000000	0	0	7.9250	0
3	1	1	35.000000	1	0	53.1000	0
4	3	0	35.000000	0	0	8.0500	0
...
886	2	0	27.000000	0	0	13.0000	0
887	1	1	19.000000	0	0	30.0000	0
888	3	1	29.699118	1	2	23.4500	0
889	1	0	26.000000	0	0	30.0000	1
890	3	0	32.000000	0	0	7.7500	2

[891 rows x 7 columns]

+

 Code

+

 Text

📄

 Copy to Drive

0	1	0	3	Braund, Mr. Owen Harris	0	22.0	1	0	A/5 21171	7.2500	0
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	1	38.0	1	0	PC 17599	71.2833	1
2	3	1	3	Heikkinen, Miss. Laina	1	26.0	0	0	STON/O2. 3101282	7.9250	0
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	1	35.0	1	0	113803	53.1000	0
4	5	0	3	Allen, Mr. William Henry	0	35.0	0	0	373450	8.0500	0

Separating features & Target

```
[ ] X = titanic_data.drop(columns = ['PassengerId', 'Name', 'Ticket', 'Survived'], axis=1)
Y = titanic_data['Survived']
```

```
[ ] print(X)
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	0	22.000000	1	0	7.2500	0
1	1	1	38.000000	1	0	71.2833	1
2	3	1	26.000000	0	0	7.9250	0
3	1	1	35.000000	1	0	53.1000	0
4	3	0	35.000000	0	0	8.0500	0
...
886	2	0	27.000000	0	0	13.0000	0
887	1	1	19.000000	0	0	30.0000	0
888	3	1	29.699118	1	2	23.4500	0
889	1	0	26.000000	0	0	30.0000	1
890	3	0	32.000000	0	0	7.7500	2

[891 rows x 7 columns]



Project 1: Titanic Survival Prediction using Machine Learning.ipynb



File Edit View Insert Runtime Tools Help [Changes will not be saved](#)

+ Code + Text Copy to Drive



```
[ ] print(Y)
```

```
0      0
1      1
2      1
3      1
4      0
..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

Splitting the data into training data & Test data

```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size=0.2, random_state=2)
```

```
[ ] print(X.shape, X_train.shape, X_test.shape)
```

```
(891, 7) (712, 7) (179, 7)
```



Model Training



Logistic Regression



Project 1: Titanic Survival Prediction using Machine Learning.ipynb



File Edit View Insert Runtime Tools Help [Changes will not be saved](#)

+ Code + Text Copy to Drive



Model Training

Logistic Regression

```
[ ] model = LogisticRegression()
```

```
# training the Logistic Regression model with training data
model.fit(X_train, Y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, l1_ratio=None, max_iter=100,
                    multi_class='auto', n_jobs=None, penalty='l2',
                    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                    warm_start=False)
```



+ Code + Text Copy to Drive



Model Evaluation



Accuracy Score



```
[ ] # accuracy on training data
X_train_prediction = model.predict(X_train)
```

```
print(X_train_prediction)
```

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



+ Code + Text Copy to Drive

```
[ ] training_data_accuracy = accuracy_score(Y_train, X_train_prediction)
print('Accuracy score of training data : ', training_data_accuracy)
```

```
Accuracy score of training data : 0.8075842696629213
```

```
[ ] # accuracy on test data
X_test_prediction = model.predict(X_test)
```

```
[ ] print(X_test_prediction)
```

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

```
[ ] test_data_accuracy = accuracy_score(Y_test, X_test_prediction)
print('Accuracy score of test data : ', test_data_accuracy)
```

```
Accuracy score of test data : 0.7821229050279329
```

Conclusion

The analysis revealed interesting patterns across individual-level features. Factors such as socioeconomic status, social norms and family composition appeared to have an impact on likelihood of survival. These conclusions, however, were derived from findings in the given data set.

It has been observed that female survival rates are very high (approx 74%) while male survival rates are very low. To make predictions in classification problem, the technique of logistic regression is primarily used.

It would be interesting to play more with dataset and introducing more attributes which might lead to better results. Various other machine learning techniques like Naive Bayes, K-NN classification can be used to solve the problem.

References

- [1] Kaggle, Titanic: Machine Learning from Disaster [Online]. Available: <http://www.kaggle.com/>

- [2] Prediction of Survivors in Titanic Dataset: A Comparative Study using Machine Learning Algorithms, Tryambak Chatterlee, IJERMT-2017.

- [3] Eric Lam, Chongxuan Tang, "Titanic Machine Learning from Disaster", LamTang-Titanic Machine Learning From Disaster, 2012.

- [4] Analyzing Titanic disaster using machine learning algorithms-Computing, Communication and Automation (ICCCA), 2017 International Conference on 21 December 2017, IEEE.

- [5] <https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers->
Niklas Donges

- [6] <https://www.analyticsvidhya.com/machine-learning>

- [7] Wikipedia. Logistic Regression [Online]. Available: https://en.wikipedia.org/wiki/Logistic_regression