**A Review of Liver Patient Analysis Methods**

**Using Machine Learning**

1.INTRODUCTION

Liver diseases averts the normal function of the liver. This disease is caused by an assortment of elements that harm the liver.

**1.1Overview**

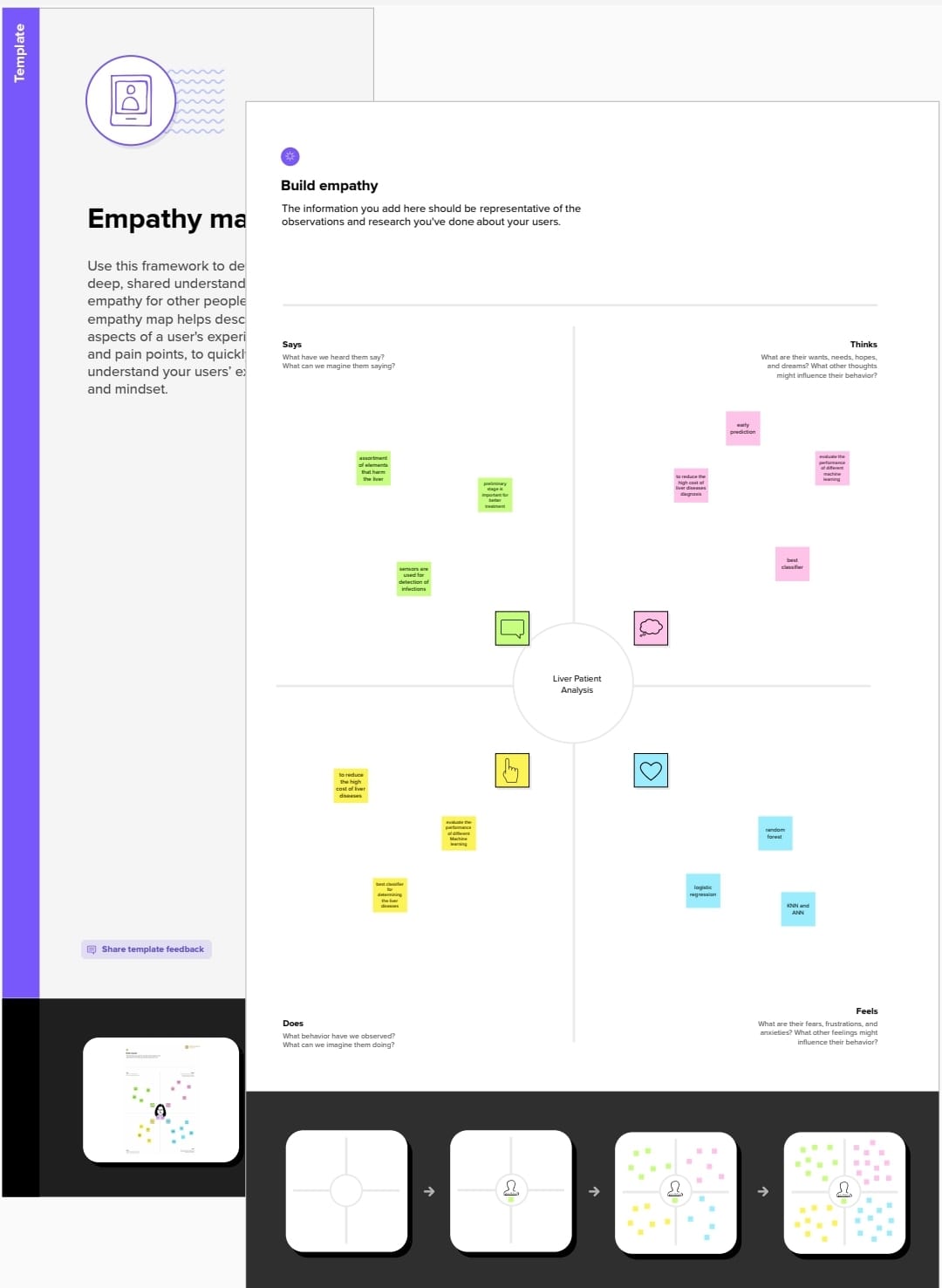
Diagnosis of liver infection at the preliminary stage is important for better treatment. In today’s scenario devices like sensors are used for detection of infections. Accurate classification techniques are required for automatic identification of disease samples. This disease diagnosis is very costly and complicated. Therefore, the goal of this work is to evaluate the performance of different Machine Learning algorithms in order to reduce the high cost of liver disease diagnosis. Early prediction of liver disease using classification algorithms is an efficacious task that can help the doctors to diagnose the disease within a short duration

**1.2 Purpose**

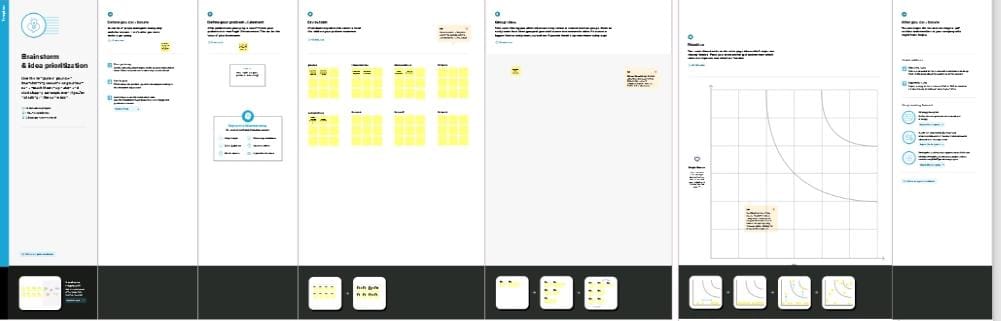
In this project we will analyse the parameters of various classification algorithms and compare their predictive accuracies so as to find out the best classifier for determining the liver disease. This project compares various classification algorithms such as Random Forest, Logistic Regression, KNN and ANN Algorithm with an aim to identify the best technique. Based on this study, Random Forest with the highest accuracy outperformed the other algorithms and can be further utilised in the prediction of liver disease and can be recommended to the user.

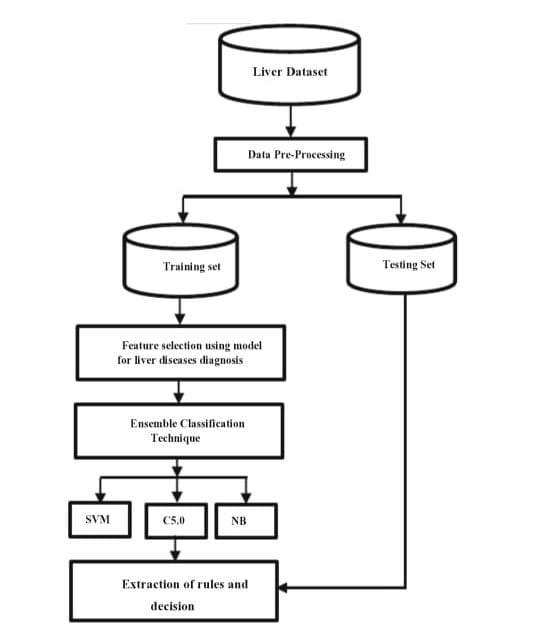
**2.Problem Definition & Design Thinking**

**2.1Empathy Map**



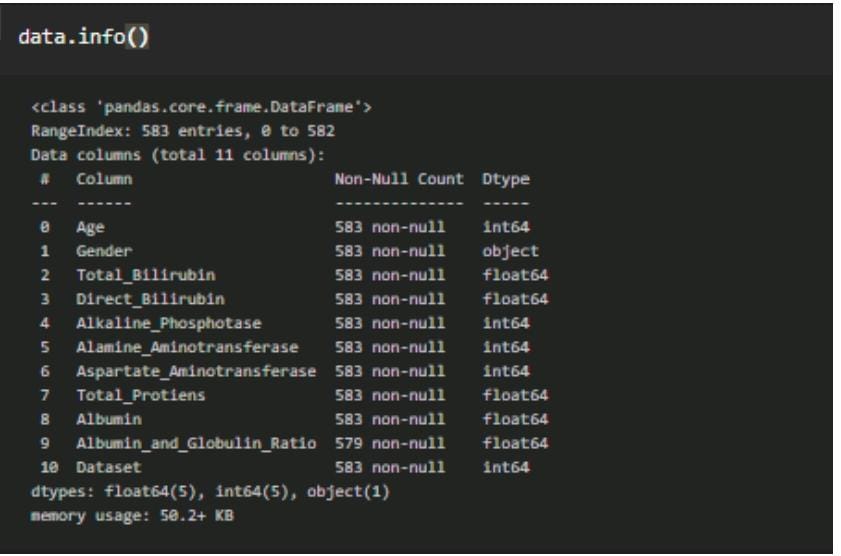
**2.2 Ideation & Brainstorming Map**

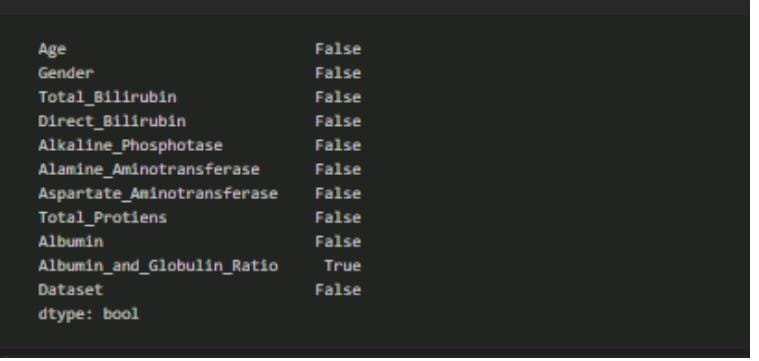
****

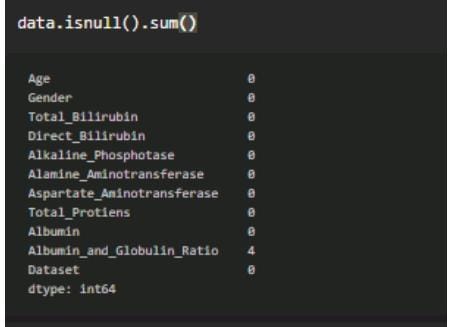
****

**3. RESULT**

****

****

****

****

****

****

**4.Advantages and Disadvantages**

Diagnostic criterion standard Confirmed diagnostic value Highly invasive test Etiologic suggestion Differential diagnosis. The potential complications include death Grade and stage evaluation Therapeutic decision Significant sampling error (eligibility) High costTreatment evaluation Inter-observer variation (effectiveness)Follow-up comparison of treated and untreated patients.

**5.Application**

The gold standard for the diagnosis of liver fibrosis and nonalcoholic fatty liver disease (NAFLD) is liver biopsy. Various noninvasive modalities, e.g., ultrasonography, elastography and clinical predictive scores, have been used as alternatives to liver biopsy, with limited performance. Recently, artificial intelligence (AI) models have been developed and integrated into noninvasive diagnostic tools to improve their performance.

Methods

We systematically searched for studies on AI-assisted diagnosis of liver fibrosis and NAFLD on MEDLINE, Scopus, Web of Science and Google Scholar. The pooled sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV) and diagnostic odds ratio (DOR) with their 95% confidence intervals (95% CIs) were calculated using a random effects model. A summary receiver operating characteristic curve and the area under the curve was generated to determine the diagnostic accuracy of the AI-assisted system. Subgroup analyses by diagnostic modalities, population and AI classifiers were performed.

**6.CONCLUSION**

Conclusion and Feature WorkIn this paper, we proposed and built a machine learning based on a hybrid classiﬁer tobe used as a classiﬁcation model for liver diseases diagnosis to improve performanceand experts to identify the chances of disease and conscious prescription of furthertreatment healthcare and examinations.In future work, the use of fast datasets technique like Apache Hadoop or Spark canbe incorporated with this technique. In addition to this, we can use distributed reﬁnedalgorithms like Forest Tree implemented in Apache Hadoop to increase scalability and eﬃciency.

**7.FUTURE SCOPE**

Liver diseases have produced a big data such as metabolomics analyses, electronic health records, and report including patient medical information, and disorders. However, these data must be analyzed and integrated if they are to produce models about physiological mechanisms of pathogenesis. We use machine learning based on classifier for big datasets in the fields of liver to Predict and therapeutic discovery. A dataset was developed with twenty three attributes that include the records of 7000 patients in which 5295 patients were male and rests were female. Support Vector Machine (SVM), Boosted C5.0, and Naive Bayes (NB), data mining techniques are used with the proposed model for the prediction of liver diseases. The performance of these classifier techniques are evaluated with accuracy, sensitivity, specificity.

**8.APPENDIX**

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from matplotlib import rcParams

from scipy import stats

from sklearn.preprocessing import LabelEncoder

lc = LabelEncoder()

data['gender']=lc.fit\_transform(data['gender'])

data describe()

sns.distplot(data['age'])

plt.title('Age Distribution Graph')

plt.show()

sns.countplot(data['outcome'],hue=data['gender'])

plt.figure(figsize=(10,7))

sns.heatmap(df.corr(),annot=True)

from sklearn.preprocessing import scale

X\_scaled=pd.DataFrame(scale(X),columns=X.columns)

X\_scaled.head()

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

y=data.outcome

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

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X\_scaled,y,test\_size=0.2,random\_state=42)

from imblearn.over\_sampling import SMOTE

smote = SMOTE()

y\_train.value\_counts()

X\_train\_smote,y\_train\_smote = smote.fit\_resample(X\_train,y\_train)

Y\_train\_smote.value\_counts()

from sklearn.ensemble import RandomForestClassifier

model1=RandomForestClassifier()

model1.fit(X\_train\_smote,Y\_train\_smote)

y\_predit=model1.predict(X\_test)

rfc1=accurcy\_score(y\_test,y\_predit)

rfc1

pd.crosstab(y\_test,y\_predit)

print(classification\_report(y\_test,y\_predict))

from sklearn.tree import DecisionTreeclassifier model4-DecisionTreeClassifier() model4.fit(x\_train\_smote, y\_train\_smote) y\_predict=model4.predict(x\_test) dtc1=accuracy\_score (y\_test,y\_predict) dtc1 pd.crosstab(y\_test,y\_predict) print (classification\_report (y\_test, y\_predict))

from sklearn.neighbors import KNeighborsClassifier model2=KNeighborsClassifier() model2.fit(x\_train\_smote, y\_train\_smote) y\_predict = model2.predict(x\_test) knn1=(accuracy\_score (y\_test, y\_predict)) knn1 pd.crosstab(y\_test,y\_predict) print(classification\_report (y\_test, y\_predict))

from sklearn.linear\_model import LogisticRegression model5 Logistic Regression() models.fit(x\_train\_smote, y\_train\_smote) y\_predict model5.predict(x\_test) logit accuracy\_score(y\_test, y\_predict) logil pd.crosstab(y\_test,y predict) print(classification\_report (y\_test, y\_predict))

import tensorflow.keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

# Initialising the ANN

classifier = Sequential()

# Adding the input layer and the first hidden layer

classifier.add(Dense (units=100, activation='relu', input\_dim=10))

# Adding the second hidden layer

classifier.add(Dense (units=50, activation='relu"))

# Adding the output layer

classifier.add(Dense (units=1, activation="sigmoid"))

# Compiling the ANN

classifier.compile(optimizer='adam',

loss='binary\_crossentropy', metrics=['accuracy'])

# Fitting the ANN to the training set model\_history = classifier.fit(x train, y train, batch\_size=100, validation\_split=0.7, epochs=106)

#Age Gender Total Bilrabin-Direct Lrubin Alkaline\_ Phosphatase Alant \_AminotransferoseAsparaty\_Aminot mode14.predict([[50,1,1.2,0.8,150,79,00,7.2.3.4,0.8]))

D:\Anaconda\lib\site-packages\sklearn\base.py:450: Userlarning: x does not have valid feature names, but DecisionTreeCla ifier was fitted with feature nanes warnings.warn

( array([1], dtype-Int64)

# Age Gender Total Bilrabin-Direct Lrubin Alkaline\_ Phosphatase Alant\_ Aminotransferose Asparaty\_ Aminot model1.predict([[50,1,1.2,0.8,150,79,00,7.2.3.4,0.8]))

classifier.save("liver.h5")

y\_pred= classifier.predict(X\_test)

4/4 [======================] –n 0s 2ms/step

y\_pred

y\_pred (y\_pred > 0.5)

y\_pred

def predict\_exit(sample\_value):

#Convert list to numpy array

sample\_value = np.array(sample\_value)

#Reshape because sample value contains only 1 record

sample\_value = sample value.reshape(1, -1)

#Feature Scaling sample\_value = scale(sample\_value)

return classifier.predict(sample\_value)

#Age Gender Total Bilrubin Direct Bilrubin Alkaline Phosphotase

sample\_value = [[50,1,1.2,0.8,150,70,80,7.2,3.4,0.8]]

if predict\_exit(sample\_value)>0.5:

print("Prediction: Liver Patient')

else:

print("Prediction: Healthy ')

acc\_smote= [["KNN Classifier', knn1], ['RandomForestClassifier', rfc1], 2 ['DecisionTreeClassifier', dtc1],["LogisticRegression, logil]]

Liverpatient\_pred= pd.DataFrame(acc\_smote, columns = ['classification models', 'accuracy\_score']) Liverpatient pred

plt.figure(figsize-(7,5))

plt.xticks(rotation-90)

plt.title("Classification models & accuracy scores after SHOTE, fontsize=18)

sns.barplot(x="classification models", y accuracy score", data-Liverpatient pred, palette "Set2")

from sklearn.ensemble import ExtraTreesClassifier

model-ExtraTreesClassifier()

model.fit(x,y)

ExtraTreesClassifier()

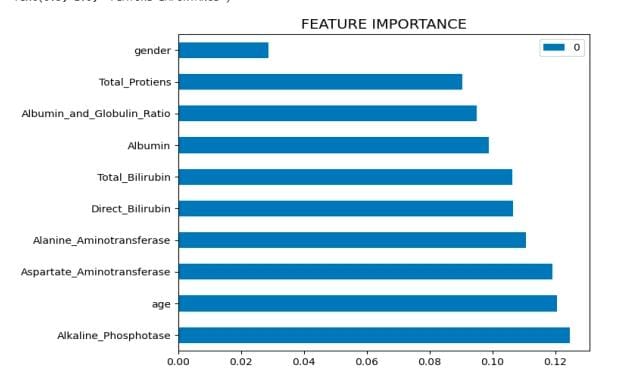
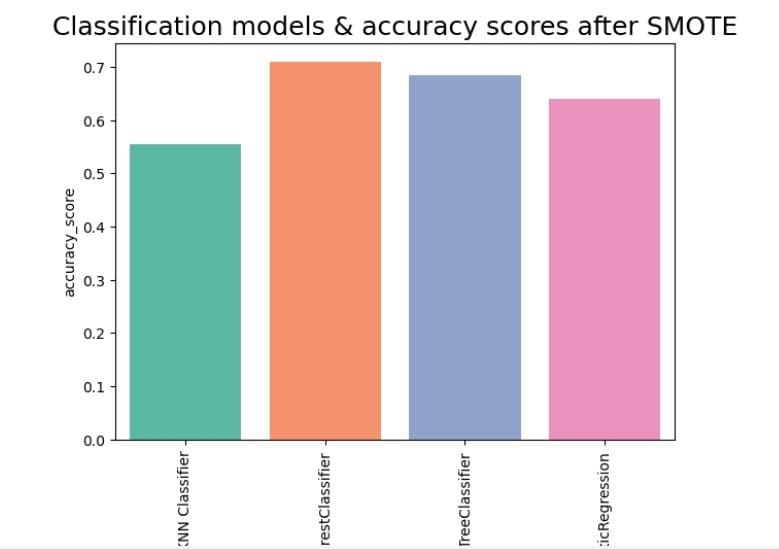
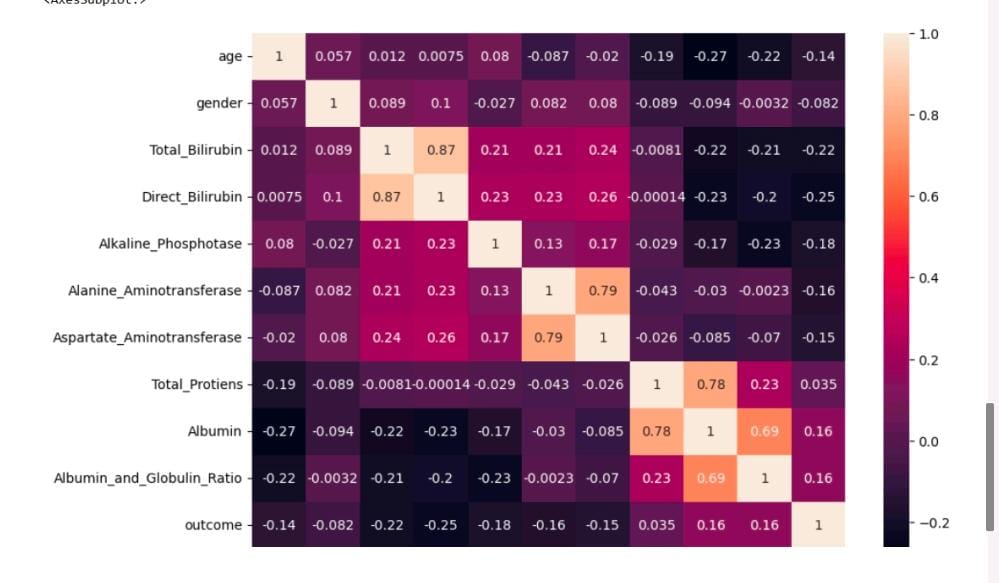
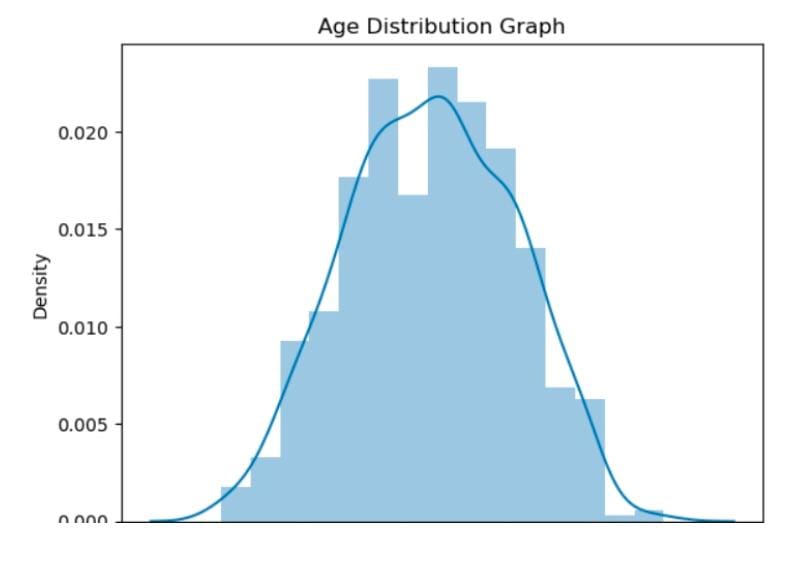
model.feature\_importances\_

dd.plot(kind-"barh', figsize-(7,6))

plt.title("FEATURE IMPORTANCE", fontsize-14)

import Joblib.

joblib.dump (model1, ETC.pk1')

**OUTPUT****