**Title of Project**

**Project-Based Internship 2020 Report**

Submitted

To

**DataRitz Technologies**

**Duration : Six Weeks**

**By**

**MANISHA JOSHI**

**VIBHANSHU JAISWAL**

**Roll Number**

**1803213092**

**1803213169**

**ABES ENGINEERING COLLEGE**

**ABDUL KALAM TECHNICAL UNIVERSITY**

**Under the guidance of**

**GOPAL SHARMA**

**SHASHANK SHEKHAR**

**TABLE OF CONTENTS**

Page No.

DECLARATION ii

CERTIFICATE iii

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF TABLES viii

LIST OF FIGURES ix

**CHAPTER-1 INTRODUCTION 1**

1.1 Problem Definition 1

1.2 Motivation 1

1.3 Objective of the Project 1

1.4 Scope of the Project 2

1.5 Need of Work 2

**CHAPTER 2 RELATED WORK 3**

**CHAPTER 3 PROPOSED METHODOLOGY 9**

3.1 Dataset Description 9

3.2 Methods 9

3.3 Hardware / Software Requirements 17

3.3 Our Methodology 17

**CHAPTER 4 EXPERIMENT AND RESULT ANALYSIS 33**

**CHAPTER 5 CONCLUSION 34**

**5.1 Discussion** 34

**5.2 Future Work** 34

**REFERENCES 36**

## CERTIFICATE

This is to certify that Project Report entitled “PREDICTING THE SURVIVAL OF TITANIC PASSENGERS” which is submitted by MANISHA JOSHI and VIBHANSHU JAISWAL in partial fulfillment of the requirement for the summer internship of DATA SCIENCE AND MACHINE LEARNING in Department of INFORMATION TECHNOLOGY of ABES ENGINEERING COLLEGE, is a record of the candidate own work carried out by her under my/our supervision.

**Supervisor 1**

**Supervisor 2**

**Date**

ACKNOWLEDGEMENT

*It gives us a great sense of pleasure to present the report of the Project Based Internship 2020 undertaken during Summer internship 2020 We owe special debt of gratitude to Mr. Gopal Sharma (Sr. Astt. Professor) and Mr. Shashank Shekhar* (*Project Consultant), DataRitz Technologies for his constant support and guidance throughout the course of our work. His constant motivation have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.*

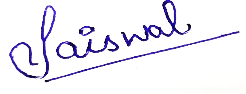
*We also take the opportunity to acknowledge the contribution of team members of DataRitz Technologies for their full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the motivation of Information Technology ABES Engineering College to provide us the opportunity to undergo training at DataRitz Technologies.*

*Name : Manisha Joshi*

**

*Vibhanshu Jaiswal*

**

*Roll No.: 1803213092,1803213169*

*Date : 08-07-2020*

**ABSTRACT**

RMS Titanic sinking is one of the most infamous shipwrecks in history. During its maiden voyage, the Titanic sank after colliding with an iceberg, killed many passengers including crew. This sensational tragedy shocked the international community and led to better safety regulations for ships. Ref.[2] One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class. In this paper we are going to make the predictive analysis of what sorts of people were likely to survive and using some tools in machine learning to predict which passengers survived the tragedy.

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. In particular, compare different machine learning techniques like Naïve Bayes, Random Forest Classifier, decision tree and some other analysis.

Project Summary

|  |  |
| --- | --- |
| 1. **Region/Unit** | 1. DataRitz Technologies |
| 1. **Location** | 1. Ghaziabad |
| 1. **Program** | 1. DataRitz Technologies.<<programcode>>.<<version>> |
| 1. **Project Number** | 1. DataRitz Technologies. <<projectcode>>.<<version>> |
| 1. **Project Description** | 1. <<project description>> |

Document Control

|  |  |
| --- | --- |
| 1. Prepared by: | Manisha Joshi, Vibhanshu Jaiswal |
| 1. Title: | Prediction the survival of titanic passengers |
| 1. College: | ABES Engineering College |
| 1. Department: | Information Technology |
| 1. Location: | Ghaziabad |
| 1. Version date: | 1. <<date of document completion>> |
| 1. Status: | 1. **<<Initial Draft**/Consultation Draft/Approved Document/Minor Revision/Major Revision>> |

Version history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version no.** | **Date** | **Changed by** | **Nature of amendment** |
| 1. **1.0** |  |  |  |
| 1. **1.1** |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Endorsement and Approval

Project Customer

I approve the business requirements specifications in this document.

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Name | <<customer name>> | | |
| 1. Position | <<customer position>> | | |
| 1. Signature |  | 1. Date |  |

The following officers have **endorsed** this document

Project Sponsor

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Name | 1. <<sponsor name>> | | |
| 1. Position | 1. <<sponsor position>> | | |
| 1. Signature |  | 1. Date |  |

Project Manager (= Component Project Customer)

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Name | 1. Mr. Gopal Gupta / Mr. Shashank Shekhar | | |
| 1. Position | 1. Lead Technical Architect / Project Consultant | | |
| 1. Signature |  | 1. Date |  |

Component Project Sponsor

I accept the business requirements specifications in this document.

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Name | 1. Dr B P Sharma | | |
| 1. Position | 1. Country Head – Delivery | | |
| 1. Signature |  | 1. Date |  |
| 1. **Comments** | | | |
|  | | | |

The following officers have **endorsed** this document

Component Program Manager

|  |  |  |  |
| --- | --- | --- | --- |
| 1. Name | 1. Mr. Gaurav Kansal | | |
| 1. Position | 1. Chief Operating Officer | | |
| 1. Signature |  | 1. Date |  |

**LIST OF TABLES**

Page No.

ML Algorithms 9

DATA PREPROCESSING 18

DATA VISUALIZATION 23

**LIST OF FIGURES**

**Page No.**

**Chapter 3**

1. **DATA PREPROCESSING**

Fig. 3.4.1 Dataset 18

Fig. 3.4.2 Dataset after updation 19

Fig. 3.4.3 Counts of Dataset 19

Fig. 3.4.4 Summary of Data 20

Fig. 3.4.5 Statistical data summary 20

Fig. 3.4.6 Classification of Attributes 22

Fig. 3.4.7 Conversion of datatypes 22

Fig. 3.4.8 Counts of non-null values 23

1. **DATA VISUALIZATION**

Fig. 3.4.9 Survival Status 24

Fig. 3.4.10 Survival wrt. Gender 24

Fig. 3.4.11 Age counts 25

Fig. 3.4.12 Total passengers wrt. Classes 25

Fig. 3.4.13 Embarked cities data 27

Fig. 3.4.14 Men survived 28

Fig. 3.4.15 Women survived 29

Fig. 3.4.16 Class vs. survived 30

Fig. 3.4.17 Class vs. age 31

Fig. 3.4.18 Correlation matrix 32

Fig. 3.4.19 Heatmap 32

1. **ML Algorithms**

Fig. 3.2.1 Feature and labels 9

Fig. 3.2.2 Test and train set 10

Fig. 3.3.3 Logistic Regression score 10

Fig. 3.2.4 Gausian Naïve Bayes Reg. score 11

Fig. 3.2.5 Decision Tree classifier score 12

Fig. 3.2.6 Random Forest Classifier score 13

Fig. 3.2.7 Comparison of models 15

Fig. 3.2.8 Visualization of ML algos. 16

**Chapter 4**

Fig.4.1 Comparison among the ML algorithms 33

**CHAPTER 1**

**INTRODUCTION**

* 1. **Problem Definition**

The problem statement entails predicting whether a passenger would survive or not survive given the features such as passenger class, sex, fare, age, number of siblings/spouse aboard, number of parents/children aboard, and others.

* 1. **Motivation**

“Titanic” was the famous tragedy of the maritime history, we all are aware of this fact. My motivation for this project is from the movie Titanic (1997) which led me to choose this objective to know more in deeper about this calamity.

This was also the result of ill planning and preparation of decision-making authority (like decrement in the total no. of lifeboats etc..).

The motive is to examine all those aspects due to which that travel result into the terrific tragedy and finding out some fruitful points so that we could apply them and could prevent the occurrences of such events in future.

* 1. **Objective of the Project:**

The purpose of this project is to understand all those causes/attributes/factors on which survival probability of a passenger depends.

The objective of this project was to build a classification model that could successfully determine whether a Titanic passenger lived or died.

* 1. **Scope of the Project**

The scope is to prepare predictive modeling for Titanic data to analyze with different python open source modules and produce prediction outputs with machine learning algorithms and find out accuracy by comparing different algorithms.

* 1. **Need of Work**

The need to work on predicting the survival through ML is to understand data analysis and applying Machine Learning algorithm to understand the procedure of prediction as it gives more efficient result after dealing with numerous algorithms.

**CHAPTER 2**

**RELATED WORK**

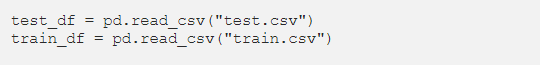
The related work done on the project through which i took the help is from the site Ref. [1]. In which first the data exploration is done in which we can see what is the missing data and learned which features are important. During the data preprocessing part, we computed missing values, converted features into numeric ones, grouped values into categories and created a few new features.

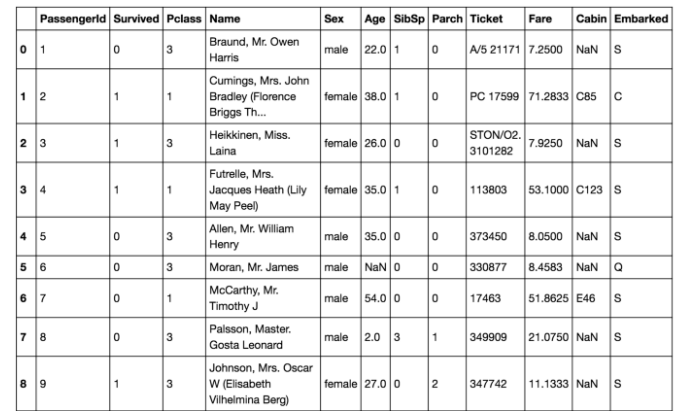
Afterwards training of 8 different machine learning models, picked one of them (random forest) and applied cross validation on it. After that how random forest works, took a look at the importance it assigns to the different features and tuned it’s performance through optimizing it’s hyperparameter values. Lastly, we looked at it’s confusion matrix and computed the models precision, recall and f-score.

So the related work done on the in which dataset is from Kaggle from preprocessing till prediction using ML is as follow step by step:

1. Getting the data

On the Kaggle there is two datasets: train and test datasets

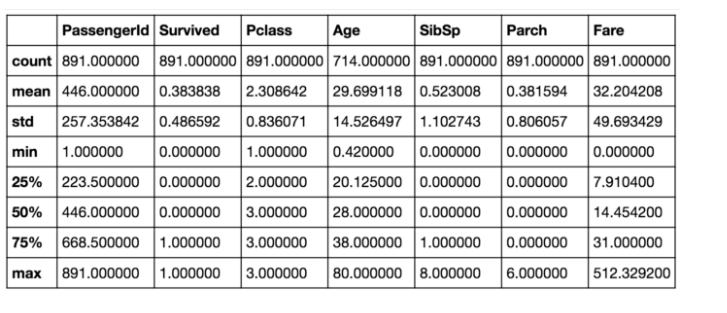




1. Summary of the data

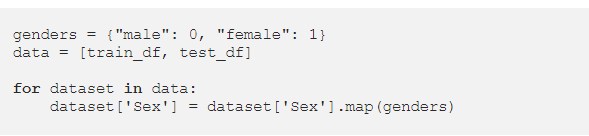
**The training-set has 891 examples and 11 features + the target variable (survived)**. 2 of the features are floats, 5 are integers and 5 are objects.

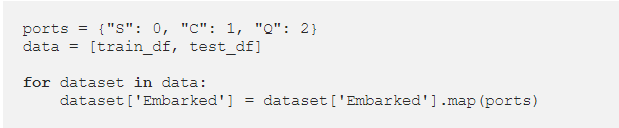
**38% out of the training-set survived the Titanic**. We can also see that the passenger ages range from 0.4 to 80. On top of that we can already detect some features, that contain missing values, like the ‘Age’ feature.



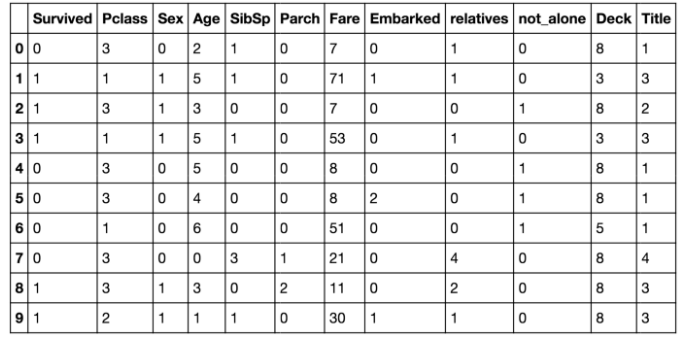
The statistical information related to train dataset.

Conversion applied on attributes:





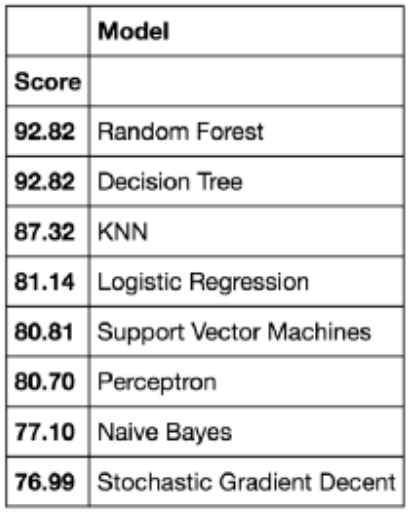
The train dataset after conversion, in which all the attributes have numerical data as follows:



Now the machine learning models that have been applied on this dataset are as follows:

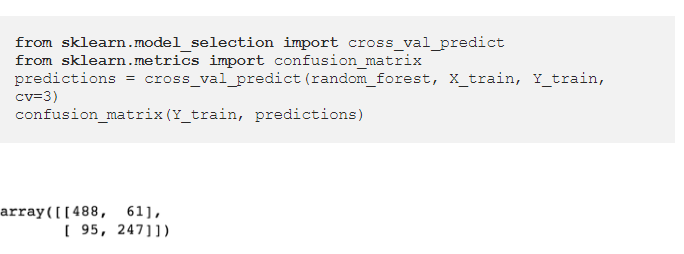
1. **Stochastic Gradient Descent (SGD)**
2. **Random Forest**
3. **Logistic Regression**
4. **K Nearest Neighbor**
5. **Gaussian Naive Bayes**
6. **Perceptron**
7. **Linear Support Vector Machine**
8. **Decision Tree**

After applying all these Machine algorithms, a comparison is made between all to get the algorithm which is giving the best result among all of them after it what result is coming is as follows:



We found out that the Random Forest classifier goes on the first place. This means it is giving the most accurate score.

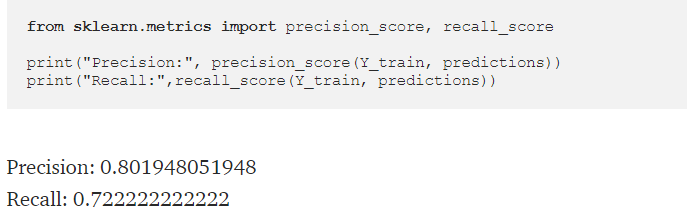
After the further training of the dataset, confusion matrix we found out, which is for the further evaluation:



A confusion matrix gives you a lot of information about how well your model does.

So here it is resulted that The first row is about the not-survived-predictions: **493 passengers were correctly classified as not survived** (called true negatives) and **56 where wrongly classified as not survived** (false positives).

The second row is about the survived-predictions: **93 passengers where wrongly classified as survived** (false negatives) and **249 where correctly classified as survived** (true positives).



This model predicts 81% of the time, a passengers survival correctly (precision). The recall tells us that it predicted the survival of 73 % of the people who actually survived.

Thus, this was the related work done on the project and it is entails that random forest classifier is the Machine Learning algorithm which is giving the best result among all of the Machine Learning algorithms that have been applied on the datasets loaded from Kaggle.

**CHAPTER 3**

**PROPOSED METHODOLOGY**

* 1. **Dataset Description**

As we know the large no. of passengers & crew around 2224 were there in the liner and many didn’t survive. We have took dataset from seaborn library which is named as 'titanic' which have 891 observations and 15 attributes. In this some attributes are repeated. We have excluded those attributes.

* survived: Outcome of survival (0 = No; 1 = Yes)
* p class: Socio-economic class (1 = Upper class; 2 = Middle class; 3 = Lower class)
* sex: Sex of the passenger
* age: Age of the passenger (Some entries contain NaN)
* sibsp: Number of siblings and spouses of the passenger aboard
* parch: Number of parents and children of the passenger aboard
* ticket: Ticket number of the passenger
* fare: Fare paid by the passenger
* cabin: Cabin number of the passenger (Some entries contain NaN)
* embarked: Port of embarkation of the passenger (C = Cherbourg; Q = Queenstown; S = Southampton)
  1. **Methods**

For the prediction purpose the **Machine Learning algorithms** used in the project are as follow:

* Logistic Regression
* Gausian naïve bayes
* Decision Tree
* Random Forest Classifier

#separating Features and Labels

X = data.iloc[:, 1:].values

y = data.iloc[:, 0:1].values

print (X.shape, y.shape)



Fig. 3.2.1

#splitting the data into train and test set

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state = 0)

print (X\_train.shape, y\_train.shape)



Fig. 3.2.2

1. **Logistic Regression**

Logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical. This article covers the case of a binary dependent variable—that is, where it can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. Cases where the dependent variable has more than two outcome categories may be analysed in multinomial logistic regression, or, if the multiple categories are ordered, in ordinal logistic regression.

#Logistic Regression

from sklearn.linear\_model import Logistic Regression

from sklearn.metrics import accuracy\_score

clf = Logistic Regression()

clf.fit(X\_train, y\_train.ravel())

y\_pred\_log\_reg = clf.predict(X\_test)

acc\_log\_reg = round( accuracy\_score(y\_pred\_log\_reg, y\_test) \* 100, 2)

print (str(acc\_log\_reg) + ' %')



Fig. 3.2.3

Thus, we are getting the testing accuracy as 79.48% by applying accuracy\_score.

1. **Gausian Naïve Bayes**

Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Bayes' theorem (alternatively Bayes' law or Bayes' rule) describes the probability of an event, based on prior knowledge of conditions that might be related to the event. For example, if cancer is related to age, then, using Bayes' theorem, a person's age can be used to more accurately assess the probability that they have cancer, compared to the assessment of the probability of cancer made without knowledge of the person's age.

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features.

#Gaussian Naïve Bayes

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

clf = GaussianNB()

clf.fit(X\_train, y\_train.ravel())

y\_pred\_gnb = clf.predict(X\_test)

acc\_gnb = round(accuracy\_score(y\_pred\_gnb, y\_test) \* 100, 2)

print (str(acc\_gnb) + '%')



Fig. 3.2.4

By perfoming the Gausian Naïve Baues we found out that it is giving accuracy\_score as 78.73%.

1. **Decision Tree**

A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

# Decision Tree Classifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

clf = DecisionTreeClassifier()

clf.fit(X\_train, y\_train.ravel())

y\_pred\_decision\_tree = clf.predict(X\_test)

acc\_decision\_tree = round(accuracy\_score(y\_pred\_decision\_tree, y\_test) \* 100, 2)

print (str(acc\_decision\_tree) + '%')



Fig. 3.2.5

By decision tree we are getting 78.36% accuracy\_score.

1. **Random Forest Classifier**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

#Random Forest Classifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

clf = RandomForestClassifier(n\_estimators=100)

clf.fit(X\_train, y\_train.ravel())

y\_pred\_random\_forest = clf.predict(X\_test)

acc\_random\_forest = round(accuracy\_score(y\_pred\_random\_forest, y\_test) \* 100, 2)

print (str(acc\_random\_forest) + '%')



Fig. 3.2.6

In this we are getting 82.09% accuracy\_score, which is the best prediction for the model till now.

# Comparing models

# dictionary to plot bar graph

classifiers = {'Logistic Regression':acc\_log\_reg,

'Gaussian Naive Bayes':acc\_gnb,

'Decision Tree':acc\_decision\_tree,

'Random Forest':acc\_random\_forest}

'Score': [acc\_log\_reg, acc\_decision\_tree,

acc\_random\_forest, acc\_gnb]

})

models = pd.DataFrame({

'Model': ['Logistic Regression', 'Decision Tree',

'Random Forest', 'Naive Bayes'],

models.sort\_values(by='Score', ascending=False)

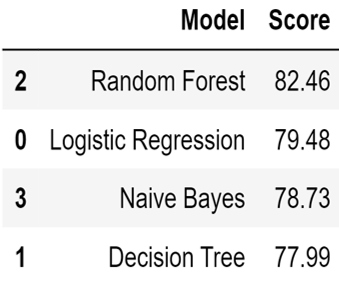


Fig. 3.2.7

# Visualization of all the classifiers

#Bar graph for models and accuracy score

p=sns.barplot(x='Model',y='Score',data='models')

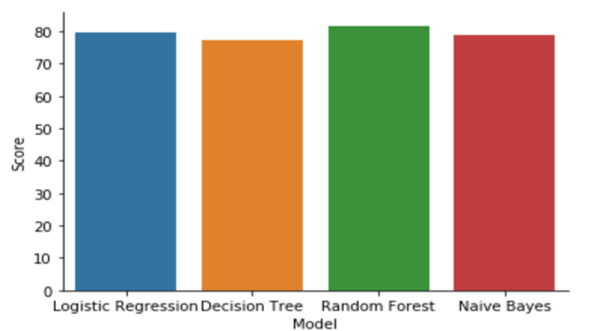
sns.despine()

Fig. 3.2.8

* 1. **Hardware / Software Requirements**

Software Requirements

# There is a need of operating System Windows 8. And a working environment IDE we have used : -

1. Anaconda Navigator - 5.3.0

2. Jupyter Notebook – 6.0.3

Hardware Requirements

# There is a need of pc/laptop of core i3 having 8GB RAM in order to work smoothly and conveniently. Less than 8GB RAM enable to operate anaconda.

* 1. **Our Methodology**

The data we collected is still raw data which is very likely to contains mistakes, missing values and corrupt values. Before drawing any conclusions from the data we need to do some data preprocessing which involves data wrangling and feature engineering. Data wrangling is the process of cleaning and unify the messy and complex data sets for easy access and analysis .Feature engineering process attempts to create additional relevant features from existing raw features in the data and to increase the predictive power of learning algorithms Our approach to solve the problem starts with collecting the raw data need to solve the problem and import the dataset into the working environment and do data preprocessing which includes data wrangling and feature engineering then explore the data and prepare a model for performing analysis using machine learning algorithms and evaluate the model and re-iterate till we get satisfactory model performance then compare the results within the algorithm and select a model which gives a more accurate results.

* Importing all the necessary libraries
* Cleaning and analyzing dataset
* Dropping null values
* Prediction through Machine Learning Model

1. **Data Preprocessing**

**Importing all the necessary libraries**

import seaborn as sns

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import sklearn

**removal of unnecessary attributes**

data=sns.load\_dataset('titanic')

print(data)

data.drop(['who','adult\_male','deck','alive','alone','embark\_town','class'],axis=1,inplace=True)

print(data)

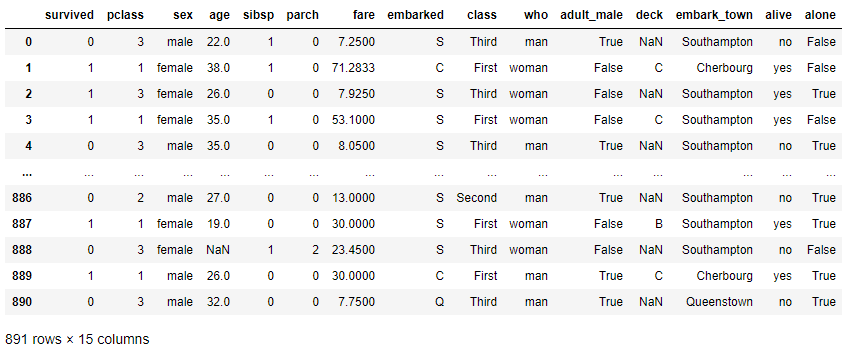


Fig.3.4.1

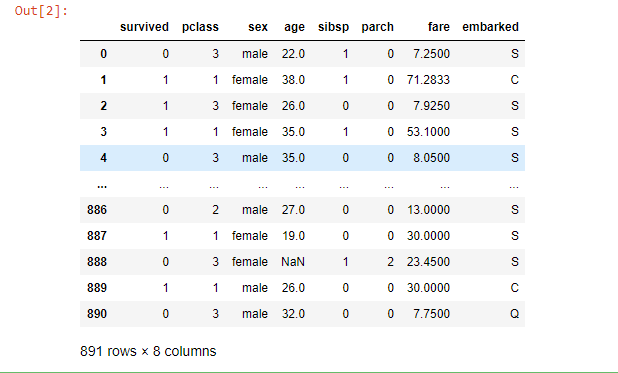


Fig.3.4.2

**non-null elements count**

data.count()

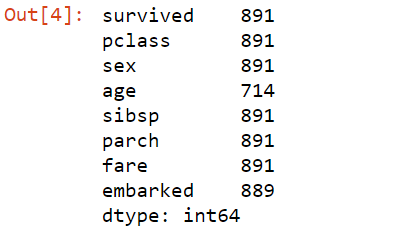


Fig. 3.4.3

**Summary of dataset**

data.info()

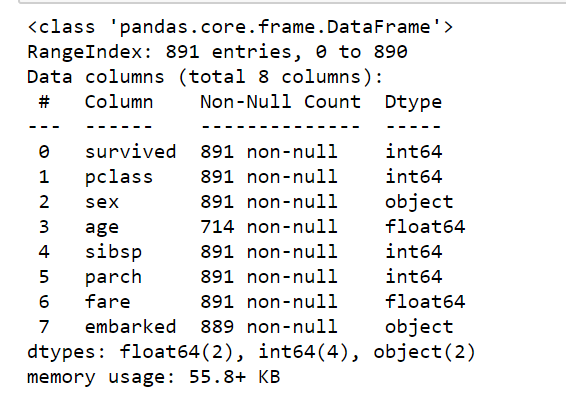


Fig. 3.4.4

**Statistical data summary**

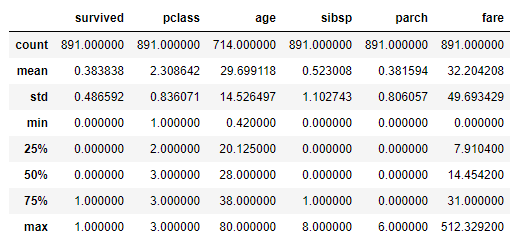


Fig.3.4.5

**Classfication of Numerical and Categorical attributes**

def get\_var\_category(data):

unique\_count=data.nunique(dropna=False)

total\_count=len(data)

if pd.api.types.is\_numeric\_dtype(data):

return "Numerical"

elif pd.api.types.is\_datetime64\_dtype(data):

return "Date"

elif unique\_count==total\_count:

return "Text(Unique)"

else:

return "Categorical"

def print\_categories(data):

for column\_name in data.columns:

print(column\_name, ": ", get\_var\_category(data[column\_name]))

print\_categories(data)

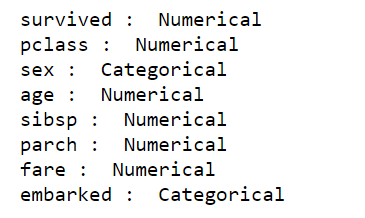


Fig. 3.4.6

**Conversion of data**

#assigning male as 0 female as 1

data['fare']=data["fare"].fillna(data['fare'].dropna().median())

data['age']=data['age'].fillna(data['age'].dropna().median())

data.loc[data['sex']=='male','sex']=0

data.loc[data['sex']=='female','sex']=1

#assigning Southampton city(S) as 3 Cherbourg(C) as 1 and Queenstown(Q) as 2

data['embarked']=data['embarked'].fillna('S')

data.loc[data['embarked']=='S','embarked']=3

data.loc[data['embarked']=='C','embarked']=1

data.loc[data['embarked']=='Q','embarked']=2

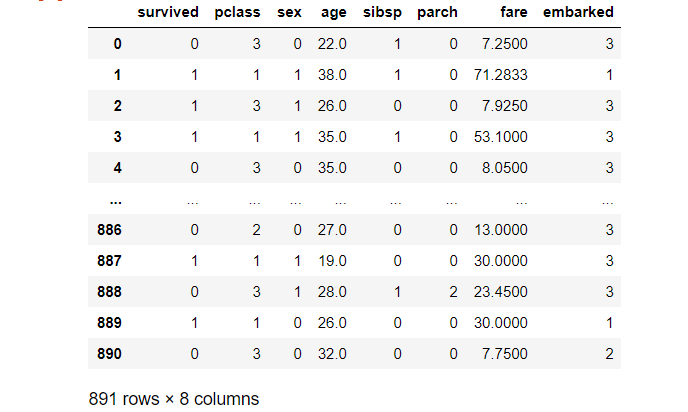


Fig. 3.4.7

**count total non-null values**

data.count()

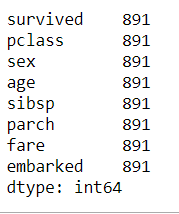


Fig.3.4.8

1. **Data Visualization**

## Perform univariate and bivariate analysis and derive meaningful insights about the dataset

## Univariate Analysis

## data.survived.value\_counts(normalize=True).plot(color='m',alpha=0.6,kind='bar')

## plt.xlabel('Survived : deceased(0) survived(1)',size=20)

## plt.ylabel('frequency',size=20)

## plt.title('Survival status',size=20)

## plt.xticks(rotation=0)

## plt.show()

## 

## Fig.3.4.9

## #this shows, total survived(male+female) are around 35-40% and not survived(male+female) are around 60%

data.sex.value\_counts().plot(color='m',alpha=0.6,kind='bar')

plt.xlabel('Gender : male(0) female(1)',size=20)

plt.ylabel('frequency',size=20)

plt.title('Survival status wrt. Gender',size=20)

plt.xticks(rotation=0)

plt.show()

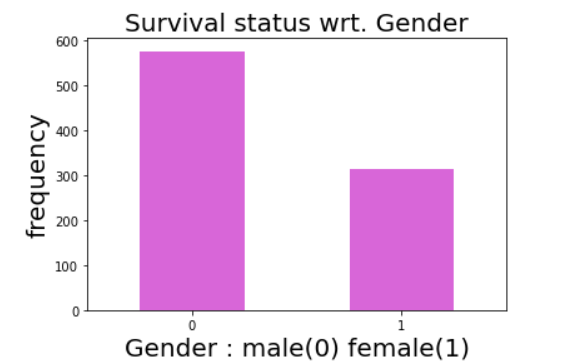


Fig. 3.4.10

#thus, the graph shows total men and women ratio on the ship.

data.age.plot(color='m',kind='box')

plt.ylabel('age frequencies',size=20)

plt.title('Age counts',size=20)

plt.show()

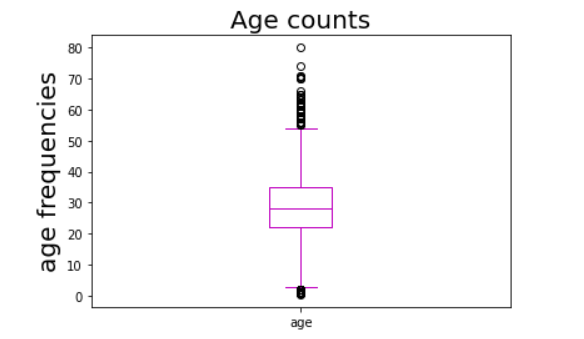


Fig. 3.4.11

#This shows that most of the passengers in the ship were of age group 30, and 25% passengers were

#less than 22 and 75% passengers were more than 22 age also shows some

#outliers(more than max. and less than min.)

data.pclass.value\_counts(normalize=True).plot(color='m',alpha=0.6,kind='bar')

plt.xticks(rotation=0)

plt.title("Total passengers of all classes",size=15)

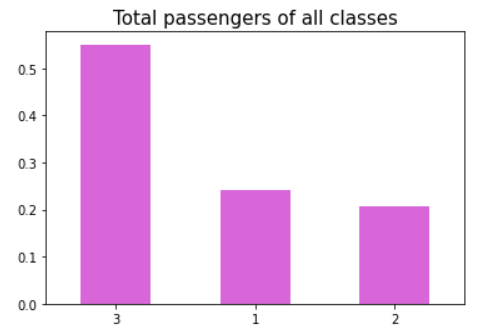


Fig. 3.4.14

#graphs shows total 1st,2nd and 3rd class people on the ship

# around 60% from 3rd class,and 20-20% of 1st and 2nd class onboarded

data.embarked.value\_counts(normalize=True).plot(color='m',alpha=0.6,kind='bar')

plt.xlabel('Southampton(0) Cherbourg(1) Queenstown(2)')

plt.ylabel('frequency')

plt.title("Embarked citites data")

plt.xticks(rotation=0)



Fig. 3.4.13

# thus, 70% passengers embarked from Southampton, 20% from Cherbourg and 10% from Queenstown

## Bivariate analysis

data.survived[data.sex==0].value\_counts(normalize=True).plot(color='m',alpha=0.6,kind='bar')

plt.xticks(rotation=0)

plt.title("Men Survived")

#thus, the graph shows that total of 15-20% of males were survived rest not.

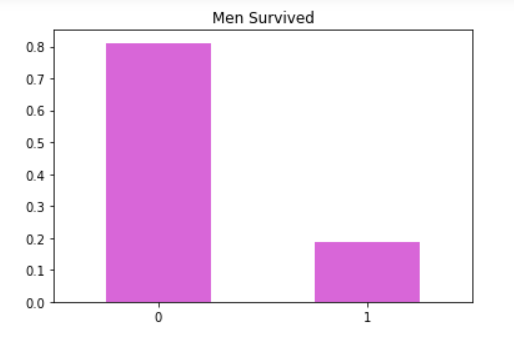


Fig. 3.4.14

data.survived[data.sex==1].value\_counts(normalize=True).plot(color='m',alpha=0.6,kind='bar')

plt.xticks(rotation=0)

plt.title("WoMen Survived")

# this graph shows that total of 70-75% of survivors were females and

#25-30% were deceased

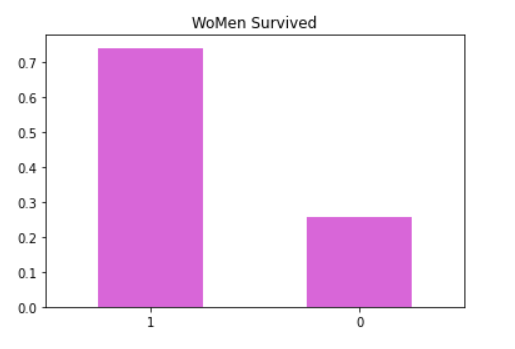


Fig. 3.4.15

for x in [1,2,3]:

data.survived[data.pclass==x].plot(kind='kde')

plt.xlabel('Survived')

plt.ylabel('class')

plt.title('class vs. Survived')

plt.legend(('1st','2nd','3rd'))

# graphs shows most of the passengers of 3rd class(green line) were deceased after 'em 2nd class and

#then first while another distribution on the right shows most of the passengers of

#1st class(blue line) were survived then 2nd then 1st

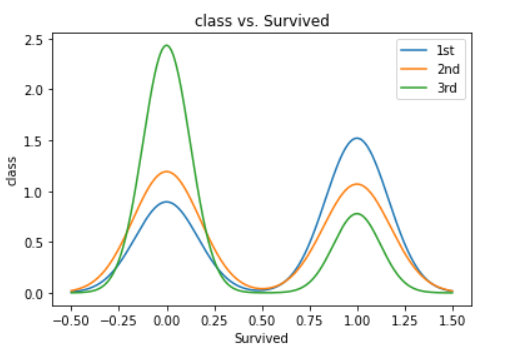


Fig. 3.4.16

for x in [1,2,3]:

data.age[data.pclass==x].plot(kind='kde')

plt.xlabel('age')

plt.ylabel('class')

plt.title("class vs. Age")

plt.legend(('1st','2nd','3rd'))

#blue line shows that 1st class most of the passengers have age 38-40

#orange line shows that 2nd class most of the passengers have age 25-30(most of them)

#green line shows that 3rd class passengers have age 20-25(mostly)

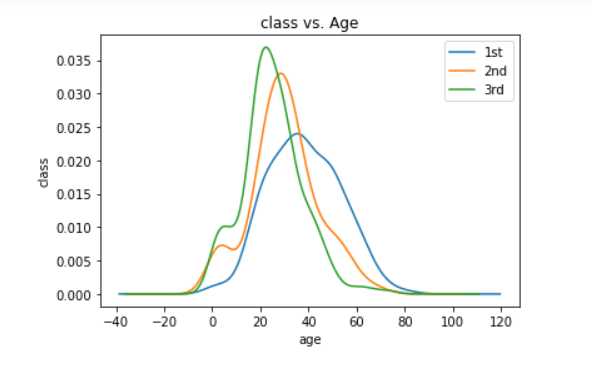
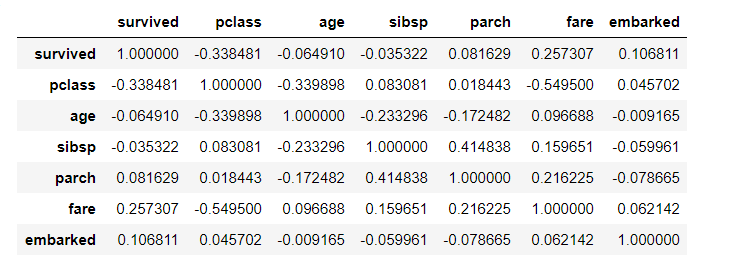


Fig. 3.4.17

data.corr()

#The above heatmap shows dependency between Sex and Survived and the correlation value is 0.54.

#And we can also see the overall distribution of every attribute with respect to that of other.

 Fig. 3.4.18

sns.heatmap(data.corr(),annot=True,cmap='magma')

plt.show()

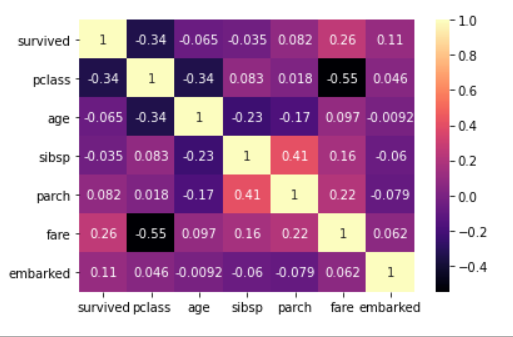


Fig. 3.4.19

**CHAPTER 4**

**EXPERIMENT AND RESULT ANALYSIS**

Of the four methods, Random Forest Classification performed the best and decision tree performed the worst. Table 1 offers a summary of the achievable accuracy using Logistic, Random Forest, Naïve Bayes and Decision Tree analysis. Even though we were given many features of passengers in our data, we found that most of the features were not useful in classification. For example, the number of sibling/spouses and the number of parents/children did not help with classification in any of the three models. Knowing the number of relatives aboard did not help with classification, but perhaps, if we were given the links between passengers then we’d be able to infer more about the survival rate. Since family units tend to all die or all survive, knowing the family links would have been useful.

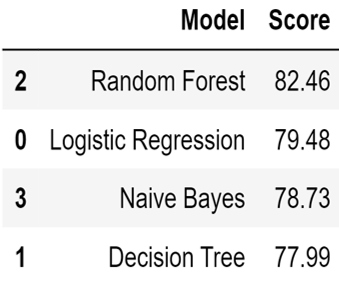


Fig. 4.1

**CHAPTER 5**

**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.

* 1. **Discussion**
     + What factors made people more likely to survive?
     + Were social-economic standing a factor in survival rate?
     + Did age, regardless of sex, determine your chances of survival?
     + Did women and children have preference to lifeboats (survival)?
     + Did women with children have a better survival rate vs women without children (adults 18+)?

1. The factors which are mainly affecting the survival rate are sex, class, age. If the sex is female, class is first and age is around 20-23 then there is higher survival rate.
2. Socio-economic status or the passenger class is a highly impacting attribute which is focusing the survival rate.
3. Yes, it is true your age regardless gender is impacting your survival rate as if age is 10 to 15 then there is a higher survival chances.
4. It is true according to the policy of that time women and children first would be saved, thus women and children have preference to lifeboats (survival)
5. Women with children have a better survival rate after them those who were without children.
   1. **Future Work**

The future work for the project is to make an interactive GUI interface so that it could be more easy to understand the conditions of survival at titanic in a more easy way and also it will also make the interest of the user to understand how we can utilize such kind of algorithms in our real world problem.

However, the missing values, survivor bias, and outliers introduced bias and affected the validity and accuracy of our study. For future studies, if we can incorporate several datasets of disasters that are similar to Titanic into one large dataset, it may allow us to gain additional insights on what features separate survivors and non-survivors, and possibly allow us to make predictions on disaster survival outcomes

It would be interesting to continue this analysis with other possible features or with other machine learning algorithms like random forests or other variants of Naïve Bayes.

**REFERENCES**

Ref. [1] <https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8>

Ref. [2] “Titanic: Machine Learning from Disaster”, 2017, Kaggle, from <https://www.kaggle.com/c/titanic>

“RMS Titanic”, 2017, Wikipedia, from <https://en.wikipedia.org/wiki/RMS_Titanic>

Foecke, Tim. "What really sank the Titanic?." *Materials Today* 11.10 (2008): 48.

Howells, Richard. "Titanic (1997)." *Film & History: An Interdisciplinary Journal of Film and Television Studies* 28.1 (1998): 70-71.

Singh, Aakriti, Shipra Saraswat, and Neetu Faujdar. "Analyzing Titanic disaster using machine learning algorithms." In 2017 International Conference on Computing, Communication and Automation (ICCCA), pp. 406-411. IEEE, 2017.