

LOYOLA COLLEGE (AUTONOMOUS), CHENNAI – 34
DEPARTMENT OF DATA SCIENCE

INTERNSHIP REPORT



Name	AGNELO CHRISTY P
Dept. No	21-PDS-006
Guide	Dr.P.Manikandan
Company	Fantasy Solution
Internship Start Date	29.06.22
Internship End Date	16.07.22
Internship Period	15 Days
Type of Internship	Machine learning

ACKNOWLEDGEMENT

This internship report is the result of a collaborative effort by a number of people. I'd want to thank the Principal Rev. Fr. Dr. A Thomas, for allowing us to experience an internship experience even amid Covid-19's difficult times. I'd also like to thank Dr. T. Rajaretnam, the Head of the Department of Data Science, for frequently stimulating and emphasizing the relevance of an internship experience to the students. I'd like to take this opportunity to thank Dr.P.Manikandan, my internship guide, deserves a lot of credit for preparing the students for a successful internship experience.

The internship I did with Fantasy solution was a fantastic opportunity for me to learn and grow professionally. I'd want to offer my heartfelt gratitude to the HR-Manager in particular. I'd also like to thank my trainer, Mr. Hadley Jones, for providing me with this one-month internship opportunity as a Data Analyst intern, who, despite being extremely busy with his responsibilities, took time out to guide me, keep me on track throughout the internship program, and make me feel at ease in the virtual work environment.

I consider this internship to be a significant step forward in my professional development. I will make every effort to put my newly acquired skills and knowledge to the greatest possible use. In order to achieve my job goals, I will continue to focus on my progress.

CERTIFICATE



FANTASY SOLUTION
Find Us Passion...
SOFTWARE DEVELOPMENT AND TRAINING

Certificate of Accomplishment

This is certify that Mr./Miss. AGNELO CHRISTY P 21-PDC-006

Student of LOYOLA COLLEGE
CHENNAI

underwent Inplant Training / Internship / Course on MACHINE LEARNING
concern from 29/06/2022 to 16/07/2022 issued on 16/07/2022.



Technical Incharge



Manager



CEO



#16, Samnath Plaza, 3rd Floor, Madurai Main Road, Melapudai, Trichy-1.
☎ 0431 4971630 ✉ fantazysolution@gmail.com 🌐 www.fantazysolution.in

TABLE OF CONTENTS

S.NO	CONTENTS
1.	Organization Profile
2.	SUMMARY OF THE INTERNSHIP
3.	INTERNSHIP ACTIVITIES
4.	ASSESSMENT OF THE INTERNSHIP & ABOUT PROJECT
5.	CONCLUSION
6.	Appendix
7.	Reference

COMPANY/INSTITUTION/ORGANISATION PROFILE

Fantasy Solution

#16, Samnath Plazza, Third Floor, Madurai Main Rd, Melapudur, Tiruchirappalli, Tamil Nadu 620001

Fantasy is launched by group of young talented experienced Engineers and Technologists. Before launching this Company the same company was recognized as '**Fantasy** solution', which was established in January 2008. The Engineering concepts are implemented ingeniously in Software Development, Web Engineering, Networking and Consultancy.

Presently **Fantasy** is in the phase of making tie-up with reputed Indian Company for the performance of Software and Testing Outsource.

Fantasy has a strong corporate philosophy when it comes to work. Beliefs that can be summed up in just two small words- 'have fun!' Enjoy whatever you do and do it with passion. This attitude is very well reflected in our office. Bright, Colorful workplace, music and cheerful faces abound. Though the atmosphere is relaxed, we must say, we do get our work done and done well at that! In Web Solutions, whether it is design, hosting or innovative e-com solutions, we make them work for you.

We believe in partnering you fully to maximize your presence on Web. To strategize and create a lasting impact for all your online brands. Till today we have developed more than 120 websites. In Software Solutions we have developed standardized packages for Inventory Management Control and Accounting systems, Specialized Inventory Control system, Hospital Management System, Hotel Management system, Textile Inventory Management Systems, Critical Job Management Systems, Personal Assistant Software and so on. Major Software Jobs executed in TechSpire are in the higher end technologies like .NET and JAVA including separate Testing Section.

In Networking & Internetworking Solutions Our Networking Development Division is one of the best divisions in Chennai. We have wide experience in providing LAN/WAN Internetworking Solutions, ISDN Solutions, DSL Solutions, RAS (Remote Access Services) & VPN (Virtual Private Network) Solutions.

Channel Business

Fantasy increases its brand presence in small, home and networked offices by utilizing the fastest-growing channels to offer more products to ore customers.

The Role of the Industry:

The IT industry can serve as a medium of e-governance, as it assures easy accessibility to information. The use of information technology in the service sector improves operational efficiency and adds to transparency. It also serves as a medium of skill formation.

SUMMARY

- My internship was done in computer management. In the above company I worked
As trainee in Python.
- In the first week I learnt the system Software. During the training session time, I collected information about the type of packages Server and other programming's.
- I was trained by well experienced staff members. After my training got over, I Was given the opportunity to learn the objectives as well as how to do the implementation of the project and the various processes involved in the development of Application.
- During my internship, I was thoroughly taught and explained many useful Machine Learning ideas. Machine learning used in many fields of medical, IT sectors etc. We had training about various Machine learning models, python. Internships are used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- During my training, we was given a short introduction to various tools that is being used in data science field. We were trained to process the data, visualize the data and build the model. To import data we used pandas, to visualize we used matplotlib and mathematical operation we used numpy.
- After the completion of the training , we were asked to do mini project under the Machine learning ,atlast 2 days of internship period. During the time of our mini project development,we get to know handle missing values , feature scaling and building various machine learning models.
- My trainer helped me and guided me in the various topic of machine learning. She helped me to understand to of various areas.I always gave it my all to develop the finest possible product of model. I successfully dealt with the issue and were able to quickly discover which techniques .

INTERNSHIP ACTIVITIES

In the first week of my internship they taught me about the Introduction of Python then about then about the basics of the python it was very east to learn. Then slowly they moved through the machine learning concepts. Machine learning concepts where easy to learn since it was already well taught by our college staff. Then it was about introduction and importing the packages. The most important packages used in machine learning for analysis process and prediction and so on. The concept of big data was also an insightful that they conducted.

Understanding the different types of class concept and collection concept. How to use the class in a python program. The first week also consists of explaining the concepts of tuple, lists, dictionaries, set, exception handling and so on.

I learned a lot of new things that have helped me develop professionally. I was also able to identify my assets, liabilities, opportunities, and risks thanks to it. I worked well with the instructor and learned a lot about the industry. Major Duties of Internship, Finally, our trainer graded the completed tasks. They requested that I complete a little project in machine learning as part of my internship after my internship had ended. They gave Introduction about AI and ML. Then gave Python Introduction, And then teach about Conditional Statements, Looping and Control Statements how to work with and to use.

They taught about Missing data, Feature Scaling for EDA process. They start with Linear and multiple Regression, Logistic Regression and then later they taught about Support Vector Machine, Random Forest. They taught about Apriori Algorithm, K-Means Clustering how it works and etc. Later that above algorithms they taught other algorithms Naïve Bayes Classification, Ensemble methods-Bagging and boosting, Density Based Clustering, Principal Component Analysis. They gave introduction about what is Reinforcement Learning and

how it works in real time. And then they assign a project work.

AN ASSESSMENT OF THE INTERNSHIP

In Fantasy, this internship has been an excellent and rewarding experience. I can conclude that there have been a lot I've learnt from my work at Fantasy. Needless to say, the technical aspects of the work I've done are not flawless and could be improved provided enough time. As someone with no prior experience with .net development whatsoever I believe my time spent in research and discovering it was well worth it and contributed to finding an acceptable solution to build a fully functional web service. Two main things that I've learned the importance of our time-management skills and self-motivation. Punctuality is one of the main factor that made me uncomfortable in the in the beginning but later realized that it was a necessary quality factor in life. It was through this that helped the faculties and the students to benefit out of me. And also students were regular at their work.

It was the main factor that helped me to be prepared for the next day's event, so that I can handle the next day's class with comfort. Preparations lead me to referencing various text books and surf the internet that was interesting and also made my understanding with ease and impart students with the knowledge what I gained out of it. And also I met people of different age and I had to adapt to their level of understanding and this helped me to improve my communication. I also interacted with the faculties as well thereby gaining the knowledge of skills that they possessed.

The ideas I learned in the first and second semesters of my M.Sc.DataScience program at Loyola College were extremely useful in my internship, since I applied many of the concepts I learned, such as exploratory data analysis, data visualization, and so on.

It also made me learn the team work, program tackling and situations handling. It also helped me to efficiently work for longer time until I complete the task that was assigned to me.

ABOUT PROJECT

The main objective of this project is to overcome the limitations and to design a robust system which works efficiently and will be able to **predict** whether the heart has any problem in order to avoid heart failure. In this modern era people are very busy and working hard in order to satisfy their materialistic needs and not able to spend time for themselves which leads to physical stress and mental disorder. There are also reports that heart suffers because of global pandemic corona virus. Inflammation of the heart muscle can be caused by corona virus. Thus heart disease is very common now a day's particularly in urban areas because of excess mental stress due to corona virus. As a result Heart disease has become one of the most important factors for death of men and women in the so called material world. It has emerged as the top killer that has affected both urban and rural population. CAD (Coronary artery disease) is one of the most common types of heart disease. In the medical field predicting the heart disease has become a very complicated and challenging task, requires patient previous health records and in some cases they even need Genetic information as well. So, in this contemporary life style there is an urgent need of a system which will predict accurately the possibility of getting heart disease. Predicting a Heart Disease in early stage will save many people's Life.

There were many heart disease prediction systems available at present. The main objective of this project is to overcome the limitations and to design a robust system which works efficiently and will be able to predict the possibility of heart failure accurately. This paper uses the data set from the UCI repository and having 13 important attributes. This work is implemented using many algorithms such as logistic regression, random forest, KNN, Catboost classifier, decision tree, gradient boosting classifier, support vector classifier and Adaboost classifier.

Dataset:

The dataset I used consists of the following feature

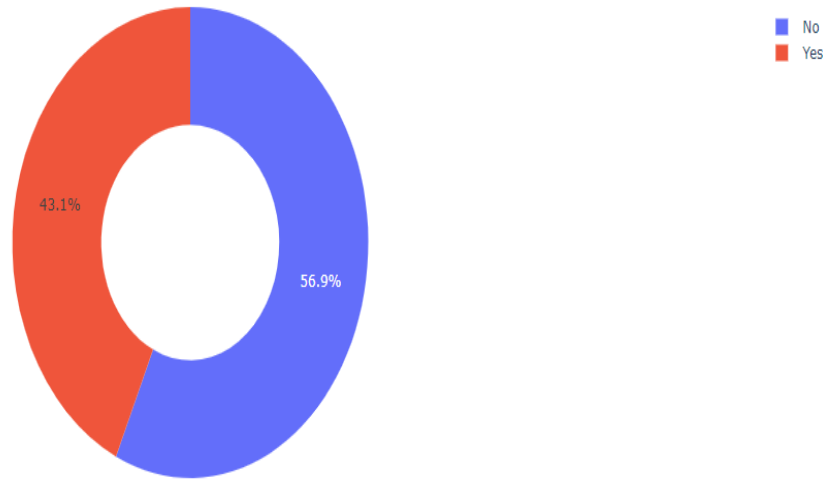
- age
- anaemia
- creatinine_phosphokinase
- diabetes
- ejection_fraction
- high_blood_pressure
- platelets
- serum_creatinine
- serum_sodium
- sex
- smoking time
- DEATH_EVENT

This features help us in identifying the causes of heart failure and to prevent it.

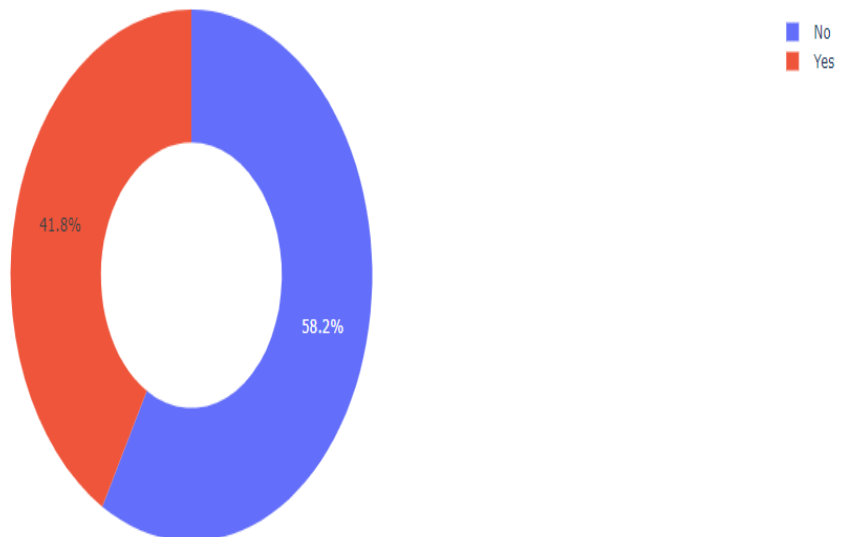
	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8	1

RESULTS:

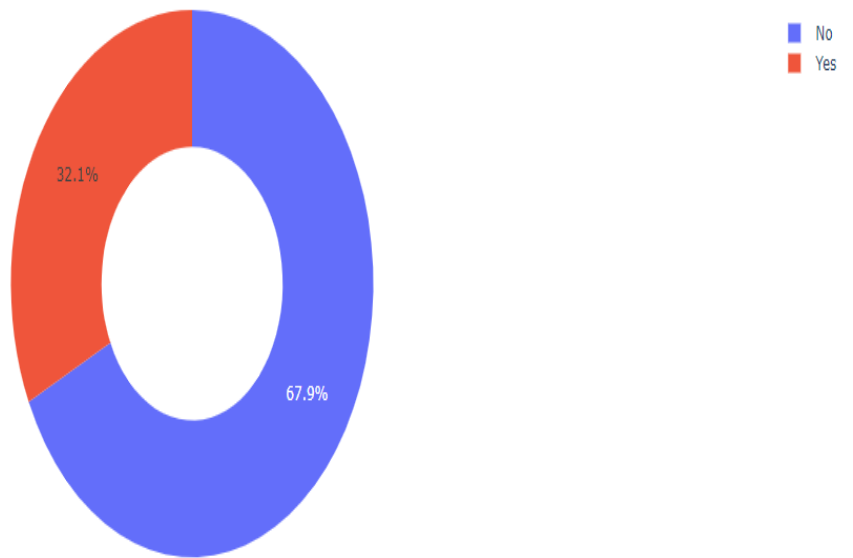
Anaemia Distribution



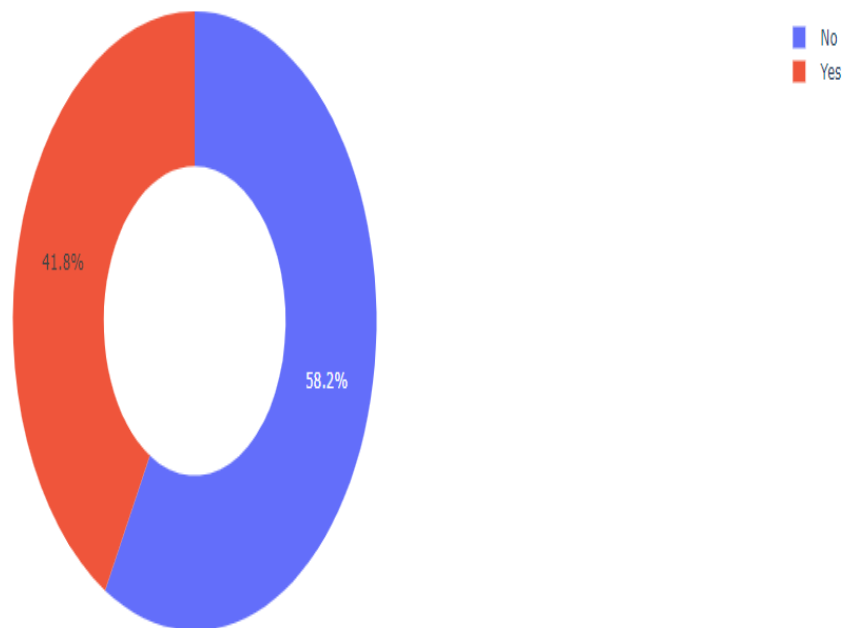
Diabetes Distribution



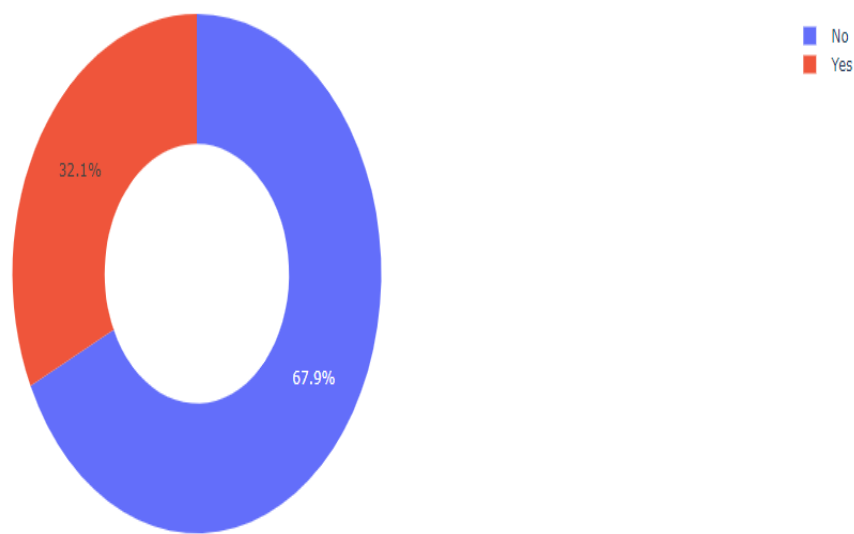
Smoking Distribution



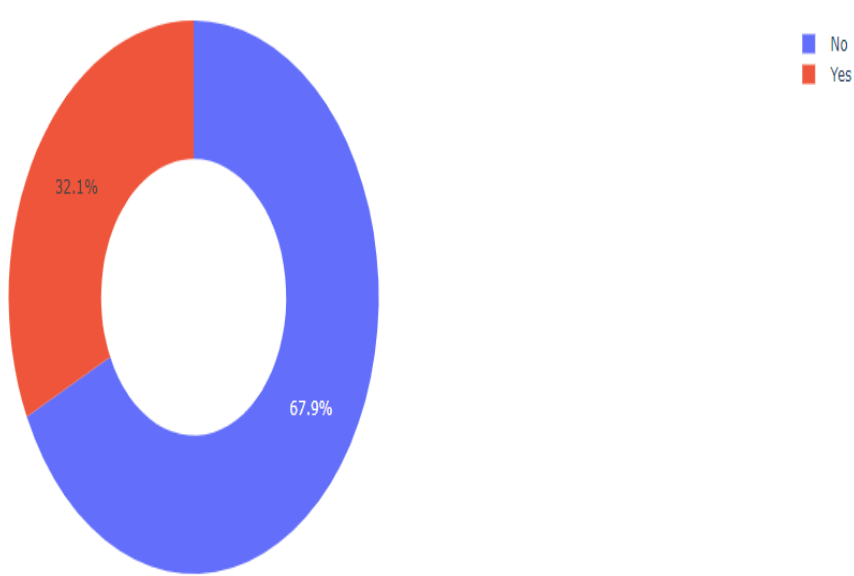
Diabetes Distribution



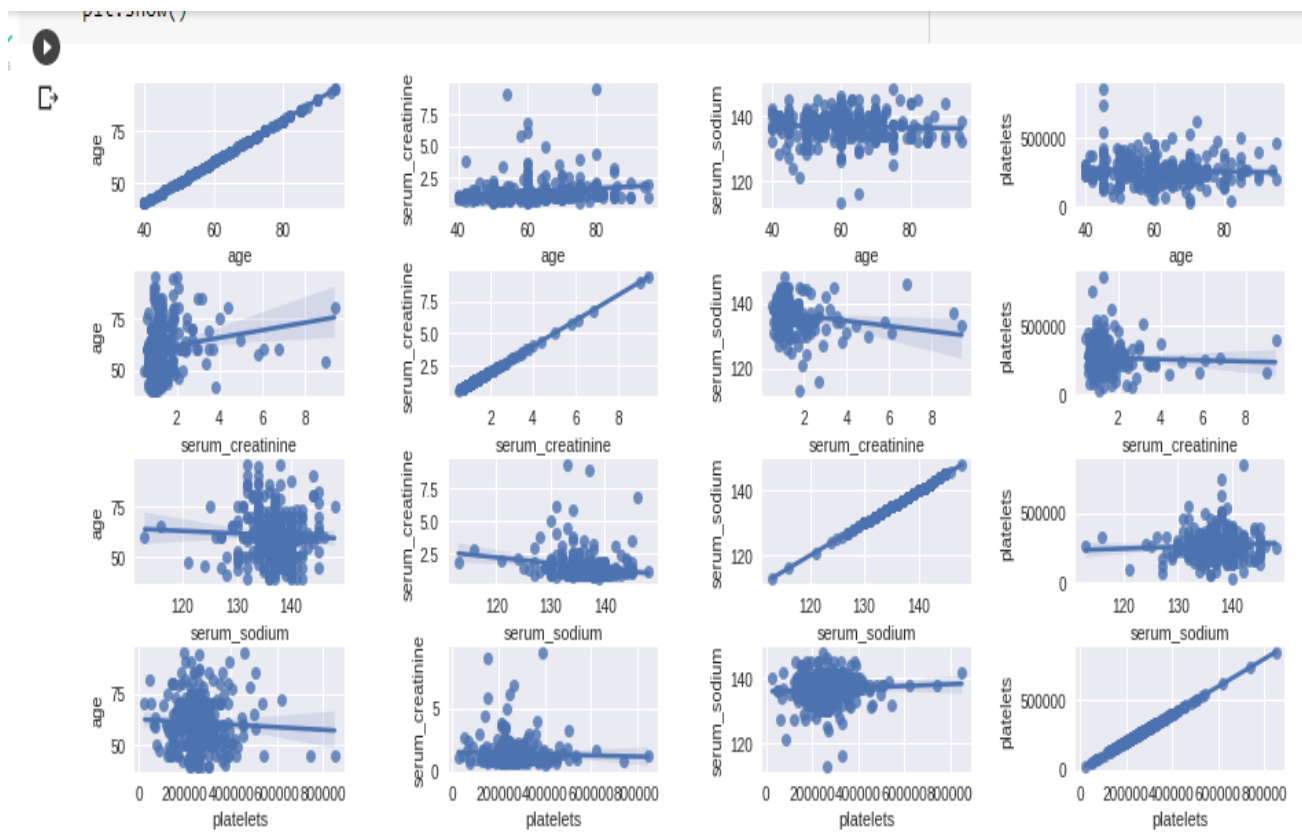
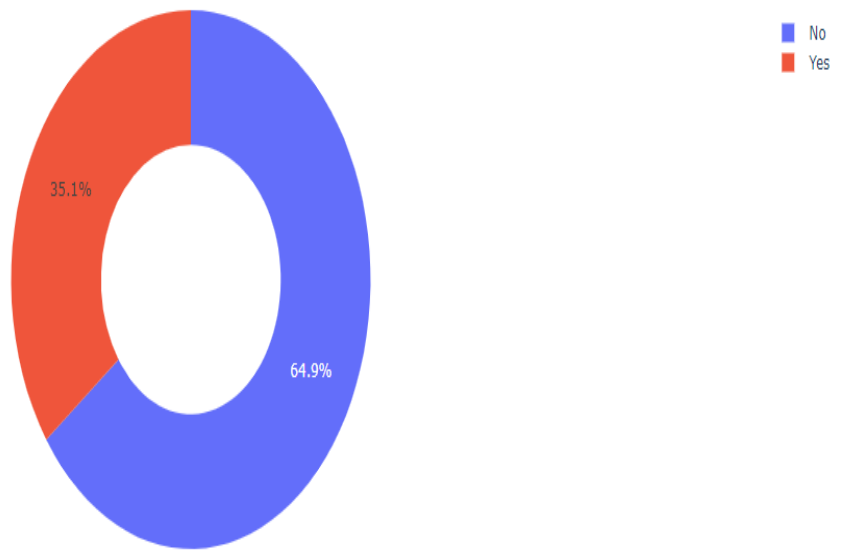
Smoking Distribution



DEATH EVENT Distribution



high blood pressure Distribution





Ds



```
df.corr()["DEATH_EVENT"].sort_values(ascending = False)
```

```
DEATH_EVENT          1.000000
serum_creatinine      0.294278
age                   0.253729
high_blood_pressure   0.079351
anaemia               0.066270
creatinine_phosphokinase 0.062728
diabetes              -0.001943
sex                   -0.004316
smoking               -0.012623
platelets             -0.049139
serum_sodium          -0.195204
ejection_fraction     -0.268603
time                  -0.526964
Name: DEATH_EVENT, dtype: float64
```

Build Random Forest Classifier and evaluate the model



```
clf_rf = RandomForestClassifier(random_state=777)
clf_rf = clf_rf.fit(X_train,y_train)
y_pred_rf = clf_rf.predict(X_test)
acc = accuracy_score(y_test, y_pred_rf)
print('Testing-set Accuracy score is:', acc)
print('Training-set Accuracy score is:',accuracy_score(y_train,clf_rf.predict(X_train)))
cm = confusion_matrix(y_test, y_pred_rf)
```



```
Testing-set Accuracy score is: 0.8333333333333334
Training-set Accuracy score is: 1.0
```

Build Ada Boost Classifier and evaluate the model



```
abc = AdaBoostClassifier(n_estimators = 50,
                          learning_rate = 1,
                          random_state = 777)

abc.fit(X_train,y_train)
y_pred_abc = abc.predict(X_test)
acc = accuracy_score(y_test, y_pred_abc)
print('AdaBoost Classifier Model Accuracy is:',acc)
cm = confusion_matrix(y_test, y_pred_abc)
```

```
AdaBoost Classifier Model Accuracy is: 0.8
```


Build Gradient Boosting Classifier and evaluate the model

```
✓ [42] gb = GradientBoostingClassifier()  
gb.fit(X_train, y_train)  
gb_pred = gb.predict(X_test)  
acc = accuracy_score(y_test, gb_pred)  
print("Gradient Boosting Classifier Model Accuracy score is:", acc)  
cm = confusion_matrix(y_test, gb_pred)
```

Gradient Boosting Classifier Model Accuracy score is: 0.8833333333333333

Build KNN Model and evaluate the model

```
✓ [43] knn = KNeighborsClassifier(n_neighbors = 10)  
knn.fit(X_train, y_train)  
knn_pred = knn.predict(X_test)  
acc = knn.score(X_test, y_test)  
print("KNN Model Accuracy is:", acc)  
cm = confusion_matrix(y_test, knn_pred)
```

KNN Model Accuracy is: 0.8333333333333334

Build SVC Model and evaluate the model

```
[44] svc = SVC(random_state = 777)
      svc.fit(X_train, y_train)
      svc_pred = svc.predict(X_test)
      acc = svc.score(X_test, y_test)
      print("SVC Accuracy score is:", acc)
      cm = confusion_matrix(y_test, svc_pred)
```

SVC Accuracy score is: 0.8666666666666667

Build Logistic Regression Model and evaluate the model

```
[45] lr = LogisticRegression()
      lr.fit(X_train, y_train)
      lr_pred = lr.predict(X_test)
      acc = lr.score(X_test, y_test)
      print("LogisticRegression accuracy score is:", acc)
      report = classification_report(y_test, lr_pred)
      print(report)
```

LogisticRegression accuracy score is: 0.8666666666666667

	precision	recall	f1-score	support
0	0.91	0.91	0.91	45
1	0.73	0.73	0.73	15
accuracy			0.87	60
macro avg	0.82	0.82	0.82	60
weighted avg	0.87	0.87	0.87	60

Build Decision Tree Classifier and evaluate the model

```
[46] dt = DecisionTreeClassifier()  
      dt.fit(X_train, y_train)  
      dt_pred = dt.predict(X_test)  
      acc = accuracy_score(y_test, dt_pred)  
      print("Decision Tree accuracy score is :",acc)  
      cm = confusion_matrix(y_test, dt_pred)
```

Decision Tree accuracy score is : 0.7666666666666667

```
cbc = CatBoostClassifier( iterations=10,learning_rate=0.0001 )  
cbc = cbc.fit(X_train,y_train)  
cbc_pred = cbc.predict(X_test)  
acc = accuracy_score(y_test,cbc_pred)  
print("Cat Boost Classifier accuracy score is:",acc)  
cm = confusion_matrix(y_test,cbc_pred)
```

0:	learn: 0.6931202	total: 48.9ms	remaining: 440ms
1:	learn: 0.6930862	total: 51.9ms	remaining: 208ms
2:	learn: 0.6930573	total: 54.3ms	remaining: 127ms
3:	learn: 0.6930274	total: 56.5ms	remaining: 84.8ms
4:	learn: 0.6929965	total: 58.8ms	remaining: 58.8ms
5:	learn: 0.6929697	total: 61ms	remaining: 40.7ms
6:	learn: 0.6929348	total: 63.2ms	remaining: 27.1ms
7:	learn: 0.6928994	total: 64.5ms	remaining: 16.1ms
8:	learn: 0.6928645	total: 65.8ms	remaining: 7.31ms
9:	learn: 0.6928315	total: 67ms	remaining: 0us

Cat Boost Classifier accuracy score is: 0.8666666666666667

Build Voting Classifier and evaluate the model

```
[48] clf1 = GradientBoostingClassifier()
      clf2 = LogisticRegression()
      clf3 = AdaBoostClassifier()
      eclf1 = VotingClassifier(estimators=[('gbc', clf1), ('lr', clf2), ('abc', clf3)], voting='soft')
      eclf1.fit(X_train, y_train)
      predictions = eclf1.predict(X_test)
      print("Voting Classifier Accuracy Score is: ")
      print(accuracy_score(y_test, predictions))
      cm = confusion_matrix(y_test, predictions)
```

```
Voting Classifier Accuracy Score is:
0.8833333333333333
```

Models	Accuracy
Logistic regression	0.86
Random forest	0.83
KNN	0.83
Catboostclassifier	0.86
Decision tree	0.73
Gradient boosting classifier	0.88
Support vector classifier	0.86
Adaboost classifier	0.80

CONCLUSION

In this project I have fitted eight models. This work is implemented using many algorithms such as logistic regression, random forest, KNN, Catboostclassifier, decision tree, gradient boosting classifier, support vector classifier and Adaboost classifier using standard scalar. The Voting classifier Accuracy score of 0.883%. Data preprocessing is an important phase while making prediction from a dataset. Data preprocessing includes Standard Scalar, Visualization. These techniques are used to change categorical data into numerical data. Random forest classifier has been used to improve the accuracy of classifiers.

Appendix

```
import pandas as pd
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
import plotly.graph_objects as go
```

```
%matplotlib inline
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.ensemble import AdaBoostClassifier
```

```
from sklearn.ensemble import GradientBoostingClassifier
```

```
from sklearn.ensemble import VotingClassifier
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.svm import SVC
```

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import accuracy_score
```

```
from sklearn.metrics import recall_score, confusion_matrix,  
precision_score, f1_score, classification_report
```

```
from sklearn.preprocessing import StandardScaler
```

```
!pip install catboost
```

```
from catboost import CatBoostClassifier
```

```
from google.colab import drive
```

```
drive.mount('/content/drive')
```

```
df =  
pd.read_csv('/content/drive/MyDrive/heart_failure_clinical_records_d  
ataset.csv')
```

```
df.shape
```

```
df.head()
```

```
df.info()
```

```
plt.style.use('seaborn')
```

```
plt.figure(figsize=(10,5))
```

```
sns.heatmap(df.isnull(), yticklabels = False, cmap = 'plasma')
```

```
plt.title('Null Values in Data Frame')
```

```
missing_value_count = (df.isnull().sum())
```

```
print(missing_value_count[missing_value_count > 0])
```

```
total_cells = np.product(df.shape)
```

```
total_missing_value = missing_value_count.sum()
```

```
print('Percentage of missing value in Data Frame is:',  
total_missing_value / total_cells*100)  
print('Total number of our cells is:', total_cells)  
print('Total number of our missing value is:', total_missing_value)
```

```
data_eda = df.copy()
```

```
data_eda["sex"] = df["sex"].map({1: "Male", 0: "Female"})  
data_eda["diabetes"] = df["diabetes"].map({1: "Yes", 0: "No"})  
data_eda["anaemia"] = df["anaemia"].map({1: "Yes", 0: "No"})  
data_eda["high_blood_pressure"] =  
df["high_blood_pressure"].map({1: "Yes", 0: "No"})  
data_eda["smoking"] = df["smoking"].map({1: "Yes", 0: "No"})  
data_eda["DEATH_EVENT"] = df["DEATH_EVENT"].map({1:  
"Yes", 0: "No"})
```

```
# Exploratory Data Analysis(EDA)
```

```
df.describe()
```

```
df.corr()
```

```
def pie_graph(df,title,values):  
    labels = df[values].value_counts().index  
    values = df[values].value_counts()
```



```

ax = sns.distplot(df['creatinine_phosphokinase'], rug=True,
rug_kws={"color": "g"},
            kde_kws={"color": "k", "lw": 3, "label": "KDE"},
            hist_kws={"histtype": "step", "linewidth": 3,
                        "alpha": 1, "color": "g"})

```

```

pie_graph(data_eda, 'Diabetes Distribution','diabetes')

```

```

pie_graph(data_eda, 'Smoking Distribution','smoking')

```

```

ax = sns.distplot(df['platelets'], rug=True, rug_kws={"color": "g"},
            kde_kws={"color": "k", "lw": 3, "label": "KDE"},
            hist_kws={"histtype": "step", "linewidth": 3,
                        "alpha": 1, "color": "g"})

```

```

pie_graph(data_eda, 'DEATH EVENT
Distribution','DEATH_EVENT')

```

```

ax = sns.distplot(df['serum_creatinine'], rug=True, rug_kws={"color":
"g"},
            kde_kws={"color": "k", "lw": 3, "label": "KDE"},
            hist_kws={"histtype": "step", "linewidth": 3,
                        "alpha": 1, "color": "g"})

```

```

pie_graph(data_eda, 'high blood pressure

```

```
Distribution','high_blood_pressure')
```

```
ax = sns.distplot(df['serum_sodium'], rug=True, rug_kws={"color":  
"g"},  
                  kde_kws={"color": "k", "lw": 3, "label": "KDE"},  
                  hist_kws={"histtype": "step", "linewidth": 3,  
                             "alpha": 1, "color": "g"})
```

```
ax = sns.distplot(df['time'], rug=True, rug_kws={"color": "g"},  
                  kde_kws={"color": "k", "lw": 3, "label": "KDE"},  
                  hist_kws={"histtype": "step", "linewidth": 3,  
                             "alpha": 1, "color": "g"})
```

```
df.corr()["DEATH_EVENT"].sort_values(ascending = False)
```

```
# Pre Modeling Steps
```

```
X = df.drop(columns = ["DEATH_EVENT"])  
y = df["DEATH_EVENT"]
```

```
X_train, X_test, y_train, y_test = train_test_split(X,  
                                                    y,  
                                                    test_size = .2,  
                                                    random_state = 777)  
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

```
scaler = StandardScaler()
```

```
scaler.fit(X_train)
```

```
X_train = scaler.transform(X_train)
```

```
X_test = scaler.transform(X_test)
```

```
# Apply Machine Learning Models
```

```
clf_rf = RandomForestClassifier(random_state=777)
```

```
clf_rf = clf_rf.fit(X_train,y_train)
```

```
y_pred_rf = clf_rf.predict(X_test)
```

```
acc = accuracy_score(y_test, y_pred_rf)
```

```
print('Testing-set Accuracy score is:', acc)
```

```
print('Training-set Accuracy score is:',accuracy_score(y_train,clf_rf.predict(X_train)))
```

```
cm = confusion_matrix(y_test, y_pred_rf)
```

```
abc = AdaBoostClassifier(n_estimators = 50,
```

```
learning_rate = 1,
```

```
random_state = 777)
```

```
abc.fit(X_train,y_train)
```

```
y_pred_abc = abc.predict(X_test)
```

```
acc = accuracy_score(y_test, y_pred_abc)
```

```
print('AdaBoost Classifier Model Accuracy is:',acc)
```

```
cm = confusion_matrix(y_test, y_pred_abc)
```

```
gb = GradientBoostingClassifier()
gb.fit(X_train, y_train)
gb_pred = gb.predict(X_test)
acc = accuracy_score(y_test, gb_pred)
print("Gradient Boosting Classifier Model Accuracy score is:", acc)
cm = confusion_matrix(y_test, gb_pred)
```

```
knn = KNeighborsClassifier(n_neighbors = 10)
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)
acc = knn.score(X_test, y_test)
print("KNN Model Accuracy is:", acc)
cm = confusion_matrix(y_test, knn_pred)
```

```
svc = SVC(random_state = 777)
svc.fit(X_train, y_train)
svc_pred = svc.predict(X_test)
acc = svc.score(X_test, y_test)
print("SVC Accuracy score is:", acc)
cm = confusion_matrix(y_test, svc_pred)
```

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
lr_pred = lr.predict(X_test)
acc = lr.score(X_test, y_test)
print("LogisticRegression accuracy score is:",acc)
report = classification_report(y_test, lr_pred)
print(report)
```

```
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
dt_pred = dt.predict(X_test)
acc = accuracy_score(y_test, dt_pred)
print("Decision Tree accuracy score is :",acc)
cm = confusion_matrix(y_test, dt_pred)
```

```
cbc = CatBoostClassifier( iterations=10,learning_rate=0.0001 )
cbc = cbc.fit(X_train,y_train)
cbc_pred = cbc.predict(X_test)
acc = accuracy_score(y_test,cbc_pred)
print("Cat Boost Classifier accuracy score is:",acc)
cm = confusion_matrix(y_test,cbc_pred)
```

```
clf1 = GradientBoostingClassifier()
clf2 = LogisticRegression()
clf3 = AdaBoostClassifier()
ecf1 = VotingClassifier(estimators=[('gbc', clf1), ('lr', clf2), ('abc',
clf3)], voting='soft')
ecf1.fit(X_train, y_train)
predictions = ecf1.predict(X_test)
print("Voting Classifier Accuracy Score is: ")
print(accuracy_score(y_test, predictions))
cm = confusion_matrix(y_test, predictions)
```

Reference

<http://www.fantasysolution.in/>

<https://www.linkedin.com/company/fantasy-solution/>