IBM PROJECT DOCUMENTATION

PLASMA DONOR APPLICATION

TEAM ID -PNT2002TMID29020

TEAM MEMBERS

SINDHU S

BHAGAVATHI AARTHI A

BHAGYA LAKSHMI V

EZHIL ARASI S

CHAPTER 1

1. INTRODUCTION

1.1 Project overview

Recent Covid-19 Pandemic has raised alarms over one of the most overlooked areas to focus: Healthcare Management. While healthcare management has various use cases for using data science, patient length of stay is one critical parameter to observe and predict if one wants to improve the efficiency of the healthcare management in a hospital. During the COVID 19 crisis, the requirement of plasma became a high priority and the donor count has become low. Saving the donor information and helping the needy by notifying the current donors list, would be a helping hand. In regard to the problem faced, an application is to be built which would take the donor details, store them and inform them upon a request.

1.1 Purpose

This system's goal is to use an web application to link donors and patients. Patient of this application may post requests for plasma donations or requests . The fundamental solution is to establish a centralized system is that a admin will keep track of current and previous Plasma Donation Events and also keep track of the location of the donor's plasma using google map.

CHAPTER 2

2. LITERATURE SURVEY

2.1 Existing Problem

- The already existing model is trained with minimal parameters by leaving the necessary parameter
- Low accuracy in prediction
- No feature extraction done
- High complexity.

2.2 References

1. Yang J.-J., Li J., Mulder J., Wang Y., Chen S., Wu H., Wang Q., Pan H. Emerginginformation technologies for enhanced healthcare. *Comput. Ind.*2015;69:3 11.doi:10.1016/j.compind.2015.01.012. [CrossRef] [Google Scholar]
2. Cortada J.W., Gordon D., Lenihan B. *The Value of Analytics in Healthcare.*IBM Institute for Business Value; Armonk, NY, USA: 2012. Report No.: GBE03476-USEN-00. [Google Scholar]
3. Center for Medicare and Medicaid Services. [(accessed on 1 August 2017)];Availableonline: https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends- andReports /NationalHealthExpendData/NationalHealthAccountsHistorical.html
4. Berwick D.M., Hackbarth A.D. Eliminating waste in US health care. *J. Am.Med. Assoc.* 2012;307:1513–1516. doi: 10.1001/jama.2012.362. [PubMed] [CrossRef] [Google Scholar]
5. Makary M.A., Daniel M. Medical error-the third leading cause of death inthe US. *Br. Med. J.* 2016;353:i2139. doi: 10.1136/bmj.i2139. [PubMed] [CrossRef] [Google Scholar]

Prokosch H.-U., Ganslandt T. Perspectives for medical informatics. *MethodsInf. Med.*2009;48:38–44. doi: 10.3414/ME9132. [PubMed] [CrossRef] [Google Scholar]

6. Simpao A.F., Ahumada L.M., Gálvez J.A., Rehman M.A. A review of analytics andclinical informatics in health care. *J. Med. Syst.* 2014;38:45.doi:
10.1007/s10916-014-0045-x. [PubMed] [CrossRef] [Google Scholar]

7. Ghassemi M., Celi L.A., Stone D.J. State of the art review: The data revolution incritical care. *Crit. Care.* 2015;19:118. doi: 10.1186/s13054-015-0801-4. [PMC free article] [PubMed] [CrossRef] [Google Scholar]

8. Tomar D., Agarwal S. A survey on Data Mining approaches for Healthcare. *Int. J.*
Bio-Sci. Bio-Technol. 2013;5:241–266. doi: 10.14257/ijbsbt.2013.5.5.25.[CrossRef] [Google Scholar]

9. Panagiota Galetsia , Korina Katsaliakia , Sameer Kumarb,∗ a School of Economics, Business Administration & Legal Studies, International HellenicUniversity, 14th km
Thessaloniki-N. Moudania, Thessaloniki, 57001, Greece b Opus College ofBusiness,University of St. Thomas Minneapolis Campus, 1000 LaSalle Avenue, SchulzeHall 435,
Minneapolis, MN 55403, USA

2.3 Problem Statement Definition

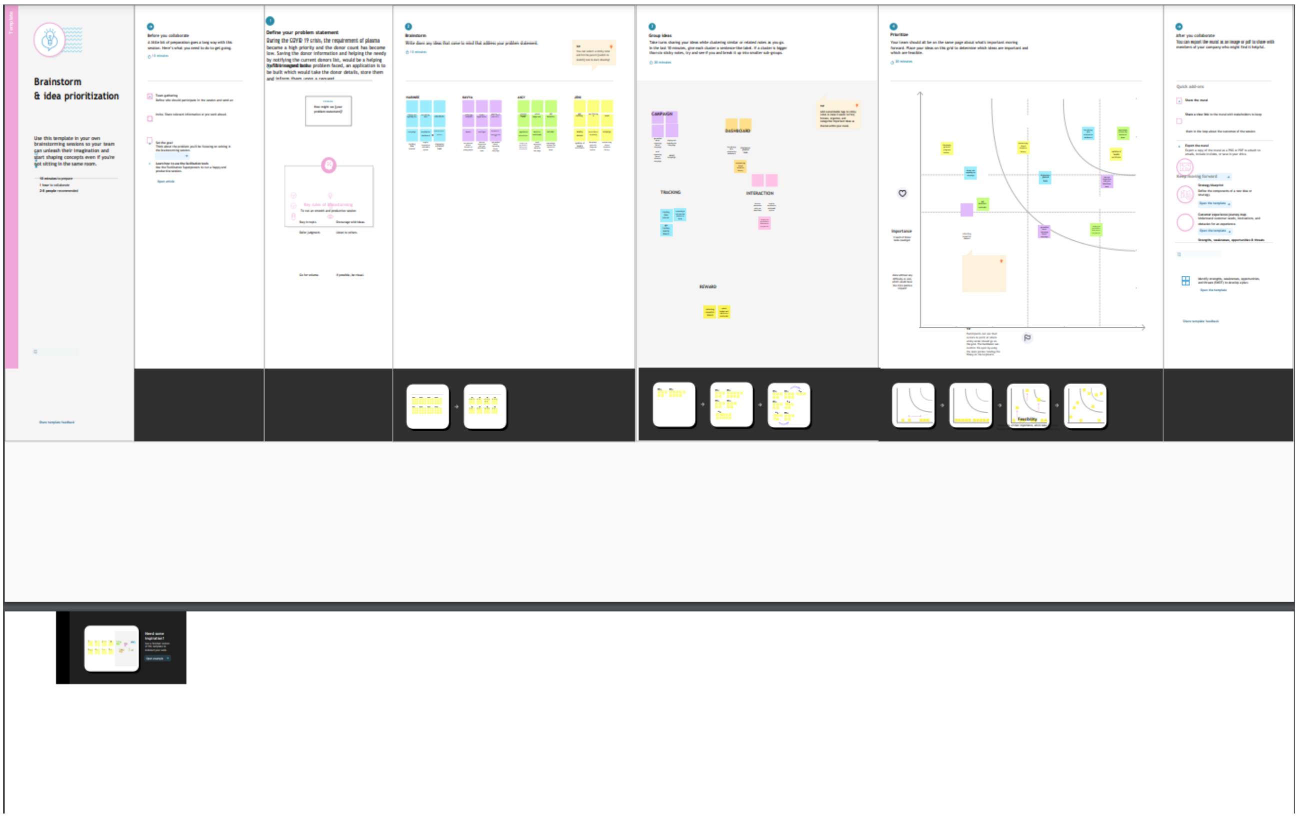
Many major medical conditions are treated by plasma. One of the most well-known techniques known as plasma treatment, plasma is used to cure various incurable diseases. As there were no vaccines available to treat the infected patients during the Covid-19 emergency, the need for plasma increased dramatically. Plasma therapy had a high probability of recovery but a very low donor count, therefore it was crucial to learn more about the donors in these circumstances. It would be helpful to save the contributor information and let clients know about the recurring donors because it can help them find the crucial information more quickly.

CHAPTER 3

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

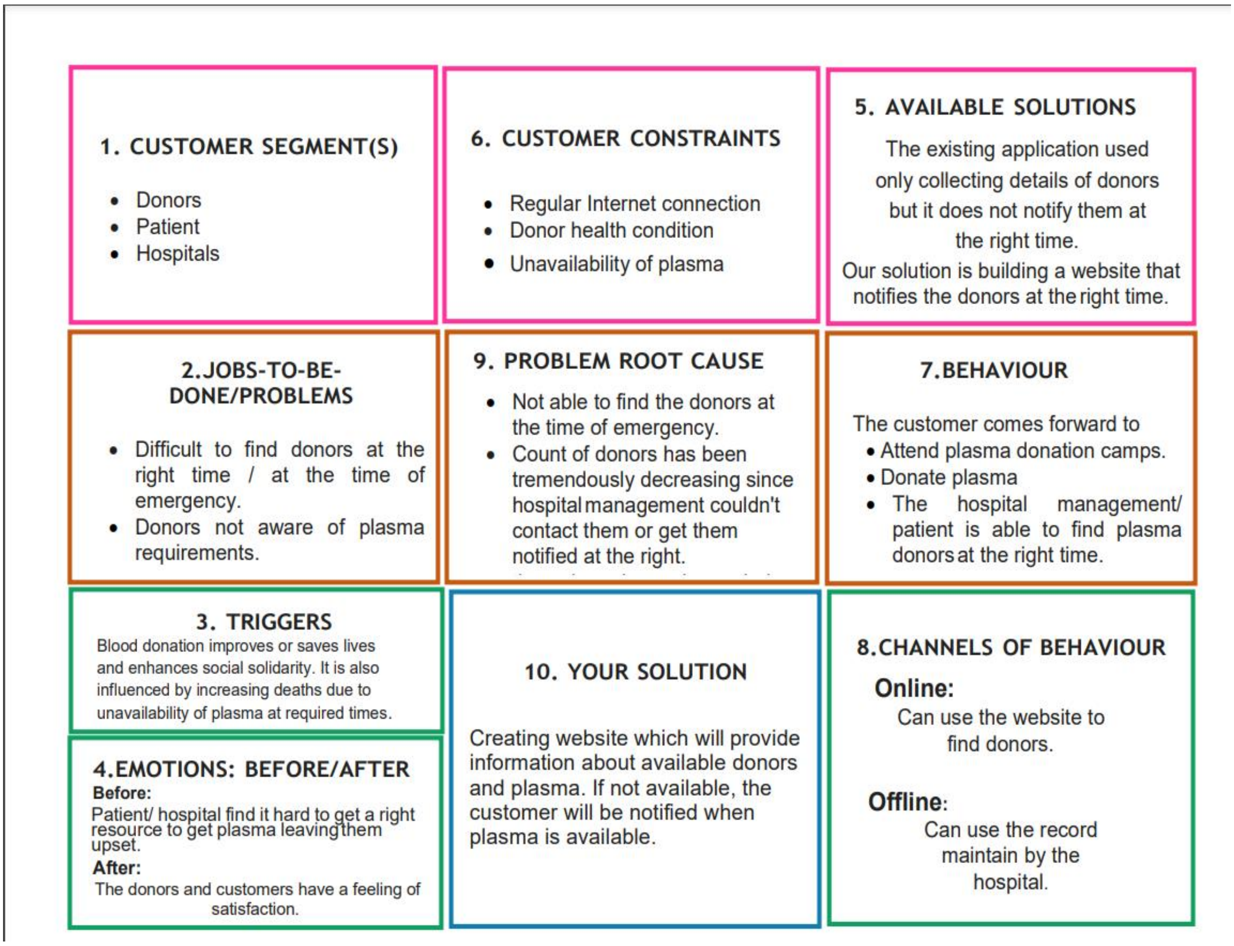
3.2 Ideation & Brainstorming



3.3 Proposed Solution

The length of the stay can be predicted using either Random forest or Decision Tree for more accuracy. Certain parameters like age, stage of the diseases, disease diagnosis, severity of illness, type of admission, facilities allocated, etc., are used for prediction. IBM Cognos will be used for data analytic s. The model will be trained using colab. It predicts the length of stay (LOS) of the patients with more accuracy. As a result proper resources and therapy can be provided . Patients can get proper treatment and better medical care than before which helps them for their faster recovery. So the prediction minimizes the overflow of patients and helps in resource management and optimize their resource utilization. Hence this leads to faster recovery and lower the expenses for treatment. It improves the trust in hospital management. It avoids the major risk of spreadinginfection among the hospital staff. This leads to overall safety of hospital staff and patients. Resource consumption is optimized. This model can be used by all government hospitals, private hospitals, and even in the model is trained with the real world hospital survey for better prediction small clinics. Length of the stay will be predicted with more accuracy.This model predicts the length of the stay for all kinds of patients and predicts with more accuracy.

3.4 Problem Solution fit



CHAPTER 4

4.REQUIREMENT ANALYSIS

4.1 Functional requirement

FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FRNo.	FunctionalRequirement(Epic)	SubRequirement(Story/Sub-Task)
FR-1	User Registration	Registration through Form(WebApp)
FR-2	User Confirmation	Confirmation via Email
FR-3	Certification	After the donor donates plasma, we will give them a certificate of appreciation and authentication.
FR-4	Statistical data	The availability of plasma is given in the page as stats,which will be helpful for the users.
FR-5	User Plasma Request	Users can request to donate plasma by filling out the request form on the page.
FR-6	Searching/reporting requirements	Users can use the search bar to lookup information about camps and other topics.

4.2 Non-Functional requirements

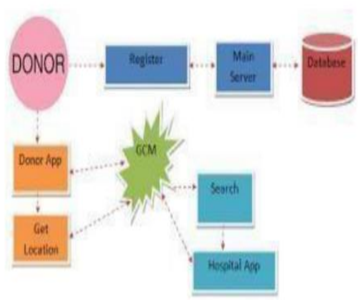
NFR-4	**Performance**	Users should have a proper Internet Connection.
NFR-5	**Availability**	The system including the online and offline components should be available 24/7.
NFR-6	**Scalability**	The application has the ability to handle growing number of users and load without compromising on Performance and causing disruptions to user experience.

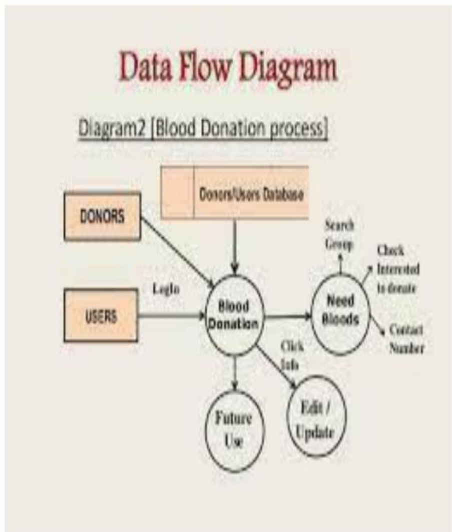
CHAPTER 5

5.PROJECT DESIGN

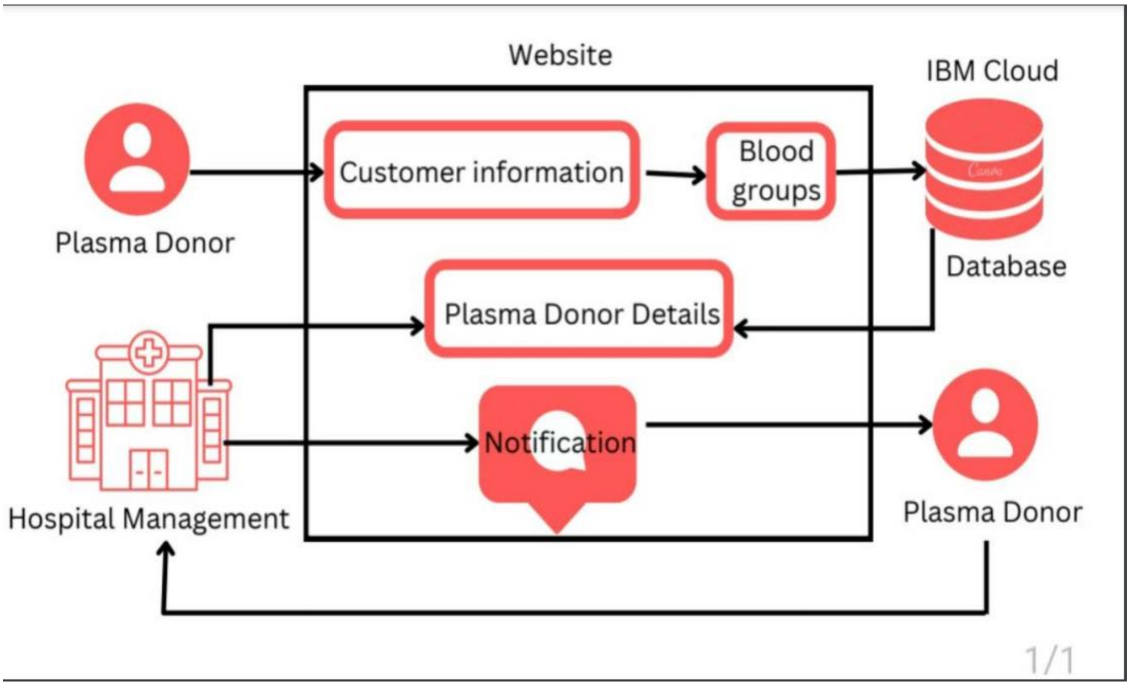
5.1 Data Flow Diagrams

A data flow diagram is a traditional visual representation of the information flows within asystem. A neat and clear DFD can depict the right amount of the system requirement graphically.it shows how data enters and leaves the system, what changes the information and, where data is stored





5.2 Solution & Technical Architecture



5.3 USER STORIES

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation creation	USN-1	Connect with python code	2	High	Srinithi K Sutharshini PR Swathi A Yuvpriya G
Sprint-2	Software	USN-2	Creating an IBM Watsonin Cloud platform	2	High	Srinithi K Sutharshini PR Swathi A Yuvpriya G
Sprint-3	MIT App Inventor	USN-3	Develop an Plasma donor application	2	High	Srinithi K Sutharshini PR Swathi A Yuvpriya G
Sprint-4	Dashboard	USN-4	Design the Modules andtest the app	2	High	Srinithi K Sutharshini PR Swathi A Yuvpriya G
Sprint-5	Web UI	USN-5	To make the user to interact with software.	2	High	Srinithi K Sutharshini PR Swathi A Yuvpriya G

CHAPTER 6

6. PROJECT PLANNING

6.1 SPRINT PLANNING & ESTIMATIONS

6.2 SPRINT DELIVERY SCHEDULE

Project Planning Phase - Project Planning Template
Sprint Delivery plan

Date	06 November 2022
Team ID	PNT2022TMID20573
Project Name	Plasma Donor Application
Marks	4 Marks

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	05 Nov 2022	05 Nov 2022	20	05 Nov 2022
Sprint-2	20	6 Days	11 Nov 2022	11 Nov 2022	20	11 Nov2022
Sprint-3	20	6 Days	16 Nov 2022	16 Nov 2022	20	16 Nov 2022
Sprint-4	20	6 Days	22 Nov 2022	22 Nov 2022	20	22 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day) .

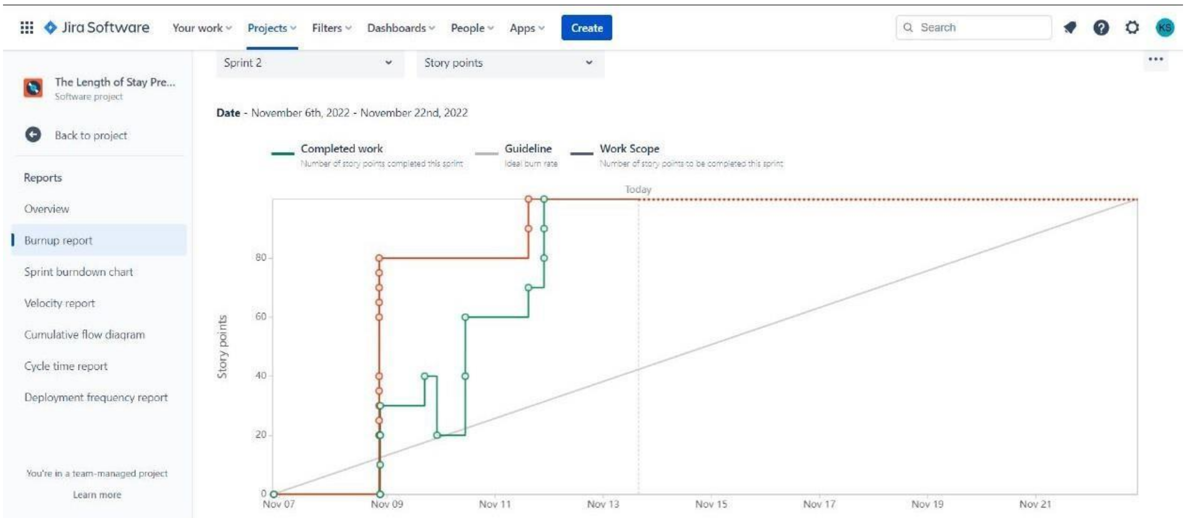
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:

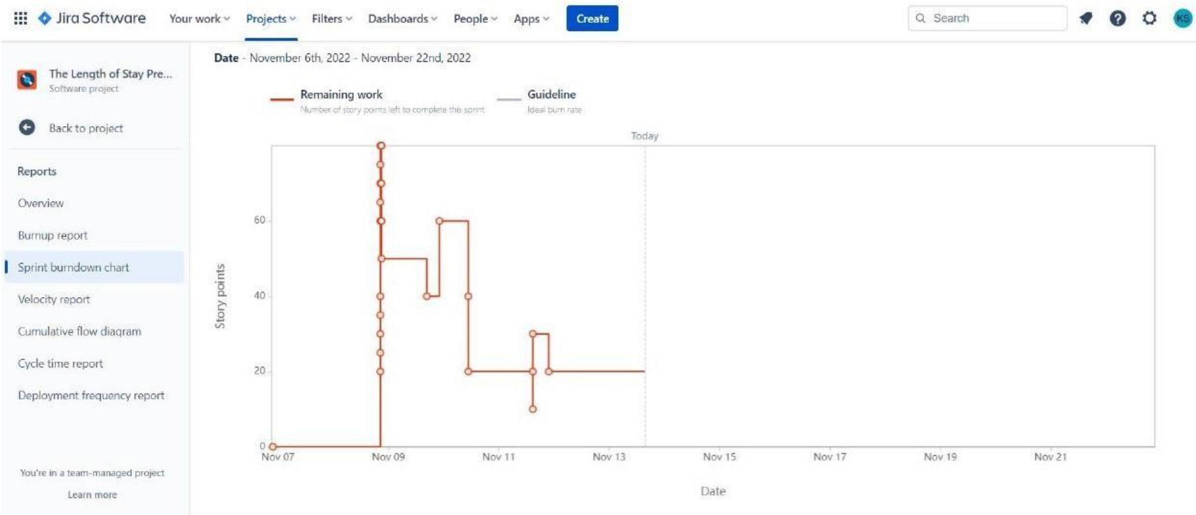
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

6.3 Reports from JIRA

Burnt Up Chart



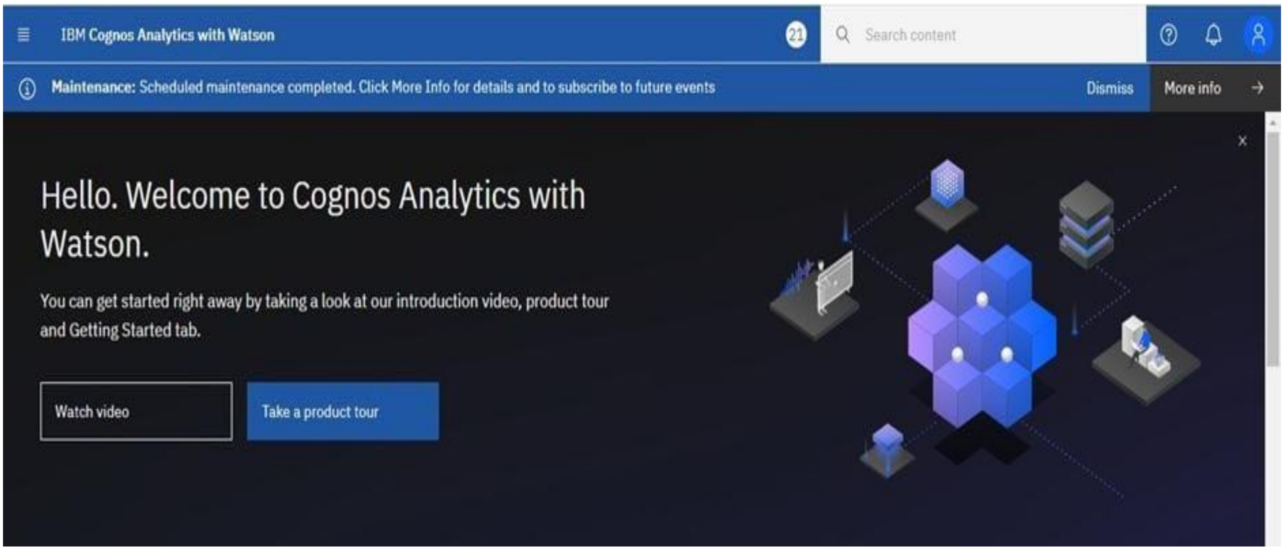
Burnt Down Chart



CHAPTER 7

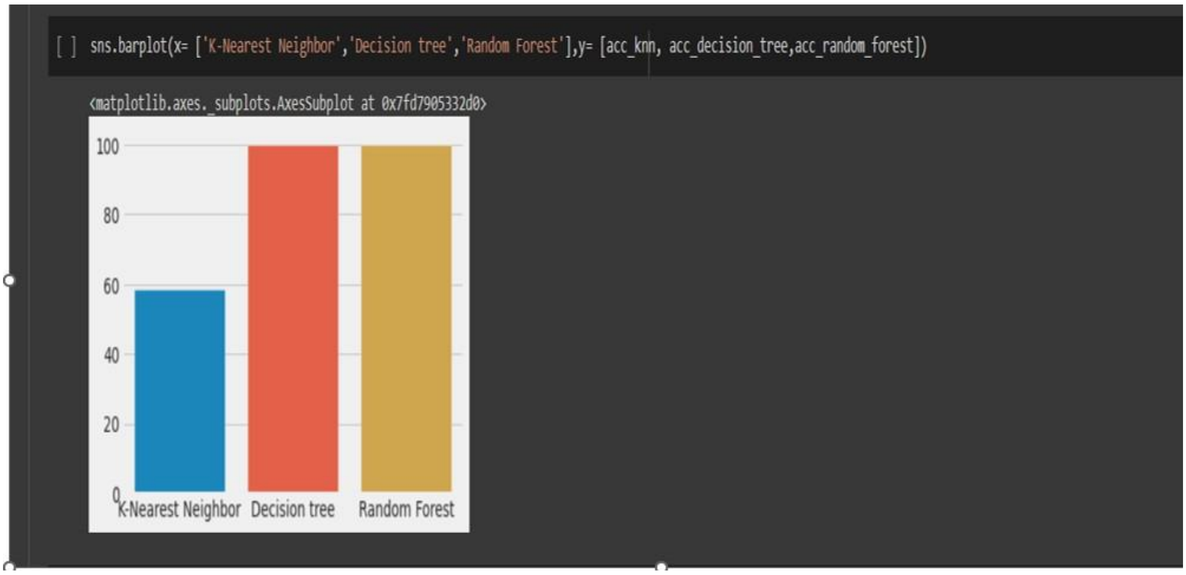
7.CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1



7.2RESULTS

7.2.1Performance Metrics



CHAPTER 8

ADVANTAGES & DISADVANTAGES

Advantages

- Analyzing clinical data to improve medical research
- Using patient data to improve health outcomes
- Gaining operational insights from healthcare provider data
- Improved staffing through health business management analytics
- Research and prediction of disease.
- Automation of hospital administrative processes.
- Early detection of disease.
- Prevention of unnecessary doctor's visits.
- Discovery of new drugs.
- More accurate calculation of health insurance rates.
- More effective sharing of patient data.

Disadvantages

Replacing Medical Personnel

Application of technology in every sphere of human life is improving the way things are done. These technologies are are also posing some threat to world of works. Robotics are replacing human labor.

Data Safety

Data security is another challenge in applying big data in healthcare. Big data storage is usually targets of hackers. This endangers the safety of medical data. Healthcare organisations are very much concerned about the safety of patients' sensitive personal data. For this, all healthcare applications must meet the requirement for data security and be HIPAA compliant before they can be deployed for healthcare service.

CHAPTER 9

CONCLUSION

Analytics is the science of analyzing raw datasets in order to derive conclusion regarding the information they hold. It enables us to discover patterns in the raw data and draw valuable information from them. To some, the domain of healthcare data analytics may look new, but it has a lot of potential, especially if you wish to engage in challenging job roles and build a strong data analytics profile in the upcoming years. In this blog, we have covered some of the major topics such as what is healthcare data analytics, its applications, scope, and benefits, etc. We hope it helps you in your decision-making as a healthcare data analytics professional.

CHAPTER 10

FUTURE SCOPE

The Future of Healthcare, Intel provides a foundation for big data platforms and AI to advance health analytics. Predictive data analytics is helping health organizations enhance patient care, improve outcomes, and reduce costs by anticipating when, where, andhow care should be provided. The future of big data in healthcare will be determined by technological breakthroughs from 2022 to 2030.
Complete patient care and cost-effective prescription procedures are required for population health management. To assess clinical and claims data, they must be combined on the same platform.

Countries around the world have started to invest more capital in medical infrastructure, pharmaceuticals, and healthcare smart analytics solutions. The market is growing and will continue to expand, given the benefits of healthcare data analytics. It has also risen as a good career option for fresh data science and data analytics graduates or professionals who wish to build their career in the healthcare sector. Due to the sensitivity of the profession, the salary offers for healthcare data analysts are lucrative around the world.
Apart from the remuneration, the opportunities to work with some of the biggest names in the healthcare sector is also worth mentioning. Hence, healthcare data analytics is growing to be one of the most rewarding branches of data analytics in the coming future.

APPENDIX

Source Code

Importing required Packages

In [72]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style("darkgrid")
plt.style.use("dark_background")
```

Importing the dataset

In [73]:
```
train = pd.read_csv('/content/input/training_data.csv')
test = pd.read_csv('/content/input/testing_data.csv')
Paramters_Description = pd.read_csv('/content/input/parameter_description.csv')
sample = pd.read_csv('/content/input/testing_target.csv')
```

Viewing dataset

In [74]:
```
train.head(5)
```

Out[74]:

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Hospital_region_code	Available_Extra_Rooms_in_Hospital	Department	Ward_Type	Ward_Facility_Code	Bed_Grade
0	1	8	c	3	Z	3	radiotherapy	R	F	2.(
1	2	2	c	5	Z	2	radiotherapy	S	F	2.(
2	3	10	e	1	X	2	anesthesia	S	E	2.(
3	4	26	b	2	Y	2	radiotherapy	R	D	2.(
4	5	26	b	2	Y	2	radiotherapy	S	D	2.(

Dataset Column Description

```
Paramters_Description
```

	Column	Description
0	case_id	It is identity number given by hospital admini...
1	Hospital_code	It is the code (identity number) given to the ...
2	Hospital_type_code	It is the unique code given to the type of hos...
3	City_Code_Hospital	It is the code given to the city where the hos...
4	Hospital_region_code	It is the code given to the region where the h...
5	Available_Extra_Rooms_in_Hospital	It will display the number of rooms that are s...
6	Department	The department that is overlooking the patient...
7	Ward_Type	The unique code given to the type of ward to w...
8	Ward_Facility_Code	The unique code given to the facility in the w...
9	Bed_Grade	It is the quality or condition of the bed in t...
10	patientid	It is the unique identity value given to the p...
11	City_Code_Patient	It is the unique identity code given to the ci...
12	Type_of_Admission	It is the admission type registered in the hos...
13	Severity_of_Illness	It is the severity level of the patients' illn...
14	Visitors_with_Patient	Number of the visitors with the patients to ta...
15	Age	It is the age of patients. It is given in peri...
16	Admission_Deposit	It is the deposit amount that the patient paid...
17	Stay	It is the Length Of Stay (LOS) of patients. I...

Analysis of dataset

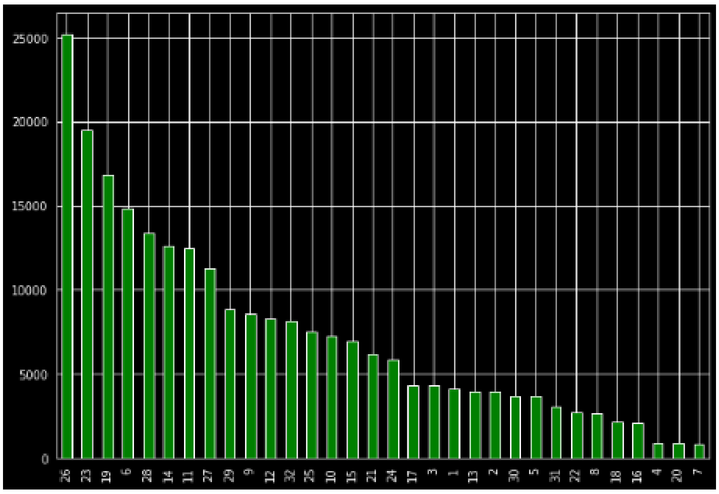
Distribution of values

Hospital_code

```
train.Hospital_code.value_counts()
```

```
26    25225
23    19505
19    16825
6     14847
28    13341
14    12594
11    12454
27    11312
29     8828
9      8558
12     8312
32     8166
25     7529
10     7257
15     6965
21     6226
24     5863
17     4319
3      4308
1      4111
13     3974
2      3940
30     3707
5      3684
31     3051
22     2740
8      2679
18     2164
16     2119
4       937
20      905
7       864
Name: Hospital_code, dtype: int64
```

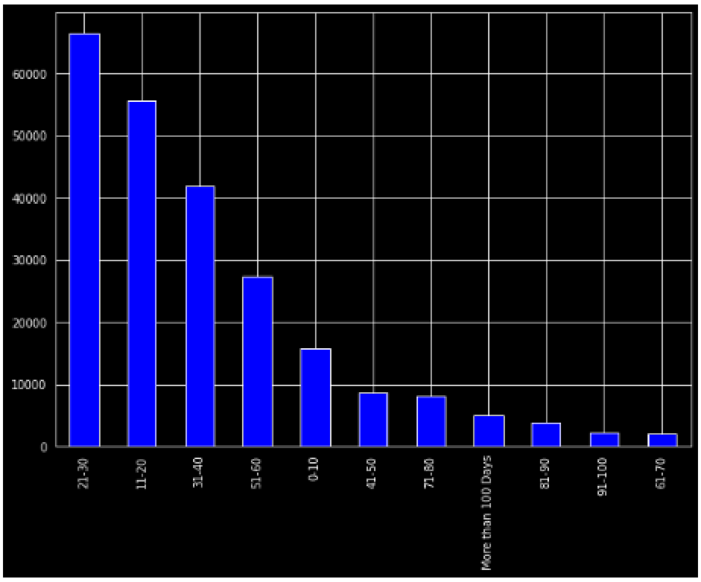
```
plt.figure(figsize=(10,7))
train.Hospital_code.value_counts().plot(kind="bar", color = ['green'])
```



Stay

```
train.Stay.value_counts()
```

```
21-30                 66497
11-20                 55691
31-40                 41951
51-60                 27458
0-10                  15866
41-50                  8665
71-80                  8061
More than 100 Days     5029
81-90                  3821
91-100                 2179
61-70                  2090
Name: Stay, dtype: int64
```



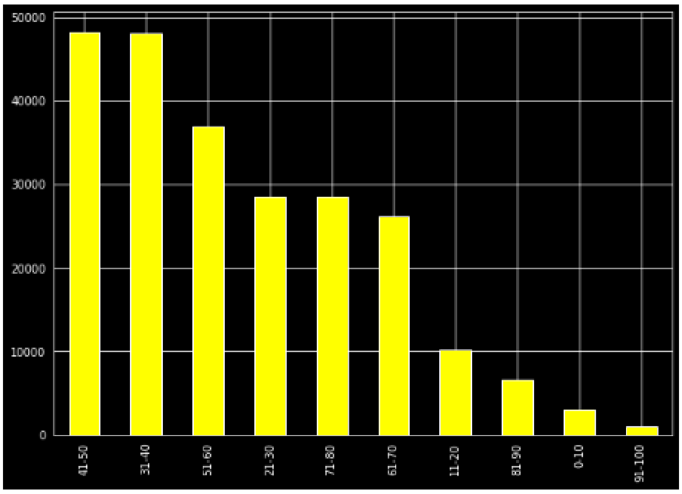
Age

```
train.Age.value_counts()
```

```
41-50     48272
31-40     48106
51-60     36969
21-30     28555
71-80     28552
61-70     26139
11-20     10141
```

```
81-90      6578
0-10       3030
91-100      966
Name: Age, dtype: int64
```

```
#Age distribution
plt.figure(figsize=(10,7))
train.Age.value_counts().plot(kind="bar", color = ['Yellow'])
```



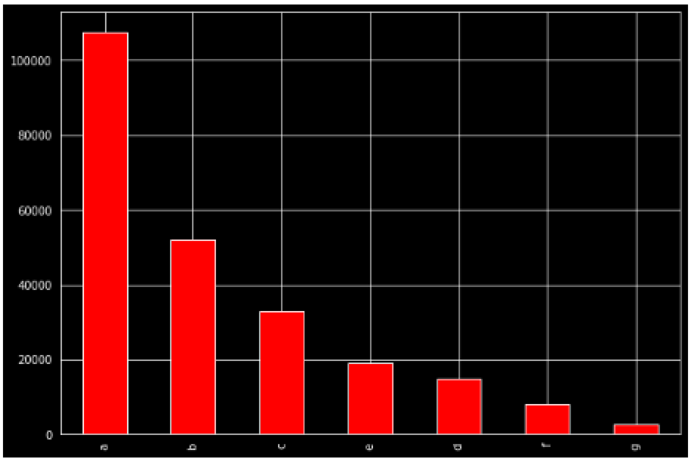
Hospital_type_code

```
train.Hospital_type_code.value_counts()
```

```
a     107545
b      51925
```

```
c      32995
e      19105
d      14833
f       8166
g       2740
Name: Hospital_type_code, dtype: int64
```

```
#Hospital_type_code distribution
plt.figure(figsize=(10,7))
train.Hospital_type_code.value_counts().plot(kind="bar", color = ['Red'])
```

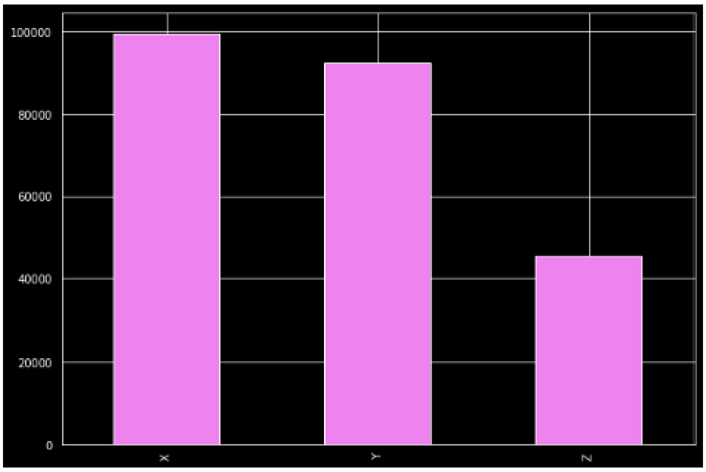


Hospital_region_code

```
train.Hospital_region_code.value_counts()
```

```
X     99568
Y     92214
Z     45527
Name: Hospital_region_code, dtype: int64
```

```
#Hospital_region_code distribution
plt.figure(figsize=(10,7))
train.Hospital_region_code.value_counts().plot(kind="bar", color = ['Violet'])
```



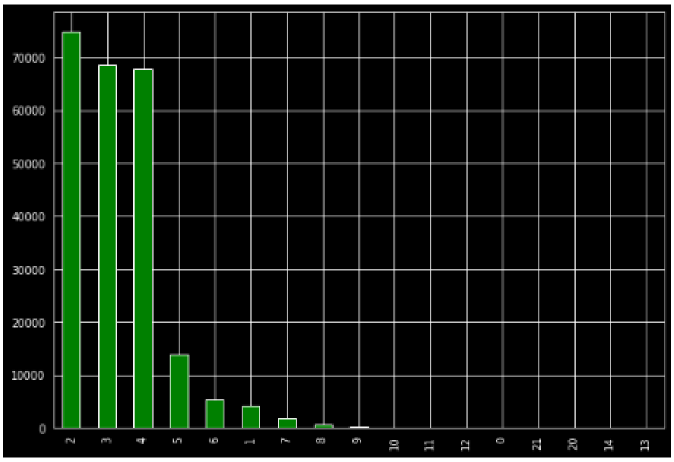
Available_Extra_Rooms_in_Hospital

```
train.Available_Extra_Rooms_in_Hospital.value_counts()
```

```
2      74877
3      68517
4      67756
5      13879
6       5344
1       4208
7       1876
8        622
9        144
10        46
```

```
11      13
12      11
0       11
21       2
20       1
14       1
13       1
Name: Available_Extra_Rooms_in_Hospital, dtype: int64
```

```
#Available_Extra_Rooms_in_Hospital distribution
plt.figure(figsize=(10,7))
train.Available_Extra_Rooms_in_Hospital.value_counts().plot(kind="bar", color = ['green'])
```



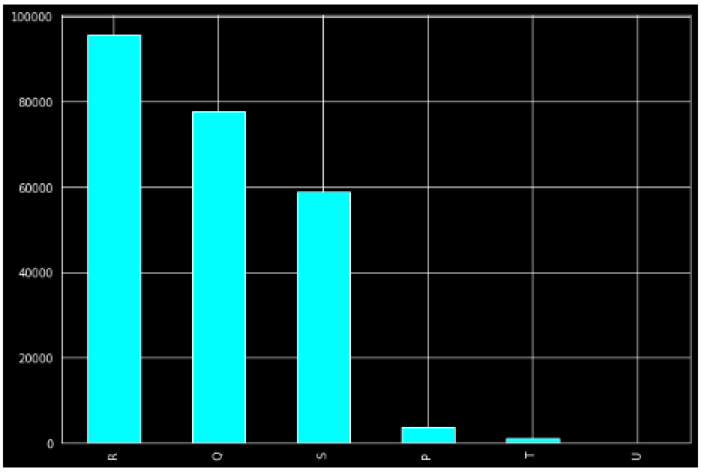
Department

```
train.Department.value_counts()
```

```
gynecology            185062
```

```
R     95788
Q     77707
S     59022
P      3691
T      1092
U         9
Name: Ward_Type, dtype: int64
```

```
#Ward_Type distribution
plt.figure(figsize=(10,7))
train.Ward_Type.value_counts().plot(kind="bar", color = ['cyan'])
```



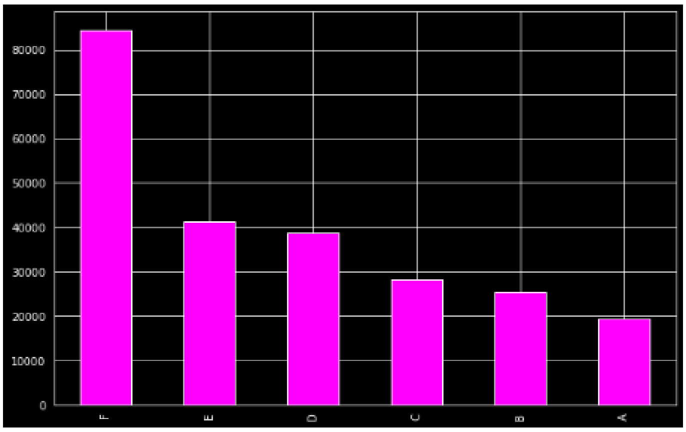
Ward_Facility_Code

```
train.Ward_Facility_Code.value_counts()
```

```
F     84438
E     41246
```

```
D     38584
C     28137
B     25493
A     19411
Name: Ward_Facility_Code, dtype: int64
```

```
#Ward_Facility_Code distribution
plt.figure(figsize=(10,7))
train.Ward_Facility_Code.value_counts().plot(kind="bar", color = ['magenta'])
```



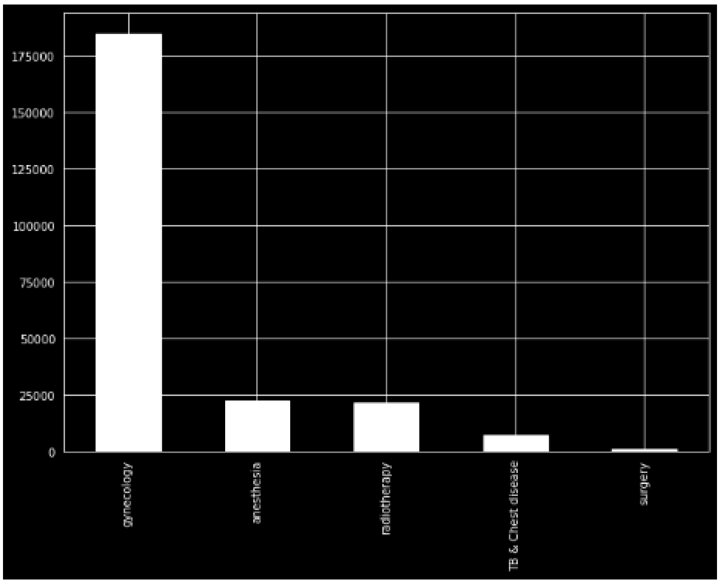
Visitors_with_Patient

```
train.Visitors_with_Patient.value_counts()
```

```
2.0     103037
4.0      59068
3.0      43860
6.0      14211
5.0       6992
```

```
anesthesia              22557
radiotherapy            21725
TB & Chest disease       7017
surgery                   948
Name: Department, dtype: int64
```

```
#Department distribution
plt.figure(figsize=(10,7))
train.Department.value_counts().plot(kind="bar", color = ['white'])
```

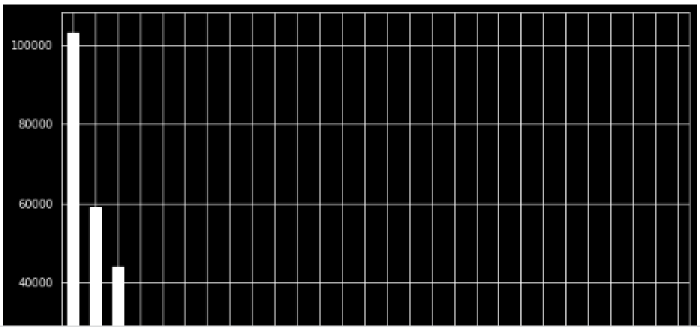


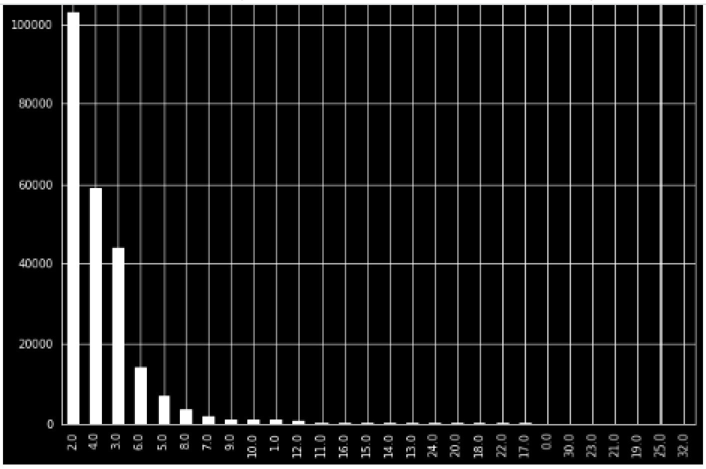
Ward_Type

```
train.Ward_Type.value_counts()
```

```
8.0       3662
7.0       1888
9.0       1024
10.0       882
1.0        871
12.0       757
11.0       242
16.0       220
15.0       146
14.0       138
13.0        84
24.0        63
20.0        46
18.0        35
22.0        16
17.0        15
0.0         13
30.0         9
23.0         8
21.0         8
19.0         6
25.0         6
32.0         1
Name: Visitors_with_Patient, dtype: int64
```

```
#Visitors_with_Patient distribution
plt.figure(figsize=(10,7))
train.Visitors_with_Patient.value_counts().plot(kind="bar", color = ['white'])
```



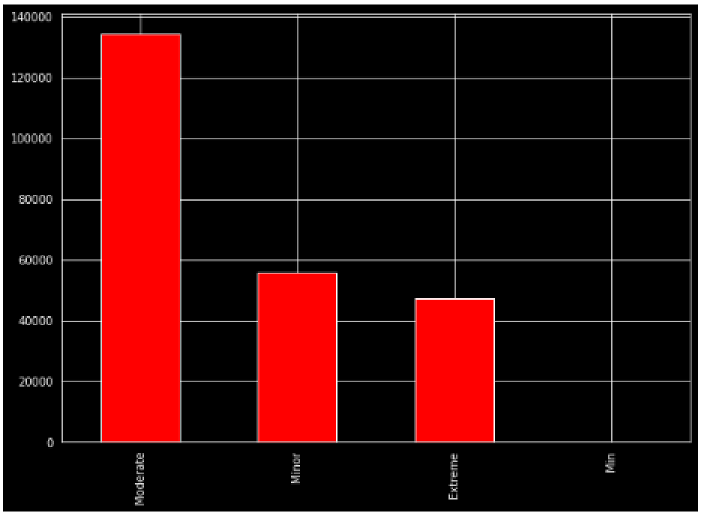


Severity of Illness

```
train.Severity_of_Illness.value_counts()
```

```
Moderate    134324
Minor        55665
Extreme      47319
Min              1
Name: Severity_of_Illness, dtype: int64
```

```
#Severity_of_Illness distribution
plt.figure(figsize=(10,7))
train.Severity_of_Illness.value_counts().plot(kind="bar", color = ['red'])
```



Unique values of columns

```
for features in train.columns:
    print('*-----------------------------------------------------------------------------------------------------*')
    print(' Unique Values for {}'.format(features))
    print(train[features].unique())
    print('*-----------------------------------------------------------------------------------------------------*')
    print()
```

```
*-----------------------------------------------------------------------------------------------------*
 Unique Values for case_id
[     1      2      3 ... 237307 237308 237309]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Hospital_code
[ 8  2 10 26 23 32  1 22 16  9  6 29 12  3 21 28 27 19  5 14 13 31 24 17
 25 15 11 30 18  4  7 20]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Hospital_type_code
['c' 'e' 'b' 'a' 'f' 'd' 'g']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for City_Code_Hospital
[ 3  5  1  2  6  9 10  4 11  7 13]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Hospital_region_code
['Z' 'X' 'Y']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Available_Extra_Rooms_in_Hospital
[ 3  2  1  4  6  5  7  8  9 10 12  0 11 20 14 21 13]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Department
['radiotherapy' 'anesthesia' 'gynecology' 'TB & Chest disease' 'surgery']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Ward_Type
['R' 'S' 'Q' 'P' 'T' 'U']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Ward_Facility_Code
['F' 'E' 'D' 'B' 'A' 'C']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Bed_Grade
[ 2.  3.  4.  1. nan]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for patientid
[31397 63418  8088 ... 37502 73756 21763]
*                                                                                                     *
```

```
*-----------------------------------------------------------------------------------------------------*
 Unique Values for City_Code_Patient
[ 7.  8.  2.  5.  6.  3.  4.  1.  9. 14. nan 25. 15. 12. 10. 28. 24. 23.
 20. 11. 13. 21. 18. 16. 26. 27. 22. 19. 31. 34. 32. 30. 29. 37. 33. 35.
 36.]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Type_of_Admission
['Emergency' 'Trauma' 'Urgent']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Severity_of_Illness
['Extreme' 'Moderate' 'Minor' 'Min']
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Visitors_with_Patient
[ 2.  4.  3.  8.  6.  7. 13.  5.  1. 10. 15. 11. 12.  9. 24. 16. 14. 20.
  0. 19. 18. 17. 23. 21. 32. 30. 22. 25. nan]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Age
['51-60' '71-80' '31-40' '41-50' '81-90' '61-70' '21-30' '11-20' '0-10'
 '91-100' nan]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Admission_Deposit
[4911. 5954. 4745. ... 2710. 2236.   nan]
*-----------------------------------------------------------------------------------------------------*

*-----------------------------------------------------------------------------------------------------*
 Unique Values for Stay
['0-10' '41-50' '31-40' '11-20' '51-60' '21-30' '71-80'
 'More than 100 Days' '81-90' '61-70' '91-100' nan]
*-----------------------------------------------------------------------------------------------------*
```

Data Preprocessing & Feature Engineering

The following features may have relevance with the Length of Stay of a patient

Department: It Relates to the type of disease. Hence it will have impact on the length of stay of the patients

Type of Admission: It Relates to patients' reason of admission to the hospital and definitely it will have impact on length of stay opf the patients

Severity of Illness: It Relates to the curability of disease

Age: Relates to the curability of diseaseThe following features may have relevance with the Length of Stay of a patient

Department: It Relates to the type of disease. Hence it will have impact on the length of stay of the patients

Type of Admission: It Relates to patients' reason of admission to the hospital and definitely it will have impact on length of stay opf the patients

Severity of Illness: It Relates to the curability of disease

Age: Relates to the curability of disease

Ward_Type: Relates to the curability of disease

\

The following features doesn't have relevance with the Length Of Stay(LOS) of Patients

Hospital_region_code: It is code given to the hospital region which is irrelevent to the Length of Stay.

Bed Grade: It is the grade given to the quality of the bed in ward it is also irrelevent to the length of stay.

patientid: It is the identity number or code given for the identification of the patient which is irrelevant to the length of stay.

City_Code_Patient: It is the city code and irrelevant to the length of stay of patients.

```python
"""
as 'Hospital_region_code', 'Bed_Grade', 'patientid', 'City_Code_Patient' are irrelevant to the health or
length of stay of patients so lets drop these parameters from training and testing dataset to improve the performace of model (high accuracy)
by reducing the complexity
"""

train = train.drop(['Hospital_region_code', 'Bed_Grade', 'patientid', 'City_Code_Patient'], axis = 1)
test = test.drop(['Hospital_region_code', 'Bed_Grade', 'patientid', 'City_Code_Patient'], axis = 1)
```

```python
# Combine test and train dataset for processing
combined = [train, test]
combined
```

```
[        case_id  Hospital_code Hospital_type_code  City_Code_Hospital  \
 0             1              8                  c                   3
 1             2              2                  c                   5
 2             3             10                  e                   1
 3             4             26                  b                   2
 4             5             26                  b                   2
 ...         ...            ...                ...                 ...
 237304   237305             23                  a                   6
 237305   237306             19                  a                   7
 237306   237307              8                  c                   3
 237307   237308             21                  c                   3
 237308   237309              5                  a                   1

         Available_Extra_Rooms_in_Hospital    Department Ward_Type  \
 0                                       3  radiotherapy         R
 1                                       2  radiotherapy         S
 2                                       2    anesthesia         S
 3                                       2  radiotherapy         R
 4                                       2  radiotherapy         S
 ...                                   ...           ...       ...
 237304                                  3    gynecology         R
 237305                                  2    gynecology         R
 237306                                  5    gynecology         Q
 237307                                  4  radiotherapy         S
 237308                                  3    gynecology         Q

        Ward_Facility_Code Type_of_Admission Severity_of_Illness  \
 0                       F         Emergency             Extreme
 1                       F            Trauma             Extreme
 2                       E            Trauma             Extreme
 3                       D            Trauma             Extreme
 4                       D            Trauma             Extreme
 ...                   ...               ...                 ...
 237304                  F            Trauma             Extreme
 237305                  C         Emergency             Extreme
237306                  F         Emergency               Minor
237307                  A         Emergency               Minor
237308                  E            Trauma                 Min

        Visitors_with_Patient    Age  Admission_Deposit   Stay
0                         2.0  51-60             4911.0   0-10
1                         2.0  51-60             5954.0  41-50
2                         2.0  51-60             4745.0  31-40
3                         2.0  51-60             7272.0  41-50
4                         2.0  51-60             5558.0  41-50
...                       ...    ...                ...    ...
237304                    5.0  41-50             4298.0  51-60
237305                    4.0  41-50             4165.0  31-40
237306                    4.0  31-40             5075.0  21-30
237307                    2.0  31-40             5179.0  11-20
237308                    NaN    NaN                NaN    NaN

[237309 rows x 14 columns],
        case_id  Hospital_code Hospital_type_code  City_Code_Hospital  \
0        318439             21                  c                   3
1        318440             29                  a                   4
2        318441             26                  b                   2
3        318442              6                  a                   6
4        318443             28                  b                  11
...         ...            ...                ...                 ...
137052   455491             11                  b                   2
137053   455492             25                  e                   1
137054   455493             30                  c                   3
137055   455494              5                  a                   1
137056   455495              6                  a                   6

        Available_Extra_Rooms_in_Hospital    Department Ward_Type  \
0                                       3    gynecology         S
1                                       2    gynecology         S
2                                       3    gynecology         Q
3                                       3    gynecology         Q
4                                       2    gynecology         R
...                                   ...           ...       ...
137052                                  4    anesthesia         Q
137053                                  2  radiotherapy         R
137054                                  2    anesthesia         R
137055                                  2    anesthesia         R
137056                                  3    gynecology         Q

       Ward_Facility_Code Type_of_Admission Severity_of_Illness  \
0                       A         Emergency            Moderate
1                       F            Trauma            Moderate
2                       D         Emergency            Moderate
3                       F            Trauma            Moderate
```

```
4                         F          Trauma           Moderate
...                     ...             ...                ...
137052                    D       Emergency              Minor
137053                    E       Emergency           Moderate
137054                    A          Urgent              Minor
137055                    E          Trauma              Minor
137056                    F          Trauma            Extreme

        Visitors_with_Patient    Age  Admission_Deposit
0                           2  71-80               3095
1                           4  71-80               4018
2                           3  71-80               4492
3                           3  71-80               4173
4                           4  71-80               4161
...                       ...    ...                ...
137052                      4  41-50               6313
137053                      2   0-10               3510
137054                      2   0-10               7190
137055                      2  41-50               5435
137056                      5  51-60               4702

[137057 rows x 13 columns]]
```

Lets encode the categorical data for traning the model

```python
# Encoding Department
from sklearn.preprocessing import LabelEncoder

for dataset in combined:
    label = LabelEncoder()
    dataset['Department'] = label.fit_transform(dataset['Department'])
combined[1].Department.unique()
```

```
array([2, 1, 0, 3, 4])
```

```python
# Encoding Ward Type, Hospital_type_code, Ward_Facility_Code, Type_of_Admission, Severity_of_Illness
for dataset in combined:
    label = LabelEncoder()
    dataset['Hospital_type_code'] = label.fit_transform(dataset['Hospital_type_code'])
    dataset['Ward_Facility_Code'] = label.fit_transform(dataset['Ward_Facility_Code'])
    dataset['Ward_Type'] = label.fit_transform(dataset['Ward_Type'])
    dataset['Type_of_Admission'] = label.fit_transform(dataset['Type_of_Admission'])
    dataset['Severity_of_Illness'] = label.fit_transform(dataset['Severity_of_Illness'])
```

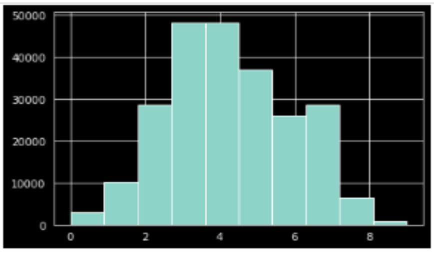
```python
combined[0]
```

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Available_Extra_Rooms_in_Hospital	Department	Ward_Type	Ward_Facility_Code	Type_of_Admission	Severit
0	1	8	2	3	3	3	2	5	0	
1	2	2	2	5	2	3	3	5	1	
2	3	10	4	1	2	1	3	4	1	
3	4	26	1	2	2	3	2	3	1	
4	5	26	1	2	2	3	3	3	1	
...	...	...	...	...	...	...	...	...	...	
237304	237305	23	0	6	3	2	2	5	1	
237305	237306	19	0	7	2	2	2	2	0	
237306	237307	8	2	3	5	2	1	5	0	
237307	237308	21	2	3	4	3	3	0	0	
237308	237309	5	0	1	3	2	1	4	1	

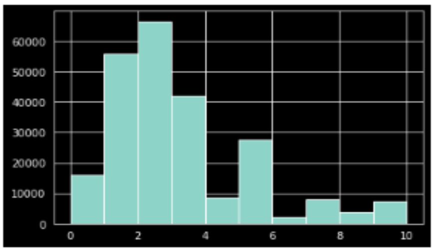
237309 rows × 14 columns

```python
combined[1]
```

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Available_Extra_Rooms_in_Hospital	Department	Ward_Type	Ward_Facility_Code	Type_of_Admission	Severit
0	318439	21	2	3	3	2	3	0	0	
1	318440	29	0	4	2	2	3	5	1	
2	318441	26	1	2	3	2	1	3	0	



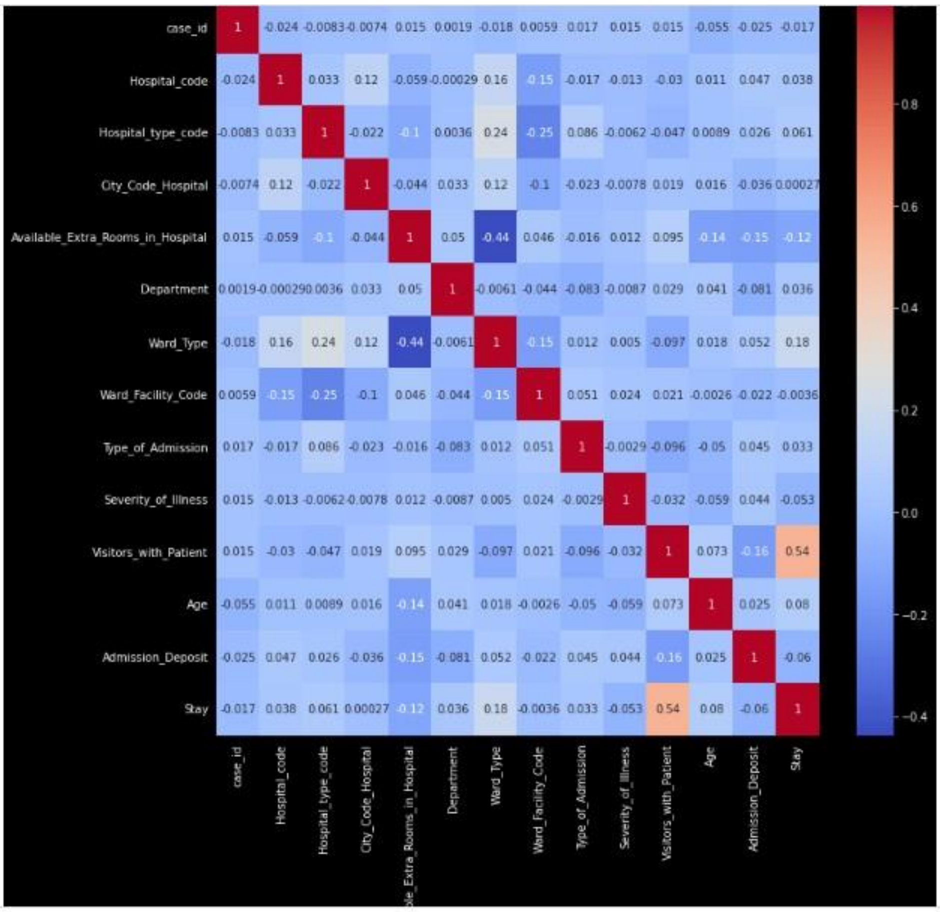

```
combined[0].Stay.hist()
```



shape of combined (train data, test data) dataset

```
for dataset in combined:
    print(dataset.shape)
```

```
(237309, 14)
(137057, 13)
```



```
combined[1]
```

	case_id	Hospital_code	Hospital_type_code	City_Code_Hospital	Available_Extra_Rooms_in_Hospital	Department	Ward_Type	Ward_Facility_Code	Type_of_Admission	Severi
0	318439	21	2	3	3	2	3	0	0	
1	318440	29	0	4	2	2	3	5	1	
2	318441	26	1	2	3	2	1	3	0	
3	318442	6	0	6	3	2	1	5	1	
4	318443	28	1	11	2	2	2	5	1	
...	...	...	...	...	...	...	...	...	...	
137052	455491	11	1	2	4	1	1	3	0	
137053	455492	25	4	1	2	3	2	4	0	
137054	455493	30	2	3	2	1	2	0	2	
137055	455494	5	0	1	2	1	2	4	1	
137056	455495	6	0	6	3	2	1	5	1	

137057 rows × 13 columns

Training the model

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import Perceptron
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
```

```
train = combined[0]
test = combined[1]
```

```
X_train = train.drop(['case_id', 'Stay'], axis=1)
Y_train = train["Stay"]
X_test  = test.drop("case_id", axis=1).copy()
```

```
X_train.shape
```

```
(237309, 12)
```

```
Y_train.shape
```

```
(237309,)
```

```
X_test.shape
```

```
(137057, 12)
```

```
X_test.columns
```

```
Index(['Hospital_code', 'Hospital_type_code', 'City_Code_Hospital',
       'Available_Extra_Rooms_in_Hospital', 'Department', 'Ward_Type',
       'Ward_Facility_Code', 'Type_of_Admission', 'Severity_of_Illness',
       'Visitors_with_Patient', 'Age', 'Admission_Deposit'],
      dtype='object')
```

```
Y_train
```

```
0         0.0
1         4.0
2         3.0
3         4.0
4         4.0
         ...
237304    5.0
237305    3.0
237306    2.0
237307    1.0
237308    NaN
Name: Stay, Length: 237309, dtype: float64
```

```
X_train.fillna(0,inplace=True)
Y_train.fillna(0,inplace=True)
X_test.fillna(0,inplace=True)
```

K-Nearest Neighbor Algorithm

```
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)
acc_knn = round(knn.score(X_train, Y_train) * 100, 2)
acc_knn
```

```
53.99
```

Descision Tree Algorithm

```
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
Y_pred = decision_tree.predict(X_test)
acc_decision_tree = round(decision_tree.score(X_train, Y_train) * 100, 2)
acc_decision_tree
```

```
99.76
```

Random Forest Algorithm

```
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, Y_train)
Y_pred = random_forest.predict(X_test)
random_forest.score(X_train, Y_train)
acc_random_forest = round(random_forest.score(X_train, Y_train) * 100, 2)
acc_random_forest
```

```
99.76
```

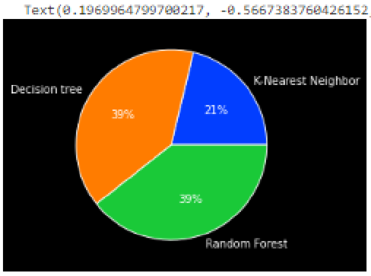
Prediction accuracy comparison

```
palette_color = sns.color_palette('bright')
data=[acc_knn, acc_decision_tree,acc_random_forest]
keys=['K-Nearest Neighbor','Decision tree','Random Forest']

#getting the algorithm with highest accuracy
max_accuracy=max(data)
index=[0,0,0]
j=0;
for i in data:
  if(i==max_accuracy):
    index[j]=1
    j=j+1
  else:
    index[j]=0.01
    j=j+1

plt.pie(data, labels=keys, colors=palette_color, autopct='%.0f%%')
```

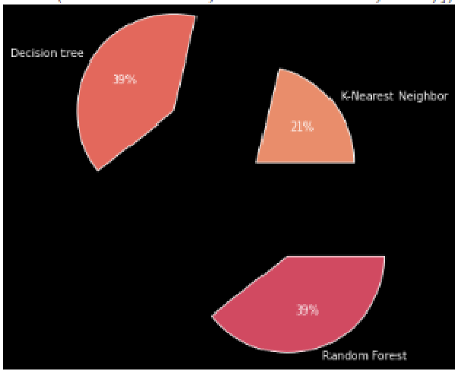
```
([,
  ,
  ],
 [Text(0.8628423642631272, 0.682277842548633, 'K-Nearest Neighbor'),
  Text(-0.9277499083745313, 0.590999244932723, 'Decision tree'),
  Text(0.36116021327837317, -1.0390203560781281, 'Random Forest')],
 [Text(0.4706412895980693, 0.3721515504810725, '21%'),
  Text(-0.5060454045679261, 0.322363224508758, '39%'),
  Text(0.1969964799700217, -0.5667383760426152, '39%')])
```




```
palette_color = sns.color_palette('flare')
plt.pie(data, labels=keys, colors=palette_color,explode=index, autopct='%.0f%%')
```

..

```
 ],
[Text(0.8706863857564283, 0.6884803683899842, 'K-Nearest Neighbor'),
 Text(-1.7711589159877414, 1.1282712857806532, 'Decision tree'),
 Text(0.689487679895076, -1.9835843161491535, 'Random Forest')],
[Text(0.47848531109137044, 0.37835407632242374, '21%'),
 Text(-1.349454412181136, 0.859635265356688, '39%'),
 Text(0.5253239465867245, -1.5113023361136406, '39%')])
```




```
output = pd.DataFrame({
        "case_id": test["case_id"],
        "Stay": Y_pred
})
```

```
output['Stay'] = output['Stay'].replace(stay_labels.values(), stay_labels.keys())
```

```
output.to_csv('LOS_Prediction.csv', index = False)
```

```
output
```

	case_id	Stay
0	318439	0-10
2	318441	21-30
3	318442	11-20
4	318443	31-40
...	...	...
137052	455491	0-10
137053	455492	0-10
137054	455493	21-30
137055	455494	21-30
137056	455495	51-60

137057 rows × 2 columns

```
data=np.array([[29,0,4,2,2,3,5,1,2,4,7,4018]])
p=random_forest.predict(data)
p
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not have valid feature names, but RandomForestClassifier was fitted wi
th feature names
  "X does not have valid feature names, but"
array([5.])
```

```
def prediction(p):
  if(p[0]==0):
    print("The predicted LOS of patient is : 0-10")
  elif(p[0]==1):
    print("The predicted LOS of patient is : 11-20")
  elif(p[0]==2):
    print("The predicted LOS of patient is : 21-30")
  elif(p[0]==3):
    print("The predicted LOS of patient is : 31-40")
  elif(p[0]==4):
    print("The predicted LOS of patient is : 41-50")
  elif(p[0]==5):
    print("The predicted LOS of patient is : 51-60")
  elif(p[0]==6):
    print("The predicted LOS of patient is : 61-70")
  elif(p[0]==7):
    print("The predicted LOS of patient is : 71-80")
  elif(p[0]==8):
```