DATASET IMPROVEMENT USING OUTLIER REMOVAL

Major project report submitted in partial fulfillment of the

requirements for the award of degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

GUNNAM RISHITHA (221810303018)

SREEMAN RAGHAVA PODICHETI (221810303053)

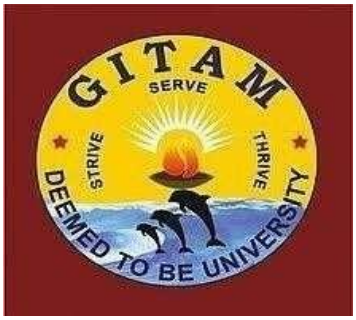
BURRA ARAVIND (221810303008)

DONDA GANGADHAR RAJU (221810303015)

Under the Esteemed Guidance of

MR. JETHYA ROOPAVATH

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

(Declared as Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS

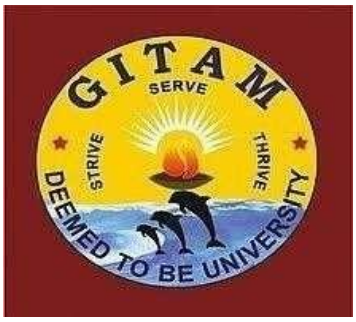
APRIL-2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

(Declared as Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS APRIL-2022



DECLARATTION

We, hereby declare that the major project report entitled "**DATASET IMPROVEMENT USING OUTLIER REMOVAL**" is an orginal work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of "Bachelor of Technology" in Computer Science and Engineering. The work had not submitted to any other college or University for the award of any degree or diploma.

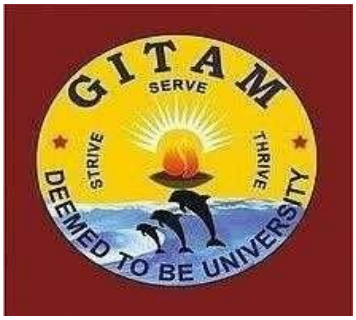
Name	**Registration No.**	**Signature's**
GUNNAM RISHITHA	221810303018	
SREEMAN RAGHAVA PODICHETI	221810303053	
BURRA ARAVIND	221810303008	
DONDA GANGADHAR RAJU	221810303015	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT(GITAM)

Declared as Deemed-to-be-University u/s 3 of UGC Act 1956)

HYDERABAD CAMPUS APRIL-2022



CERTIFICATE

This is to certify that the major project entitled "**DATASET IMPROVEMENT USING OUTLIER REMOVAL**" is the bonofide done by **GUNNAM RISHITHA (221810303018), SREEMAN RAGHAVA PODICHETI (221810303053), BURRA ARAVIND (221810303008), DONDA GANGADHAR RAJU (221810303015)**. In the Department of "**COMPUTER SCIENCE ENGINEERING**" at GITAM University, Hyderabad. It is faithful record work carried out by batch at the **Computer Science Engineering Department**, GITAM School of Technology, GITAM Deemed to be University, Hyderabad Campus under my guidance and supervision.

Mr. Jethya Roopavath	**Dr. S. Aparna**	**Prof.S. Phani Kumar**
Assistant Professor	Assistant Professor	Professor and HOD
Department of CSE	Department of CSE	Department of CSE

ACKNOWLEDGEMENT

This major project would not have been successful without the help of several people. We would like to thank the personalities who were part of project in numerous ways, those who gave us outstanding support from the birth of the reviews.

We are extremely thankful to our honorable **Pro-Vice Chancellor, Prof. Siva Prasad** for providing necessary infrastructure and resources for the accomplishment of this seminar.

We are very much obliged to our beloved **Prof. S. Phani Kumar**, Head of the Department of Computer Science and Engineering for providing the opportunity to undertake this project and encouragment in completion of this seminar.

We hereby wish to express a deep sense of gratitude to **Mr.Jethya Roopavath**, Assitant Professor, Department of Computer Science and Enigneering, for their esteemed guidance, moral support and invaluable advice provide by them for the success of this internship.

We are also thankful to all the staff members of Computer Science and Engineering department who have cooperated in making this project a success. We would like to thank my parents and friends who extended their help, encouragement and moral support either directly or indirectly in this seminar.

CSEBNUM_C12

GUNNAM RISHITHA (221810303018)

SREEMAN RAGHAVA PODICHETI (221810303053)

BURRA ARAVIND (221810303008)

DONDA GANGADHAR RAJU (221810303015)

ABSTRACT

Outlier detection is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Outlier detection is also known as Anomaly Detection. When the data is fitted into a model, like e regression model, the Mean absolute error of the model will be relatively more, when there are outliers in the data. Mean absolute error can be reduced by removing the outliers. This project aims at removing the outliers in a dataset, by using Isolation Forest method and Ellecptic Envelope method, and verify the change in the Mean absolute error because of the removal of outliers by using these methods.

The project comprises of four modules. The first module is used to calculate the mean absolute error of the original dataset by fitting the dataset into a Regression model. The second module is used to remove the outliers in the dataset by using Isolation Forest method. The third module is used to remove the outliers in the dataset by using Ellecptic Envelope method. The fourth module is used to find out the mean absolute errors, after removing the errors.

The isolation Forest algorithm isolates each point in the data and splits them into outliers or inliers. This split depends on how long it takes to separate the points. If we try to segregate a point which is obviously a non-outlier, it'll have many points in its round, so that it will be really difficult to isolate. On the other hand, if the point is an outlier, it'll be alone and we'll find it very easily. This is the basic concept of the isolation forest algorithm. Elleptic envelope method fits an ellipse to the central data points, treating points outside the central mode, as outliers.

LIST OF FIGURES:

CHAPTER 1

INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT:

As the dataset contain the objects ,the task of outlier detection is to identify data object that are marketdly different from or imperfect labels, introduce likelihood values for each input data which denote the degree of membership of an example toward the normal and abnormal classes respectively. Practically outlier detection has been found in wide-ranging applications from fraud detection for credit cards, insurance or health care, intrusion detection for cyber-security, fault detection in safety critical systems, to military surveillance. Many outlier detection methods have been proposed to detect outliers from existing normal data. In general, the previous work on outlier detection can be broadly classified into distribution-based, clustering-based, density-based and model-based approaches, all of them with long history. This paper presents a novel outlier detection approach to address data with imperfect labels and incorporate limited abnormal examples into learning. Our main objectives are

- Make a research work on the existing algorithm for outlier detection with imperfect data labels
- Designing a unique and effective algorithm
- Testing the algorithm on different data labels
- The proposed scheme overcomes the drawbacks of existing scheme such as inefficiency and inaccuracy. It provides less search time and high retrieval accuracy.

Algorithms Used: Isolation Forest & Ellecptic Envelope algorithms.

Outputs from the project:

i. Mean absolute error of the original dataset.
ii. Mean absolute error of the dataset, after removing outliers using isolation forest.
iii. Mean absolute error of the dataset, after removing outliers using Elleptic envelope.
iv. Dataset shapes before and after removal of outliers.

In the existing system, Outlier removal is not given enough consideration in the data analysis, as a result of which the results of data analysis are relatively more erroneous. The proposed system tries to improve the data analysis by removing the outliers from the dataset.

Project Advantages:

1. The project is uesful to get accurate data analysis results by removing the outliers from the analysis process.
2. The project is useful to the data analysts to understand more about isolation forest and elleptic envelope methods.
3. The project is useful to reduce the mean absolute error in the data by removing the outliers.
4. Technical advantages:
5. Using latest python 'sklearn' module of Python to implement isolation forest and elleptic envelope methods.
6. Using Python, which is chosen as the best programming language, by the Programming Community.
7. More Functionality can be implemented with less number of lines of code in Python.
8. PyQt tool is used to create the Graphical User interfaces.
9. All the Front end code is generated automatically by PyUIC.

PROBLEM DEFINITION

Vision:

- The project aims at developing a tool for Dataset Improvement using Outlier Removal with all the above-mentioned advantages.

Mission:

- This tool is developed by using Python along with its layout toolkit PyQt, PyUIC and python’s sklearn.

CHAPTER 2

SYSTEM ANALYSIS

2.1 REQUIREMENT ANALYSIS:

2.1.1 Hardware Requirements

1. It requires a minimum of 2.16 GHz processor.
2. It requires a minimum of 4 GB RAM.
3. It requires 64-bit architecture.
4. It requires a minimum storage of 500GB.

2.1.2 Software Requirements

1. It requires a 64-bit windows Operating System.
2. Python Qt Designer for designing user interface.
3. SQLite3 server for storing database Entities.
4. Pyuic for converting the layout designed user interface (UI) to python code.
5. Python language for coding.

2.2 FEASIBILITY ANALYSIS:

As the name implies, a feasibility study is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

2.2.1 Economical Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision makers determine the positive economic benefits to the organization that the proposed project will provide. Our project is economically feasible because in this we have used “UBUNTU”, “PYTHON”, “PYQT” designer tool and “PYUIC” which are all available as an open source.

2.2.2 Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system. A prototype of the tool was developed to verify the technical feasibility. The prototype is working successfully and hence the project is feasible.

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM DESCRIPTION:

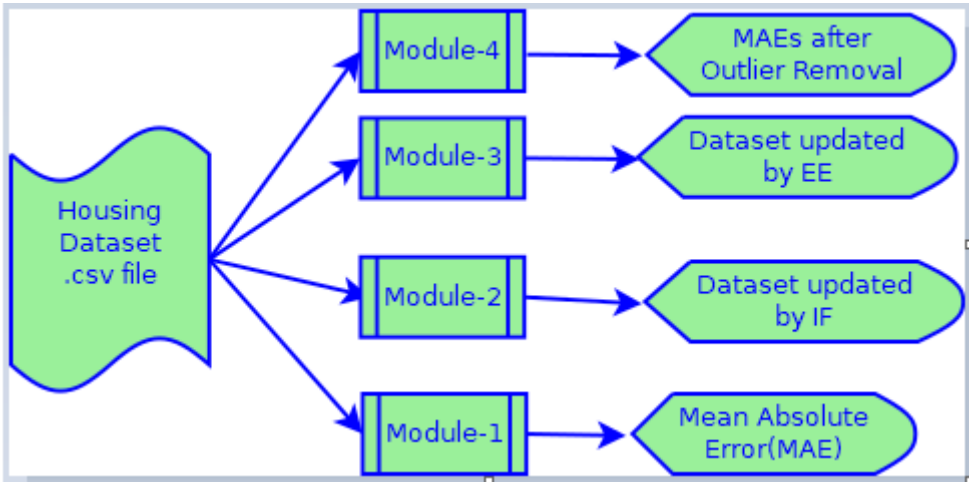


Figure 3.1.1: Architecture diagram of the project

The first module is used to calculate the mean absolute error of the original dataset by fitting the dataset into a Regression model. The second module is used to remove the outliers in the dataset by using Isolation Forest method. The third module is used to remove the outliers in the dataset by using Ellecptic Envelope method. The fourth module is used to find out the mean absolute errors, after removing the errors.

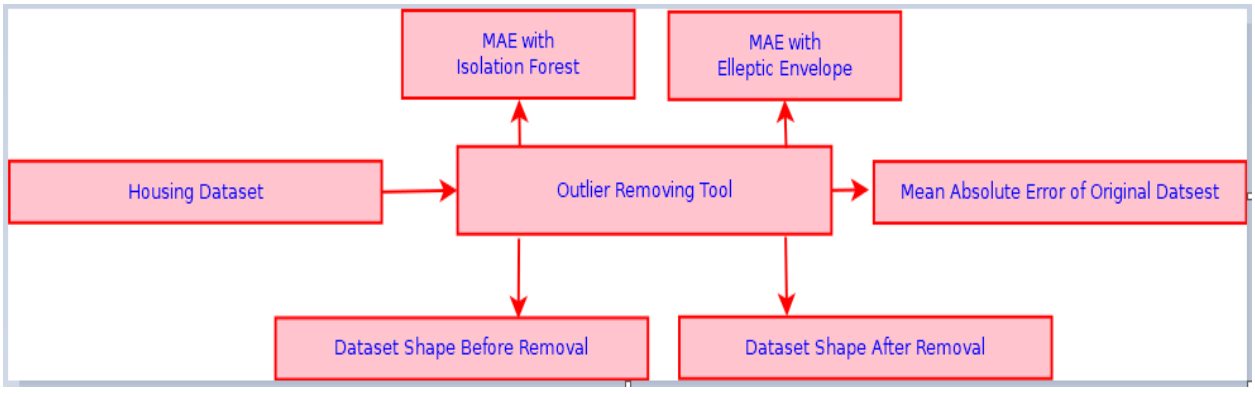


Figure 3.1.2: Data flow diagram

The housing dataset is to be provided as input to the outlier removal tool. Mean absolute errors, as well as the dataset shapes,before and after removal of outliers are produced as output..

Screen shots and descriptions:

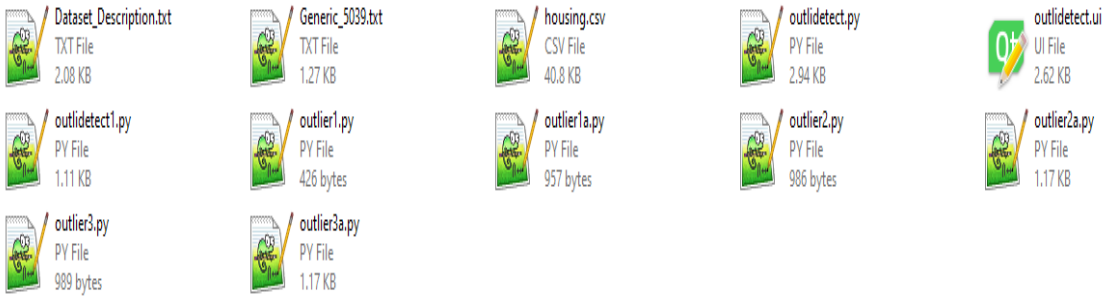


Figure 3.1.3: Screen shot showing the files created during the project

The above screen shot shows different files created during this projects. There are three different types of files: (1) .py files (2) .ui file (3).txt files

- .ui files are the user interface files, created by using PyQt layout editor
- .py files are python program files created either manually, or automatically. For instance, each .ui file has a corresponding .py file that is created automatically by using the PyUIC tool.
- .txt files contains the generic useful information about the project.

outlidetect1.py, is the entry program for this project. Execution of this python program leads to the entry screen as follows:

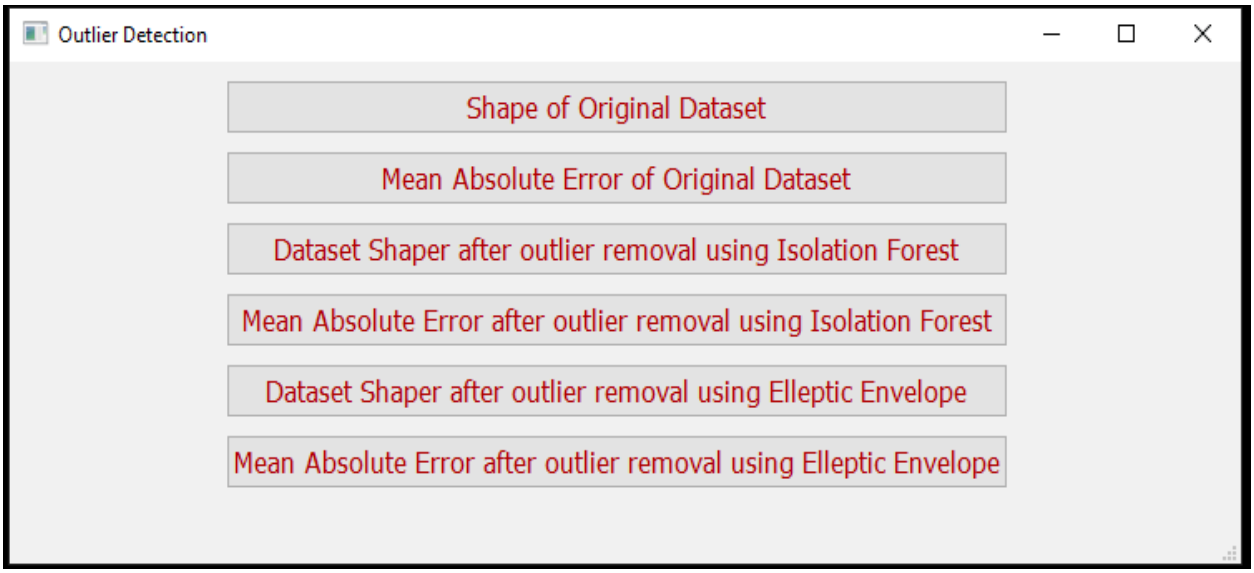


Figure 3.1.4: Entry screen of the project

The first two buttons are used to calculate the Mean Absolute Error and shape of the original dataset. The next two buttons are used to calculate the Mean Absolute Error and shape of the dataset after ouliers removal using Isolation Forest. The last two buttons are used to calculate the Mean Absolute Error and shape of the dataset after outliers removal using Elleptic Envelope.

PYTHON: Python is a widely used high level programming language for general purpose programming, created by Guido Van Rossum and first released in 1991. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than segregation brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. CPython, the reference implementation of Python, is open-source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the operating system Amoeba. Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL). About the origin of Python, Van Rossum wrote in 1996.

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting garbage collector and support for Unicode. With this release the development process was changed and became more transparent and community-backed. Python 3.0 (initially described as Python 3000 or py3k), is a major, backward-incompatible release that was released after a long period of testing on 3 December 2008. Many of its major features have been back ported to the backwards-compatible Python 2.6.x and 2.7.x version series.

The End-of-Life date (EOL, sunset date) for Python 2.7 was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code cannot easily be forward ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go trans compiler, which The Register speculated was in response to Python 2.7's planned end-of life but Google cited performance under concurrent workloads as their only motivation.

3.2 TOOLS USED:

3.2.1 Qt Designer:

Qt Designer is the tool used to create the Graphical user interfaces in this project. The tool can be found in the Libaray/Bin folder of Anaconda tool as shown in the following screen shot.

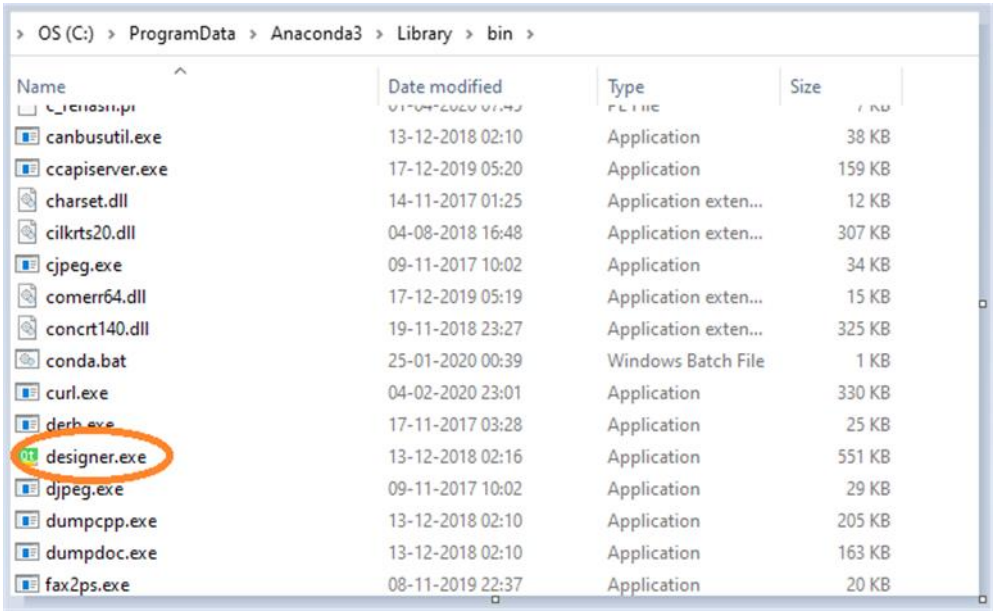
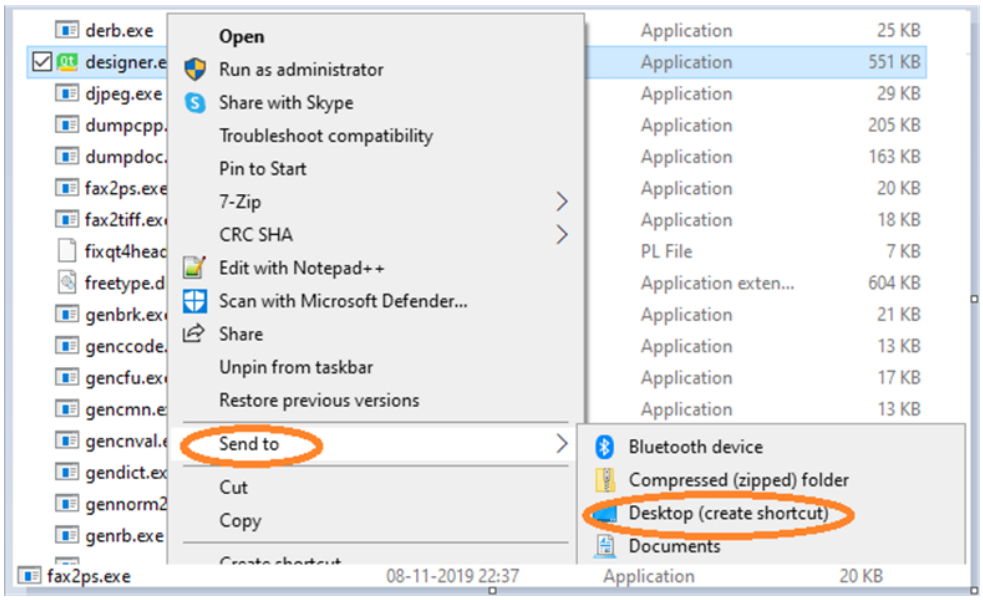
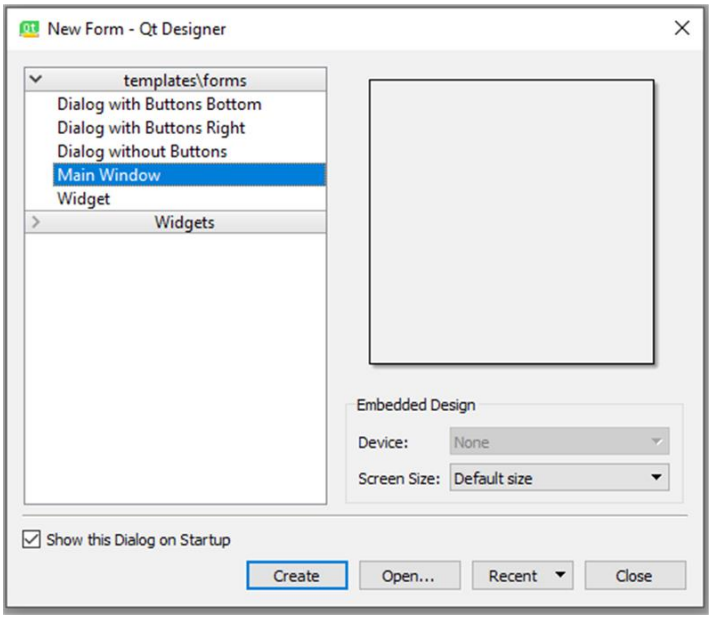


Figure 3.2.1: Designer Screen

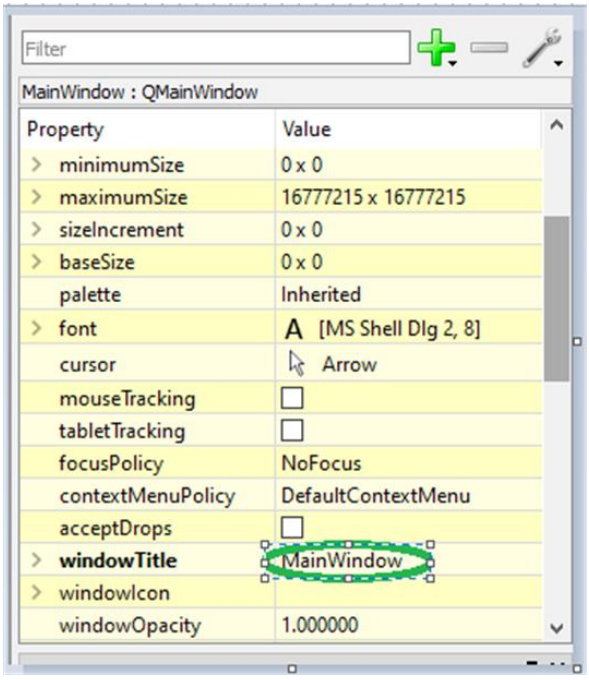
As this designer tool is used again and again, in the project, it's better to create a short cut on the desktop,by using a right-click on designer.exe, as shown in the following figure.



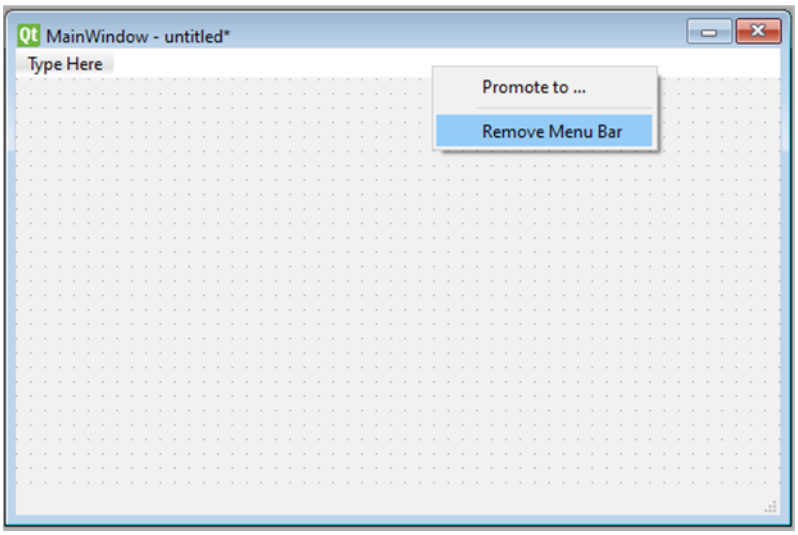
Once the designer tool is opened, it results in a screen as shown below.



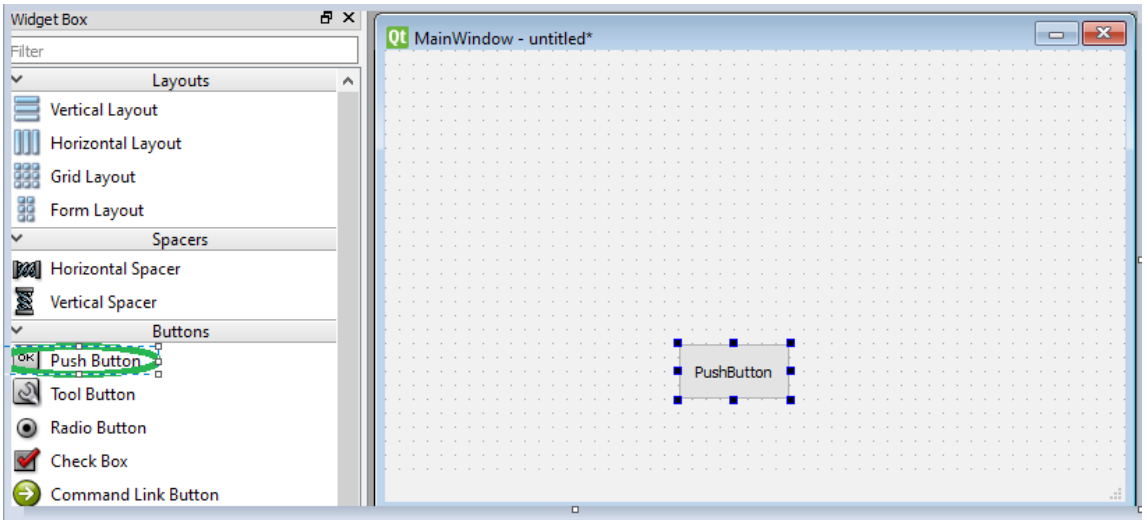
The main window can be created by clicking on the Create button, after highlighting the main window (shown in blue color, in the above figure).The main window can be given a title, by using the properties window, that is in the right hand side middle portion, as shown below.



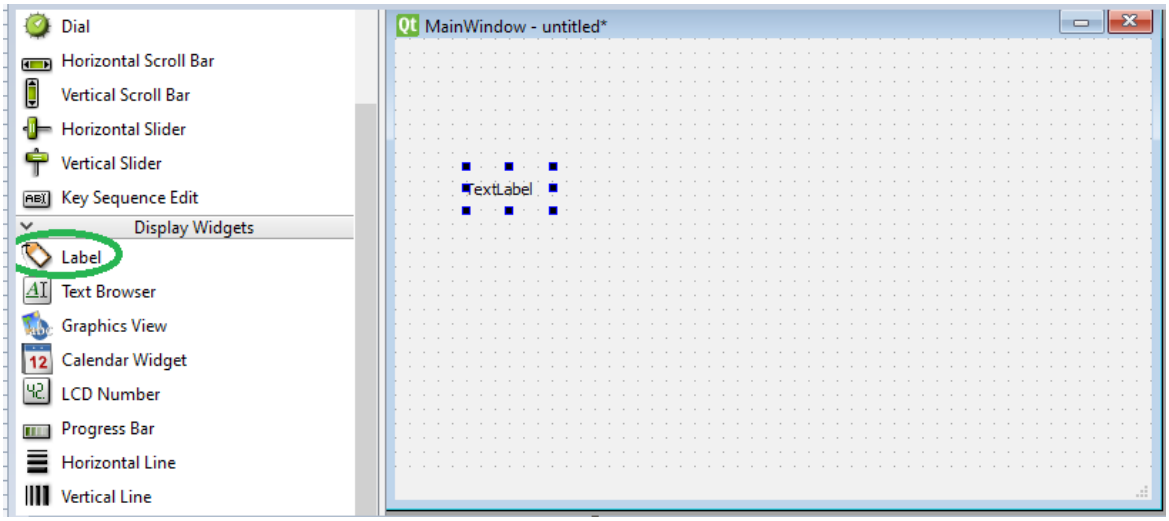
The menu bar in the main window can be removed by right clicking in the white portion after 'Type here'label, as shown in the below figure.



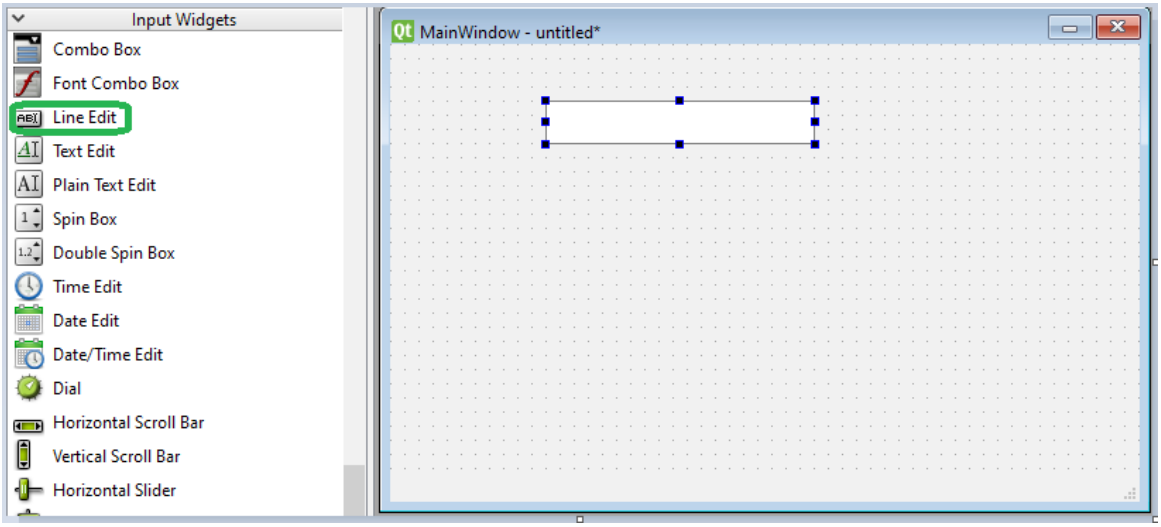
A push button can be created by dragging and dropping it, from the buttons section, on the left hand side as shown in below figure.



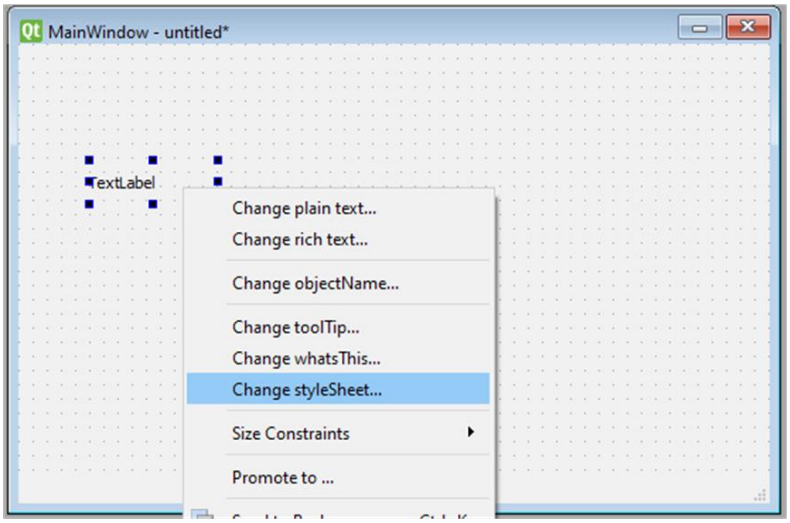
Likewise, A Label can be created by dragging and dropping it, from the Display Widgets section, on the left hand side as shown in below figure.



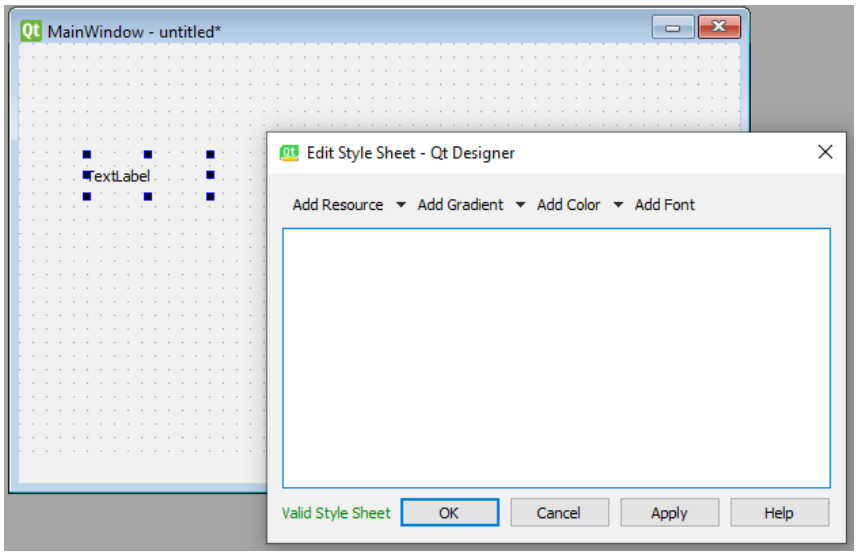
Likewise, A Line edit can be created by dragging and dropping it, from the Input Widgets section, on the left hand side as shown in below figure.



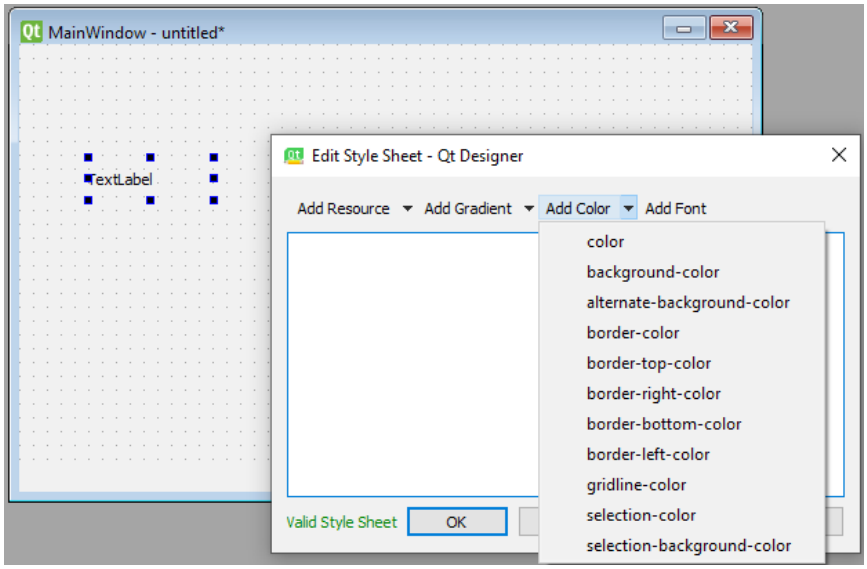
The color and of a pushbutton/label can be changed by right clicking on it and selecting the 'change style' option, as shown in the following figure.



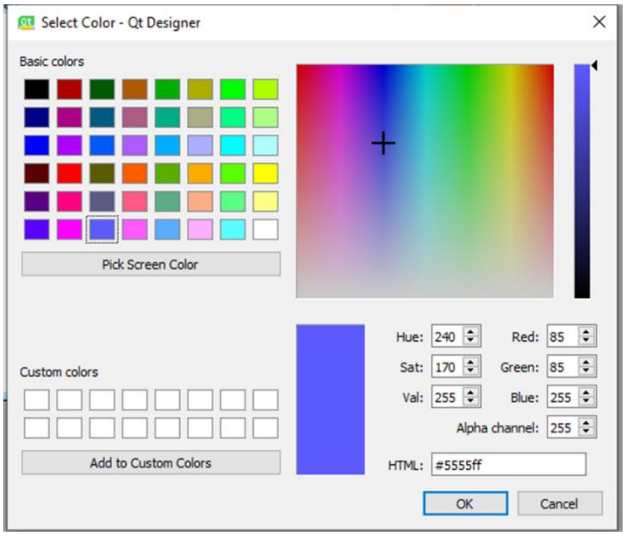
Selecting the change style sheet option, results in the following screen.



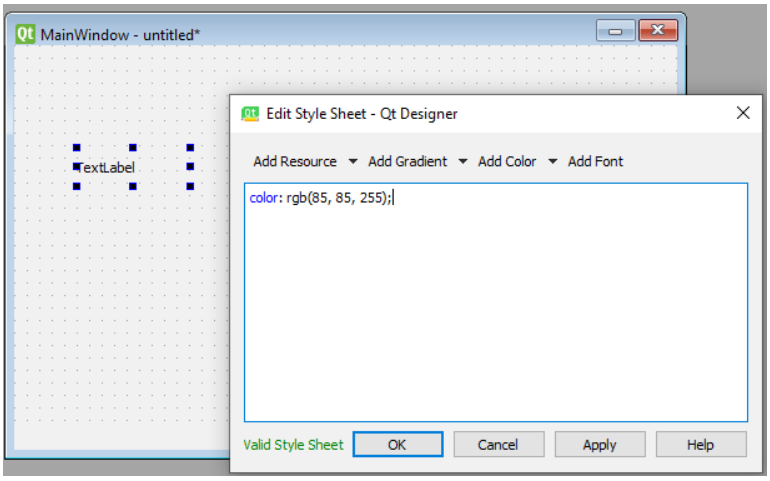
Clicking on the inverted triangle symbol, on the right of the ‘Add color’ option results in the following screen.



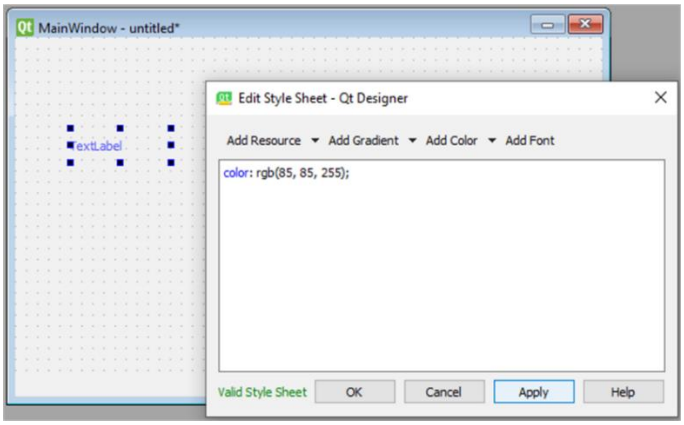
Choosing the color option results in the following screen.



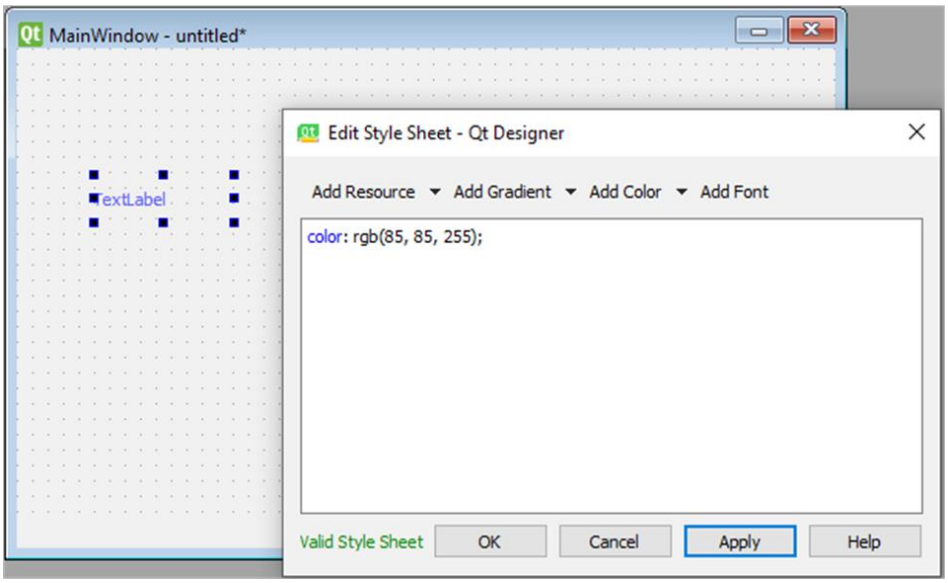
The user can choose any of his/her interesting color from the color palatte, and click ‘ok’ buttton, in order to get the following screen.



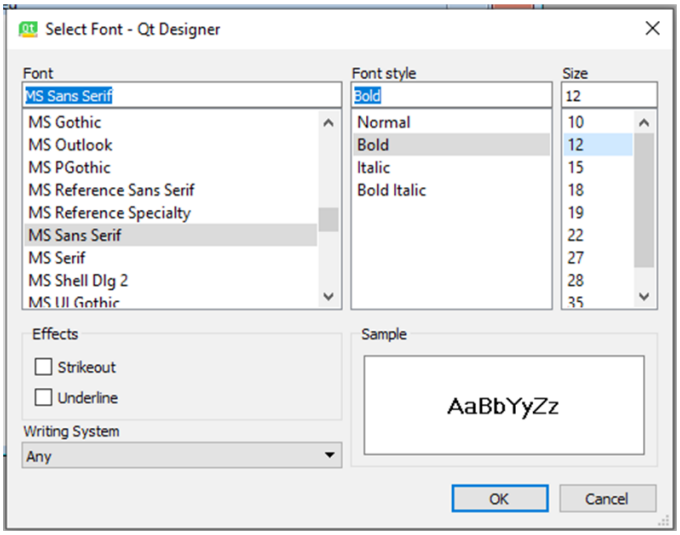
Now the user has to click on ‘Apply button’, in oder to change color of the text label, as shown in the following figure.



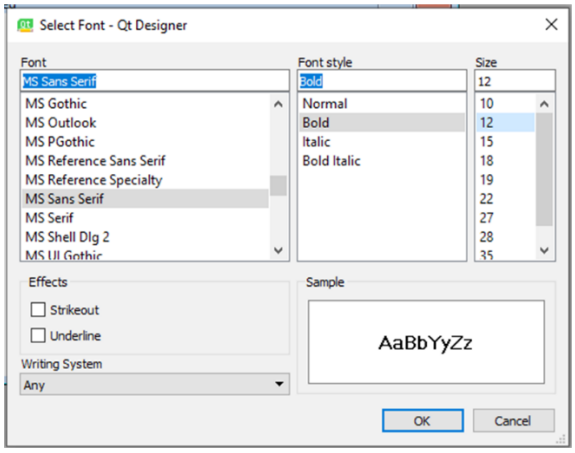
The user can change the font of the label/pushbutton, by clicking on the Addfont option, as follows:



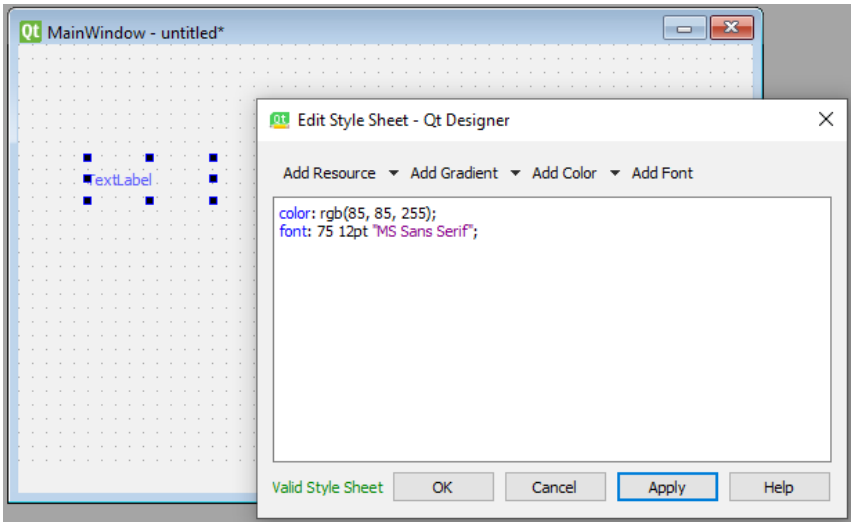
Clicking on the Add font option, will result in the following screen.



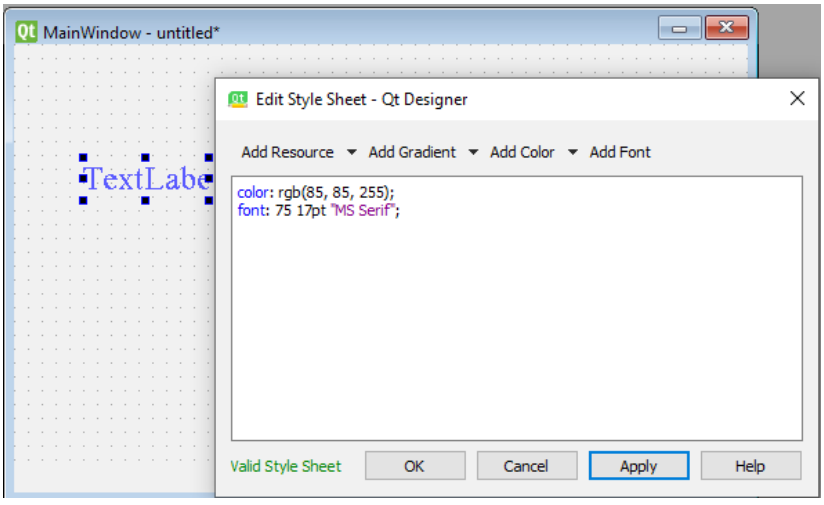
The user can choose his interested font, font style and size and click on ‘ok’ button to get the following screen.



Clicking on ‘ok’ button will result in the following screen.

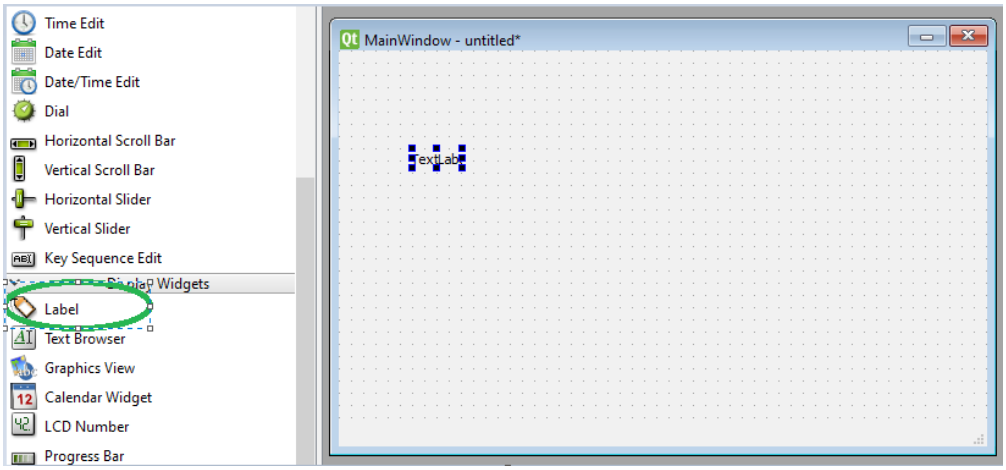


The user needs to click on ‘Apply’ button to change the font, as shown in the following figure.

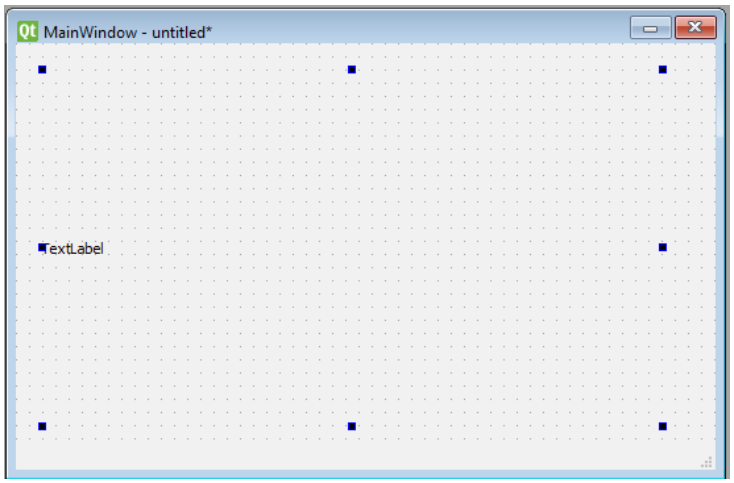


3.2.2 Steps for Inserting a back ground image:

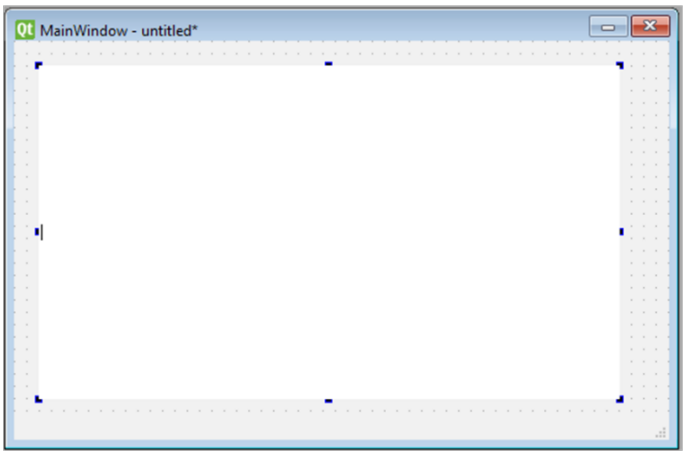
Drag and drop a text label, as shown in the following figure.



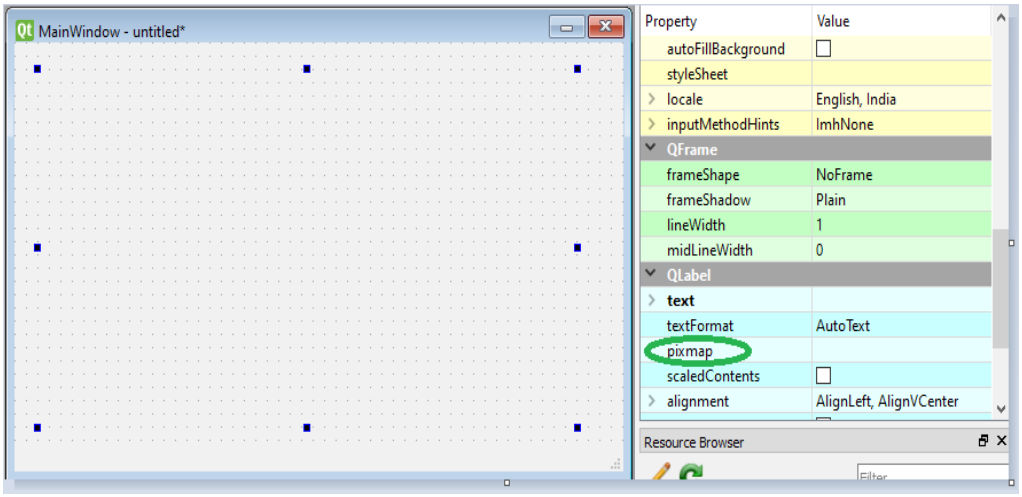
Expand the textlabel by dragging it's corners, and make it big as shown in the following figure.



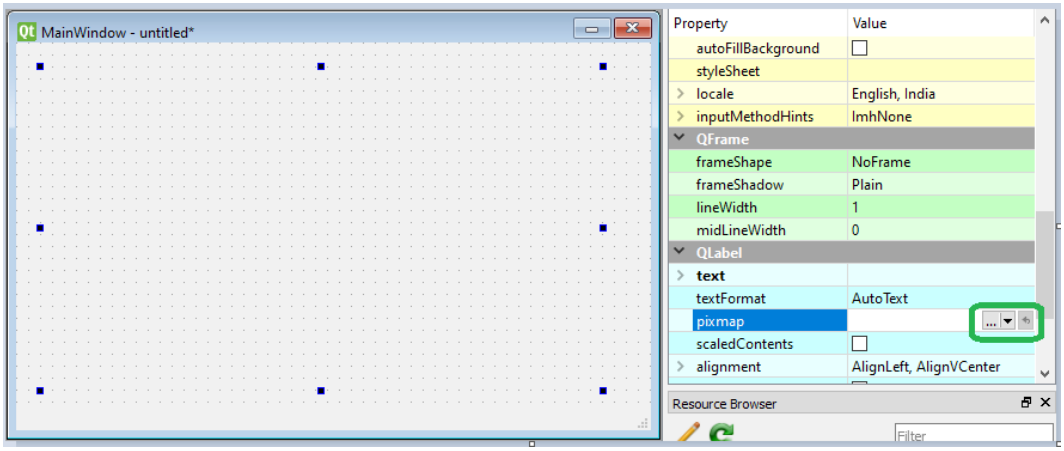
Double click on the Textlabel, and remove the textlabel, from the figure by using backspace button, as shown below.



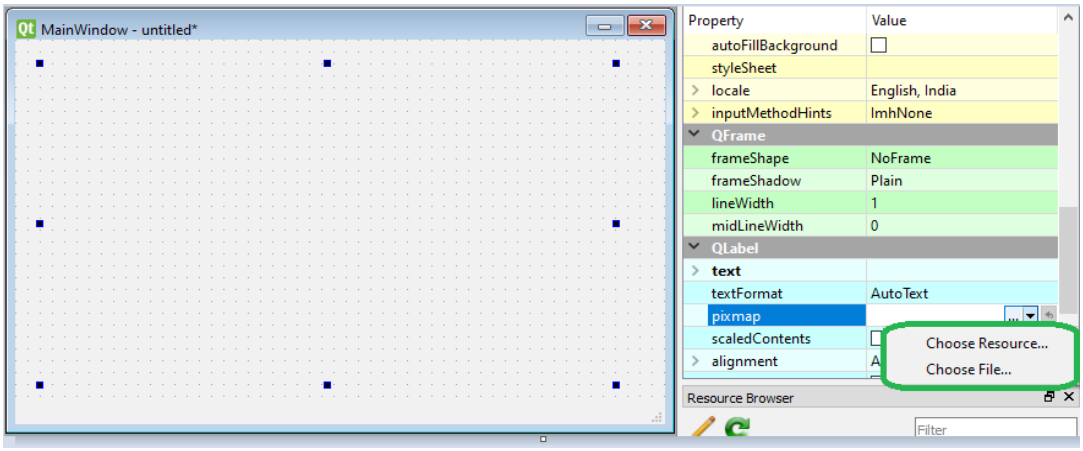
Scroll down the property window,in the left hand side middle, till you get pixmap property, as shown below.



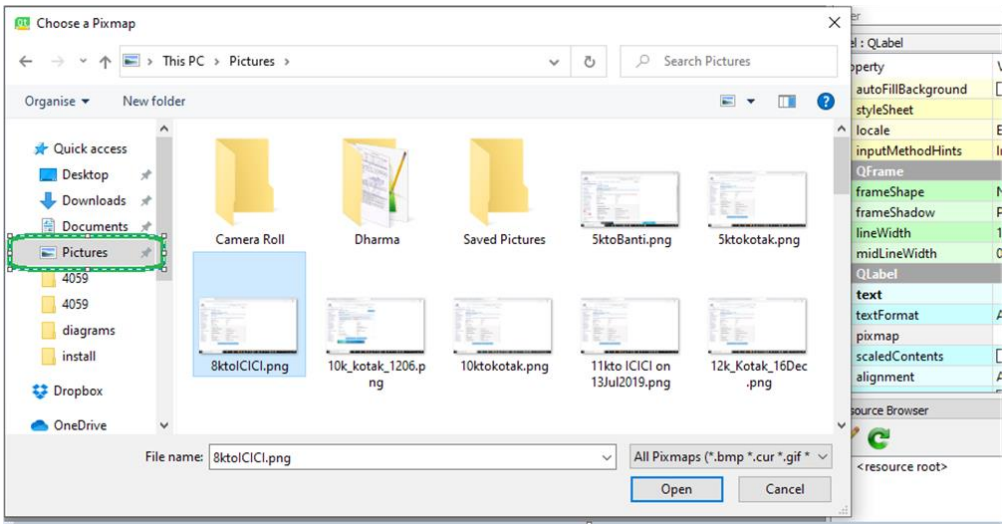
Click on the empty space below 'Autotext' and against 'pixmap' to get the following screen, with the options highlighted in green color.



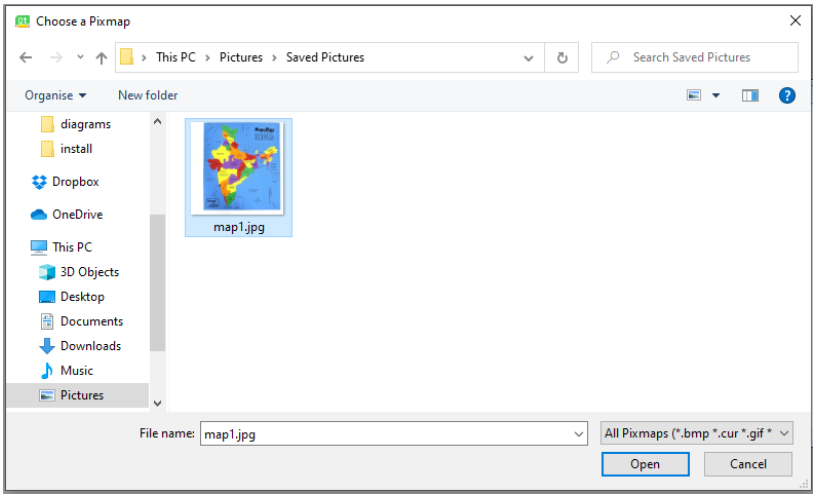
Click on the Downarrow to get 'choose file' and 'choose Reources' options, as shown below.



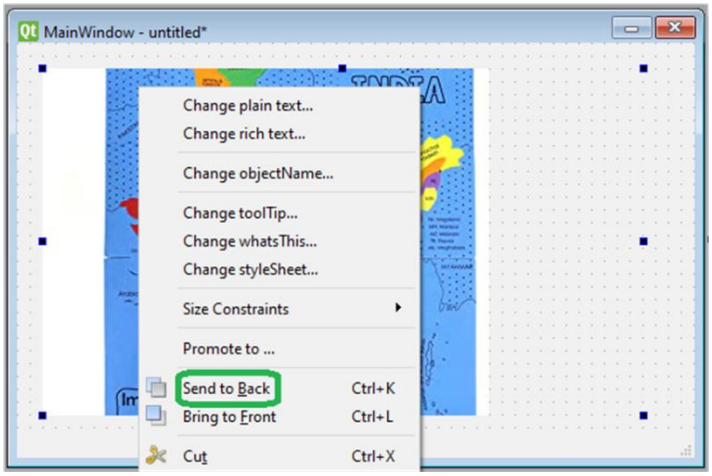
Click on choose file, and select any image file containing a back ground image that is relevant to the project.



After choosing appropriate image,click on ‘open’ to get the image in the text label field.



Right click on the image to get ‘send to back’ option, as shown below.

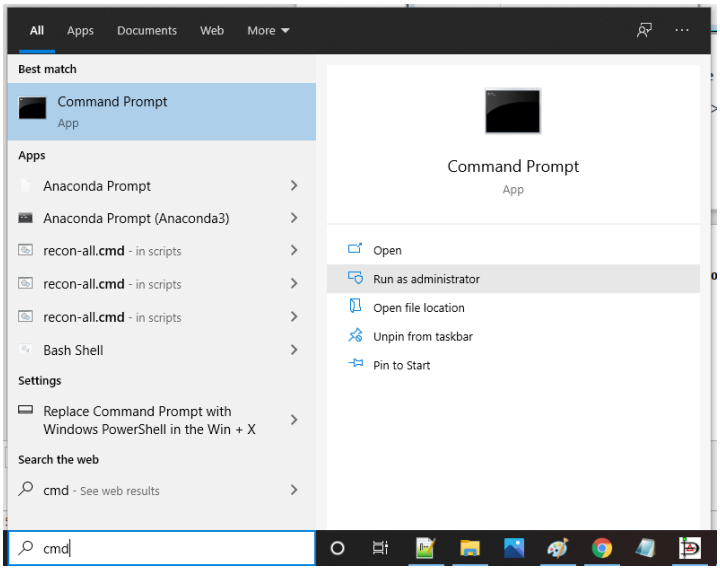


Select the send to back option, in order to make the image as a back ground image.

Save all the changes by using Ctrl+S, or, by using File ---> Save menu option.

3.2.3 Opening Command Prompt:

Open the command prompt in adminstrative mode as shown below.



Go to the project folder in the command prompt by using cd command, and give the following command to incorporate automated code changes in the project pyuic5 filename.ui -o filename.py.

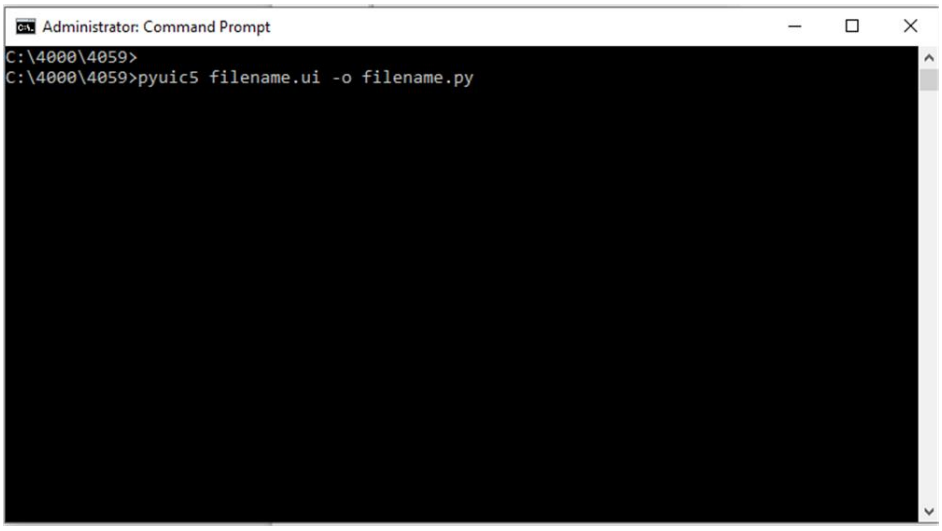


Figure 3.2.3: Command Prompt Screen

3.3 UML DIAGRAMS:

3.3.1 Use Case Diagram:

UML stands for Unified Modeling Language.

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

As we can see the user is interacting with system by providing the housing dataset as input. Mean absolute errors, as well as the dataset shapes,before and after removal of outliers are to be calculated after providing the housing dataset as input.

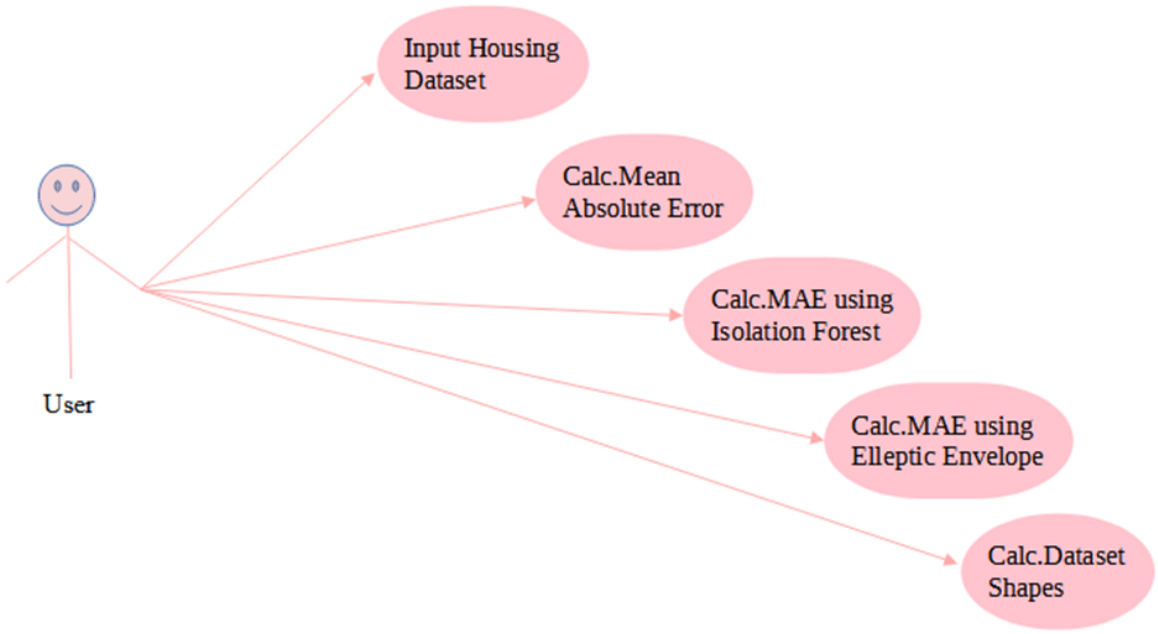


Figure 3.3.1: Use Case Diagram

3.3.2 Sequence Diagram:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

From above mentioned sequence diagram we have to go in sequence: The housing dataset is to be provided as input to the outlier removal tool. Mean absolute errors, as well as the dataset shapes,before and after removal of outliers are produced as output.

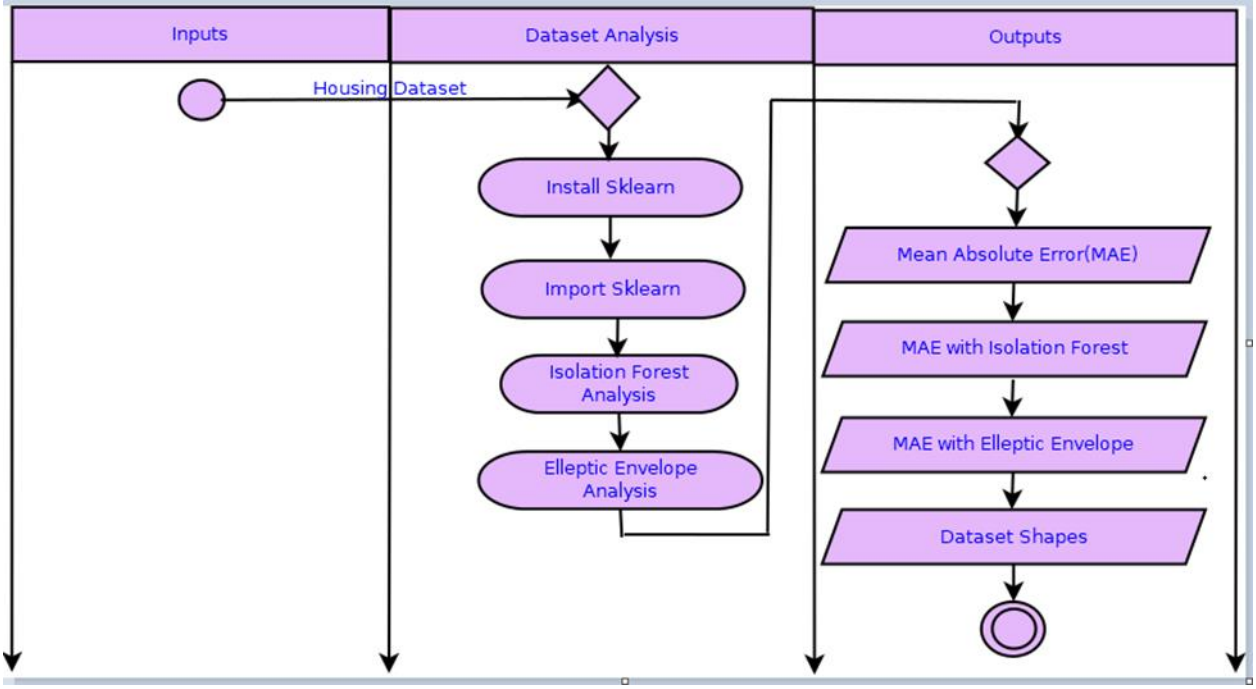


Figure 3.3.2: **Sequence** ***Diagram***

3.3.3 Activity Diagram:

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. In activity diagram The housing dataset is to be provided as input to the outlier removal tool. Mean absolute errors, as well as the dataset shapes,before and after removal of outliers are produced as outputs after performing the analysis using sklearn tool.

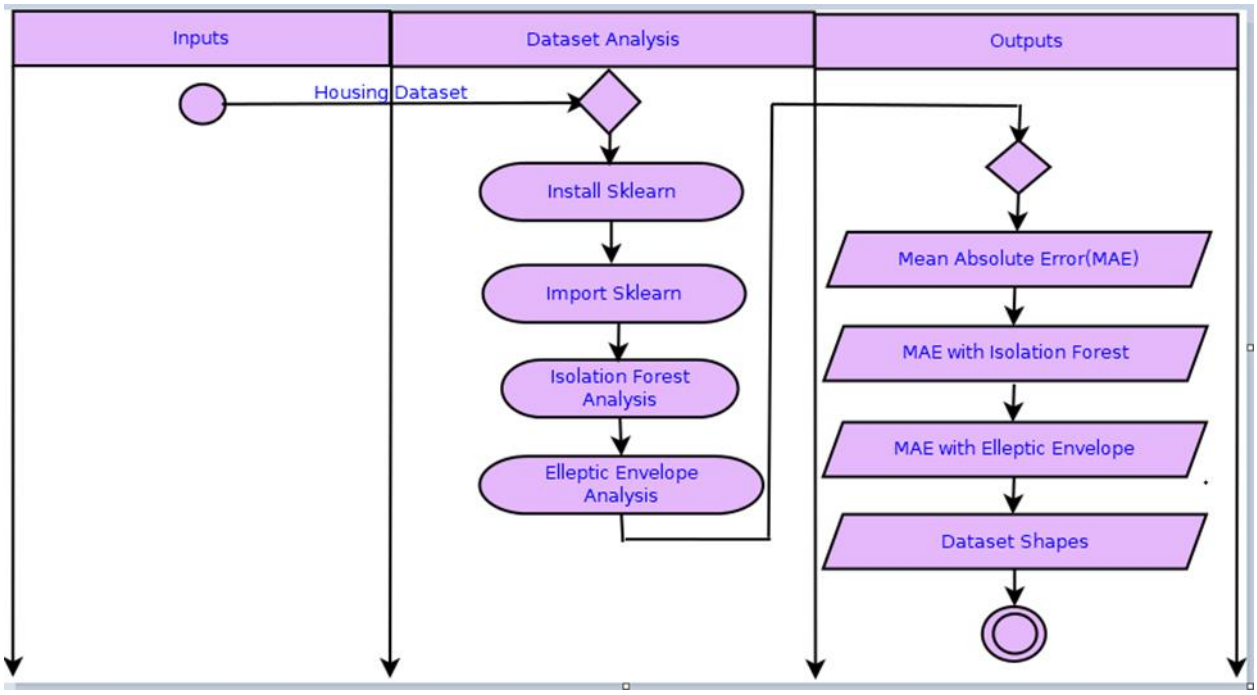


Figure 3.3.3: **Activity** ***Diagram***

CHAPTER 4

PROJECT IMPLEMENTATION

4.1 SOFTWARES USED

4.1.1 Python

Python was conceived in the late 1980s, and its implementation began in December 1989 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations.

4.1.2 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable

for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda.This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

4.1.3 VS Code

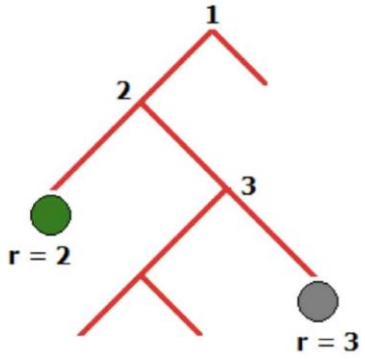
Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

4.2 ALGORITHMS USED

4.2.1 Isolation Forest

Isolation Forest or IForest is a popular Outlier Detection algorithm that uses a tree-based approach. The general concept is based on randomly selecting a feature from the dataset and then randomly selecting a split value between the maximum and minimum values of the feature. Thus, you will be able to isolate and calculate the isolation path for every sample in your dataset. For example, the isolation path for the green dot in the picture below will be 2 whereas for the gray dot it will be 3.



Such randomization guarantees that outliers will have shorter isolation paths. However, you can not be sure you found an outlier based on a single tree. To do that you need to build many trees. That is why IForest just as the name states requires building plenty of trees (Forest) and checking isolation paths of samples. If many trees have a short isolation path for a particular sample, it is likely to be an outlier.

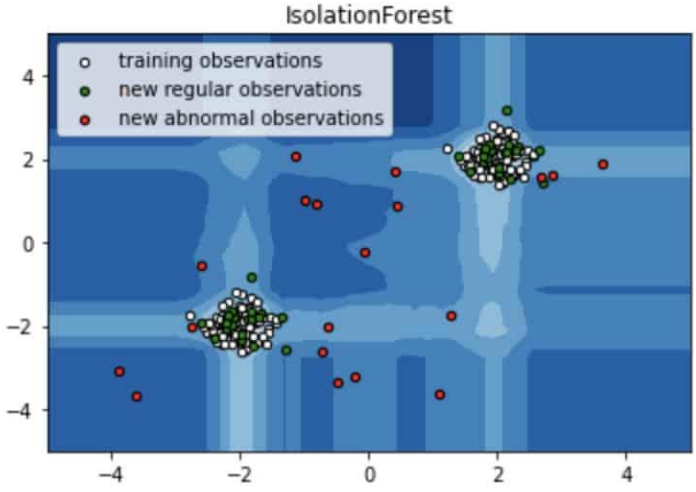


Figure 4.2.1: Isolation Forest

4.2.2 Ellecptic Envelope

If you have taken geometry classes you are probably familiar with ellipse — a geometric configuration that takes an oval shape on a two-dimensional plane. So the algorithm — Elliptical Envelope— creates an imaginary elliptical area around a given dataset. Values that fall inside the envelope are considered normal data and anything outside the envelope is returned as outliers. So, naturally, the red data points in the above diagram should be identified as outliers by this algorithm. As evident from this figure, the algorithm works best if data has a Gaussian distribution. This algorithm is used when the data is Gaussian distributed. In this it is like this model converts the data into elliptical shape and the points which are far away from this shape coordinates are considered outliers and for this minimum-covariance- determinant is found out. It is like when finding the covariance in the dataset so that minimum is excluded and which is higher those points are considered anomalies. The below picture depicts the explanation for this algorithm detection.

It has the same line of code as just to fit the data and predict on the same which identifies the anomalies in the data where -1 is allotted for anomalies and +1 for normal data or in-liers. Elliptic Envelope in the supervised model requires the target variable to be set to 1 for inliers and a value of -1 for the outlier. Contamination: specify the proportion of the outlier present in the dataset. The range is between (0, 0.5).

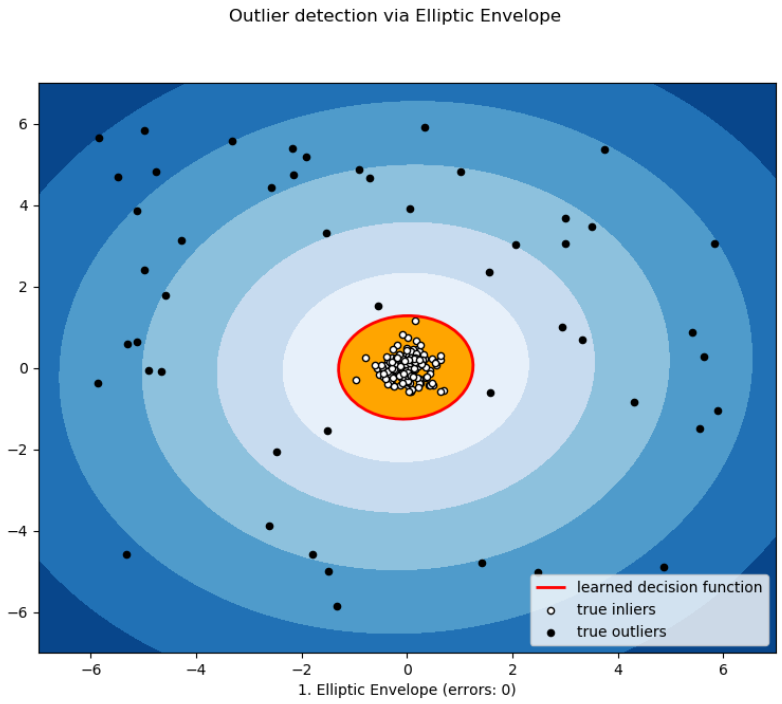


Figure 4.2.2: Elleptic Envelope

4.3 SAMPLE CODE

outlidetect1.py

```
import sys
import os
from outlidetect import *
from PyQt5 import QtWidgets, QtGui, QtCore

class MyForm(QtWidgets.QMainWindow):
  def __init__(self,parent=None):
     QtWidgets.QWidget.__init__(self,parent)
     self.ui = Ui_MainWindow()
     self.ui.setupUi(self)
     self.ui.pushButton.clicked.connect(self.oshape)
     self.ui.pushButton_6.clicked.connect(self.ishape)
     self.ui.pushButton_3.clicked.connect(self.imae)
     self.ui.pushButton_4.clicked.connect(self.eshape)
     self.ui.pushButton_5.clicked.connect(self.omae)
     self.ui.pushButton_7.clicked.connect(self.emae)


  def oshape(self):
    os.system("python outlier1.py")

  def ishape(self):
    os.system("python outlier2.py")

  def imae(self):
    os.system("python outlier2a.py")

  def eshape(self):
    os.system("python outlier3.py")

  def omae(self):
    os.system("python outlier1a.py")

  def emae(self):
    os.system("python outlier3a.py")


if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyForm()
    myapp.show()
    sys.exit(app.exec_())
```

outlier1.py

```
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
df = pd.read_csv('housing.csv')
from sklearn.model_selection import train_test_split
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# summarize the shape of the dataset
print('Shape of the dataset is:',X.shape, y.shape)
```

outlier1a.py

```
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
df = pd.read_csv('housing.csv')
from sklearn.model_selection import train_test_split
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# summarize the shape of the dataset
#print(X.shape, y.shape)
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the train and test sets
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
# fit the model
model = LinearRegression()
model.fit(X_train, y_train)
# evaluate the model
yhat = model.predict(X_test)
# evaluate predictions
mae = mean_absolute_error(y_test, yhat)
print('MAE: %.3f' % mae)
```

outlier2.py

```
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.metrics import accuracy_score
df = pd.read_csv('housing.csv')
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import IsolationForest
from sklearn.metrics import mean_absolute_error
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the training dataset
print(X_train.shape, y_train.shape)
# identify outliers in the training dataset
iso = IsolationForest(contamination=0.1)
yhat = iso.fit_predict(X_train)
# select all rows that are not outliers
mask = yhat != -1
X_train, y_train = X_train[mask, :], y_train[mask]
# summarize the shape of the updated training dataset
print(X_train.shape, y_train.shape)
```

outlier2a.py

```
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.metrics import accuracy_score
df = pd.read_csv('housing.csv')
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import IsolationForest
from sklearn.metrics import mean_absolute_error
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the training dataset
print(X_train.shape, y_train.shape)
```

```
# identify outliers in the training dataset
iso = IsolationForest(contamination=0.1)
yhat = iso.fit_predict(X_train)
# select all rows that are not outliers
mask = yhat != -1
X_train, y_train = X_train[mask, :], y_train[mask]
# summarize the shape of the updated training dataset
#print(X_train.shape, y_train.shape)
# fit the model
model = LinearRegression()
model.fit(X_train, y_train)
# evaluate the model
yhat = model.predict(X_test)
# evaluate predictions
mae = mean_absolute_error(y_test, yhat)
print('MAE: %.3f' % mae)
```

outlier3.py

```
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.metrics import accuracy_score
df = pd.read_csv('housing.csv')
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.covariance import EllipticEnvelope
from sklearn.metrics import mean_absolute_error
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the training dataset
print(X_train.shape, y_train.shape)
# identify outliers in the training dataset
ee = EllipticEnvelope(contamination=0.01)
yhat = ee.fit_predict(X_train)
# select all rows that are not outliers
mask = yhat != -1
X_train, y_train = X_train[mask, :], y_train[mask]
# summarize the shape of the updated training dataset
print(X_train.shape, y_train.shape)
```

outlier3a.py

```
import numpy as np
import pandas as pd
from sklearn import *
from sklearn.metrics import accuracy_score
df = pd.read_csv('housing.csv')
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.covariance import EllipticEnvelope
from sklearn.metrics import mean_absolute_error
# retrieve the array
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=1)
# summarize the shape of the training dataset
print(X_train.shape, y_train.shape)
# identify outliers in the training dataset
ee = EllipticEnvelope(contamination=0.01)
yhat = ee.fit_predict(X_train)
# select all rows that are not outliers
mask = yhat != -1
X_train, y_train = X_train[mask, :], y_train[mask]
# summarize the shape of the updated training dataset
#print(X_train.shape, y_train.shape)
# fit the model
model = LinearRegression()
model.fit(X_train, y_train)
# evaluate the model
yhat = model.predict(X_test)
# evaluate predictions
mae = mean_absolute_error(y_test, yhat)
print('MAE: %.3f' % mae)
```

4.4 MODULES:

4.4.1 OS Module:

The OS module in Python provides a way of using operating system dependent functionality.

The functions that the OS module provides allows you to interface with the

underlying operating system that Python is running on – be that Windows, Mac or Linux.

OS functions:

Executing a shell command

os.system()

Get the users environment

os.environ()

Returns the current working directory.

os.getcwd()

Return the real group id of the current process.

os.getgid()

Return the current process's user id.

os.getuid()

Returns the real process ID of the current process.

os.getpid()

Set the current numeric umask and return the previous umask.

os.umask(mask)

Return information identifying the current operating system.

os.uname()

Change the root directory of the current process to path.

os.chroot(path)

Return a list of the entries in the directory given by path.

os.listdir(path)

Create a directory named path with numeric mode mode.

os.mkdir(path)

Recursive directory creation function.

os.makedirs(path)

Remove (delete) the file path.

os.remove(path)

Remove directories recursively.

os.removedirs(path)

Rename the file or directory src to dst.

os.rename(src, dst)

Remove (delete) the directory path.

os.rmdir(path)

4.4.2 Sys Module:

The sys module provides information about constants, functions and methods of the Python interpreter. dir(system) gives a summary of the available constants, functions and methods. Another possibility is the help() function. Using help(sys) provides valuable detail information.

The module sys informs e.g. about the maximal recursion depth (sys.getrecursionlimit()) and provides the possibility to change (sys.setrecursionlimit()).

The current version number of Python can be accessed as well by using this module.

Lots of scripts need access to the arguments passed to the script, when the script was started. argvargv (or to be precise sys.argv) is a list, which contains the command-line arguments passed to the script. The first item of this list contains the name of the script itself. The arguments follow the script name.

Every serious user of a UNIX or Linux operating system knows standard streams, i.e. input, standard output and standard error. They are known as pipes. They are commonly abbreviated as stdin, stdout, stderr.

The standard input (stdin) is normally connected to the keyboard, while the standard error and standard output go to the terminal (or window) in which you are working.

These data streams can be accessed from Python via the objects of the sys module with the same names, i.e. sys.stdin, sys.stdout and sys.stderr.

The standard output (stdout) can be redirected e.g. into a file, so that we can process this file later with another program. The same is possible with the standard error stream, we can redirect it into a file as well. We can redirect both stderr and stdout into the same file or into separate files.

4.4.3 Numpy:

NumPy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called axes.

NumPy is module for Python. The name is an acronym for "Numeric Python" or "Numerical Python". It is an extension module for Python, mostly written in C. This makes sure that the precompiled mathematical and numerical functions and functionalities of Numpy guarantee great execution speed.

Furthermore, NumPy enriches the programming language Python with powerful data structures, implementing multi-dimensional arrays and matrices. These data structures guarantee efficient calculations with matrices and arrays. The implementation is even aiming at huge matrices and arrays, better known under the heading of "big data". Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays.

SciPy (Scientific Python) is often mentioned in the same breath with NumPy. SciPy needs Numpy, as it is based on the data structures of Numpy and furthermore its basic creation and manipulation functions. It extends the capabilities of NumPy with further useful functions for minimization, regression, Fourier-transformation and many others.

Both NumPy and SciPy are not part of a basic Python installation. They have to be installed after the Python installation. NumPy has to be installed before installing SciPy.

NumPy is based on two earlier Python modules dealing with arrays. One of these is Numeric. Numeric is like NumPy a Python module for high-performance, numeric computing, but it is obsolete nowadays. Another predecessor of NumPy is Numarray, which is a complete rewrite of Numeric but is deprecated as well. NumPy is a merger of those two, i.e. it is built on the code of Numeric and the features of Numarray.

NumPy's array class is called ndarray. It is also known by the alias array. Note that numpy.array is not the same as the Standard Python Library class array.array, which only handles one-dimensional arrays and offers less functionality. The more important attributes of an ndarray object are:

ndarray.ndim

The number of axes (dimensions) of the array.

ndarray.shape

The dimensions of the array. This is a tuple of integers indicating the size of the array in each dimension. For a matrix with n rows and m columns, shape will be (n,m). The length of the shape tuple is therefore the number of axes, ndim.

ndarray.size

The total number of elements of the array. This is equal to the product of the elements of shape.

ndarray.dtype

An object describing the type of the elements in the array. One can create or specify dtype's using standard Python types. Additionally NumPy provides types of its own. numpy.int32, numpy.int16, and numpy.float64 are some examples.

ndarray.itemsize

The size in bytes of each element of the array. For example, an array of elements of type float64 has itemsize 8 (=64/8), while one of type complex32 has itemsize 4 (=32/8). It is equivalent to ndarray.dtype.itemsize.

CHAPTER 5

TESTING

5.1 SOFTWARE TESTING:

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

5.1.1 Basics of Software Testing :

There are two basics of software testing: Black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

5.1.2 Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Regression Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Regression Testing

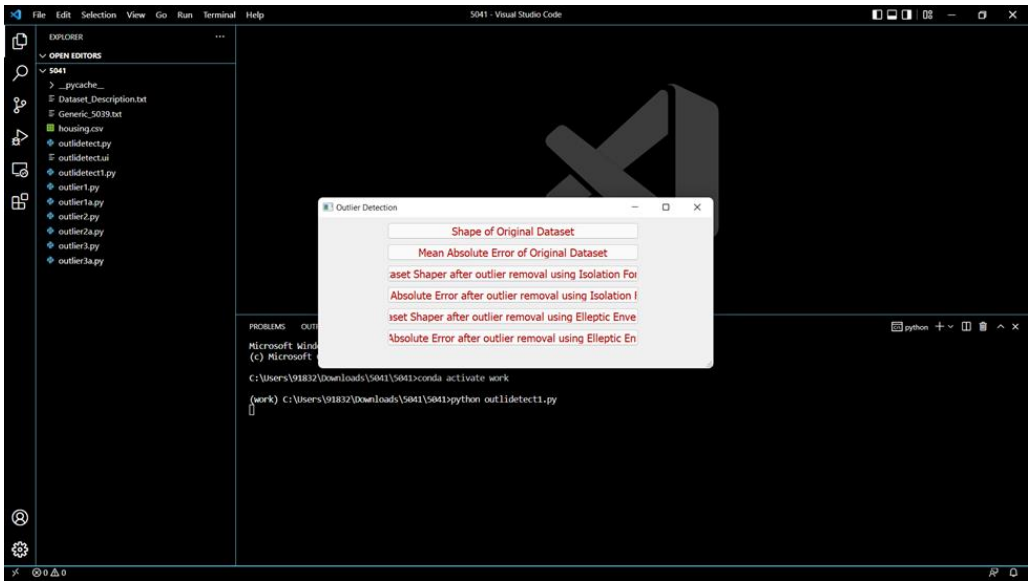
Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

CHAPTER 6

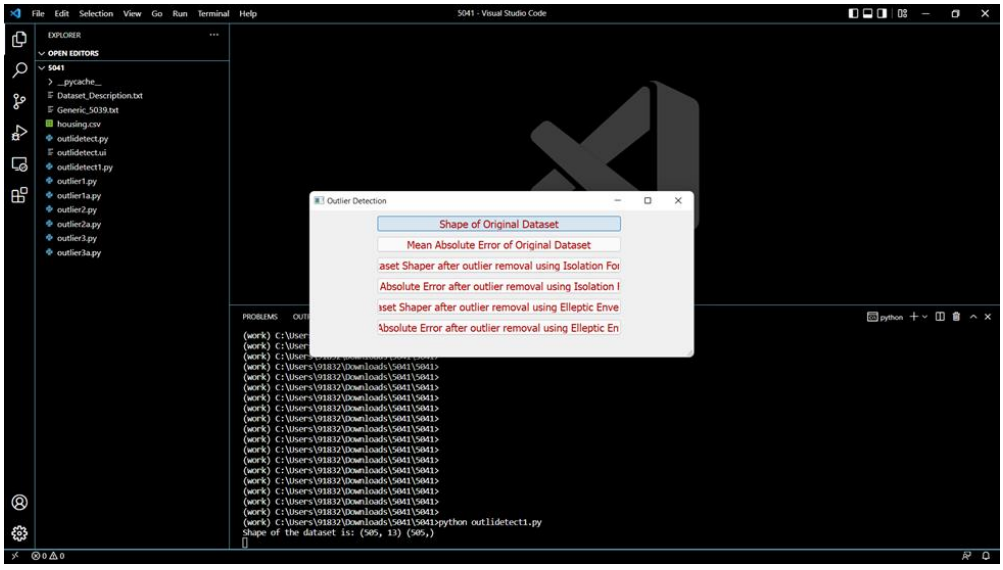
RESULTS

6.1 TESTING RESULTS:

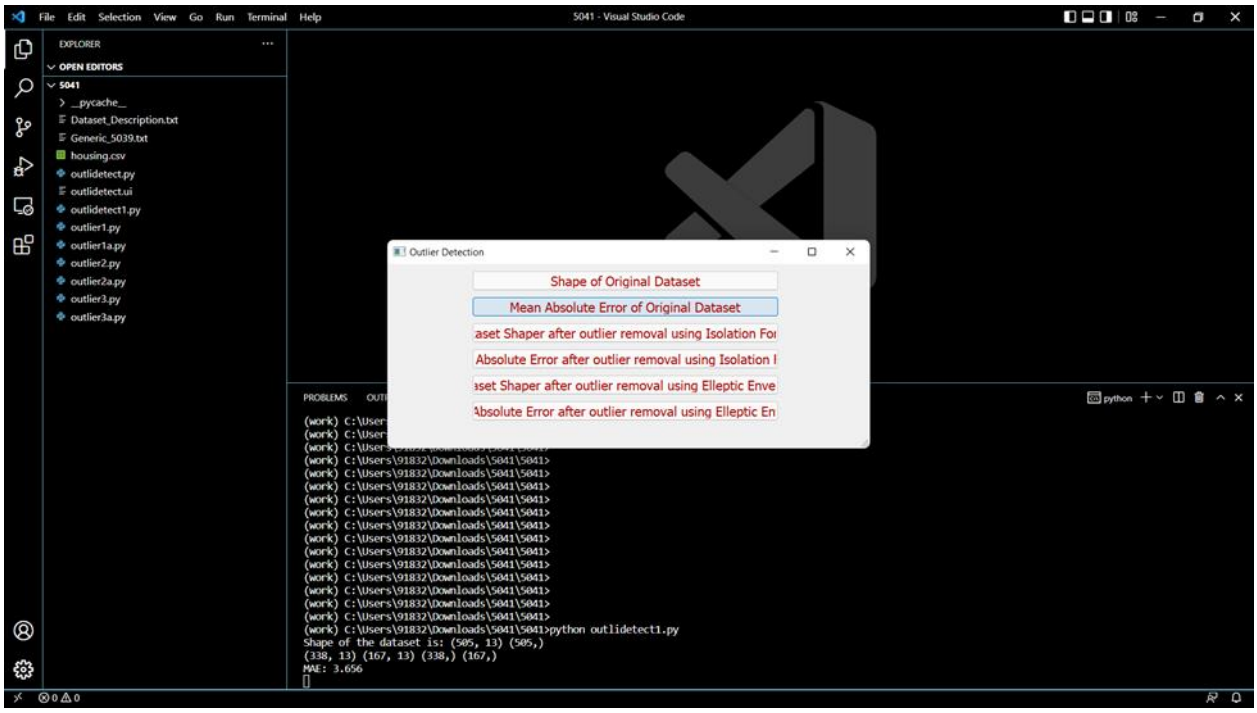
6.1.1 Outlidetect1.py



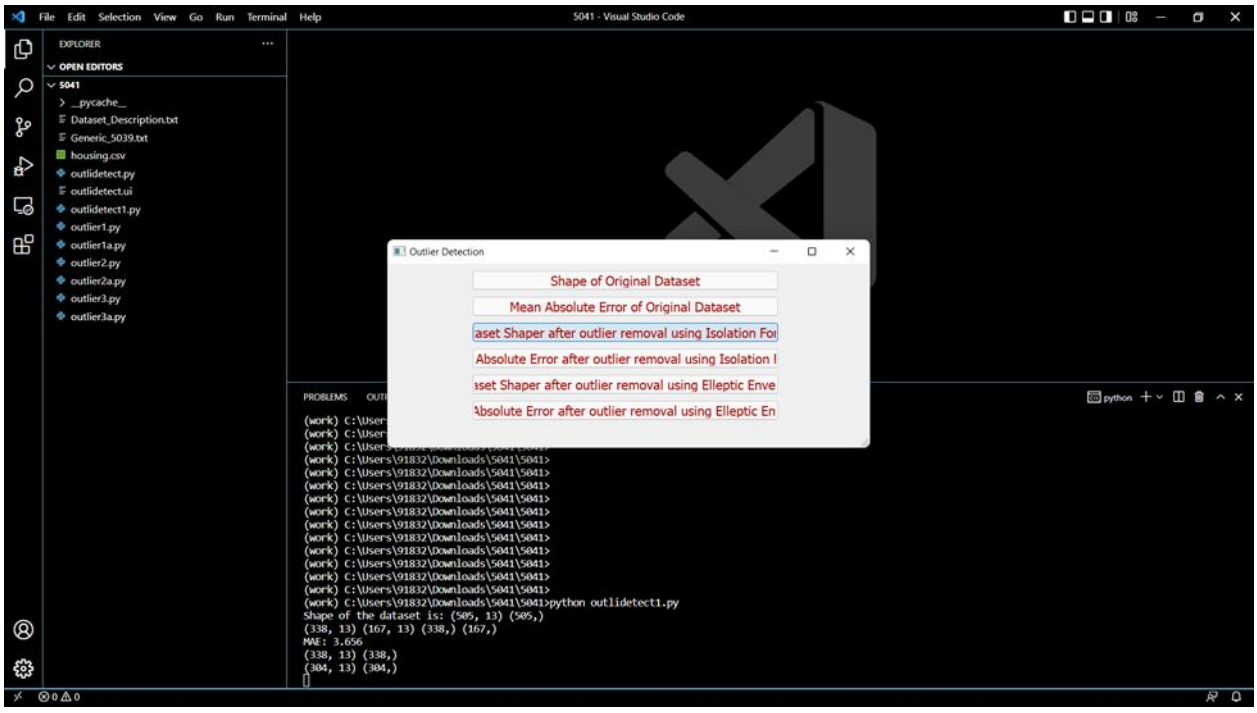
6.1.2 Outlier1.py



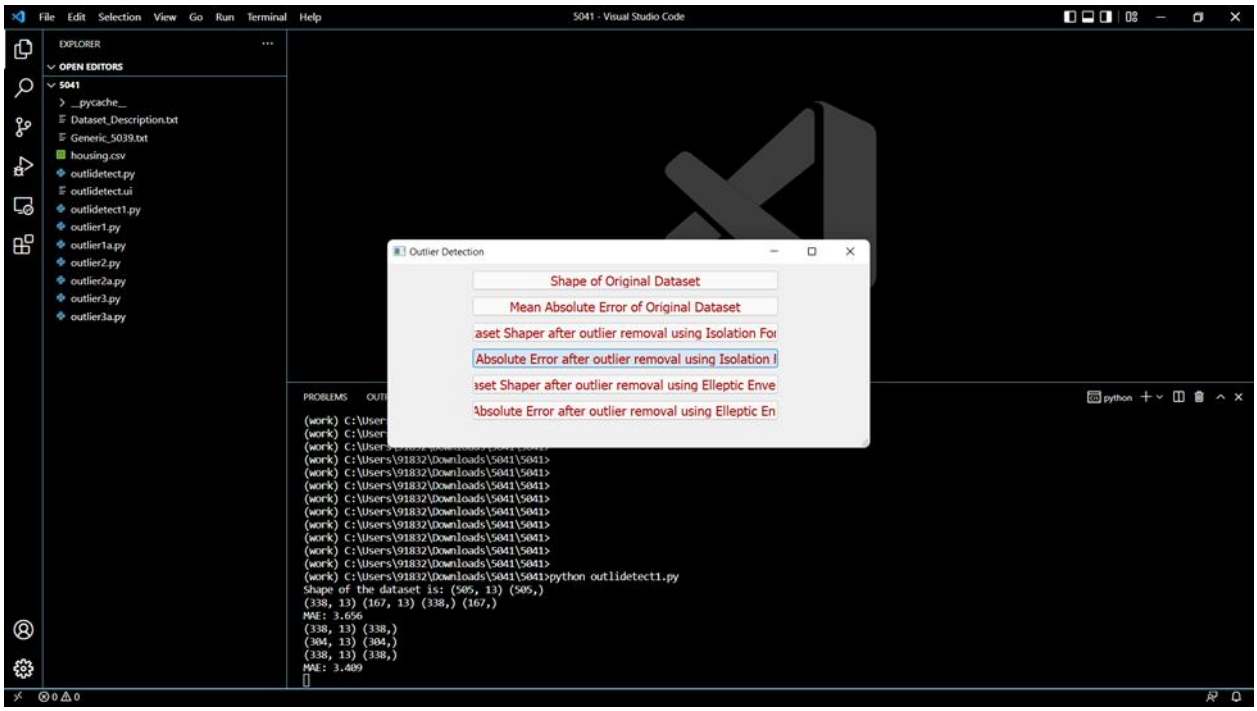
6.1.3 Outlier1a.py



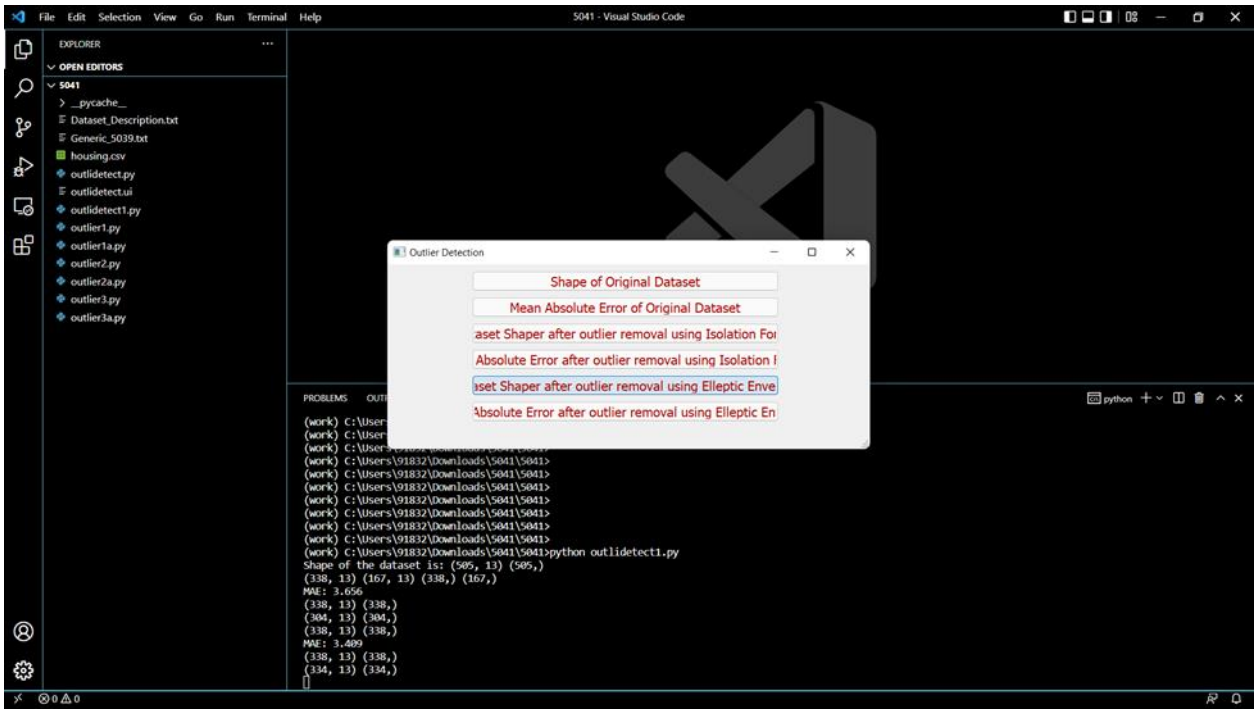
6.1.4 Outlier2.py



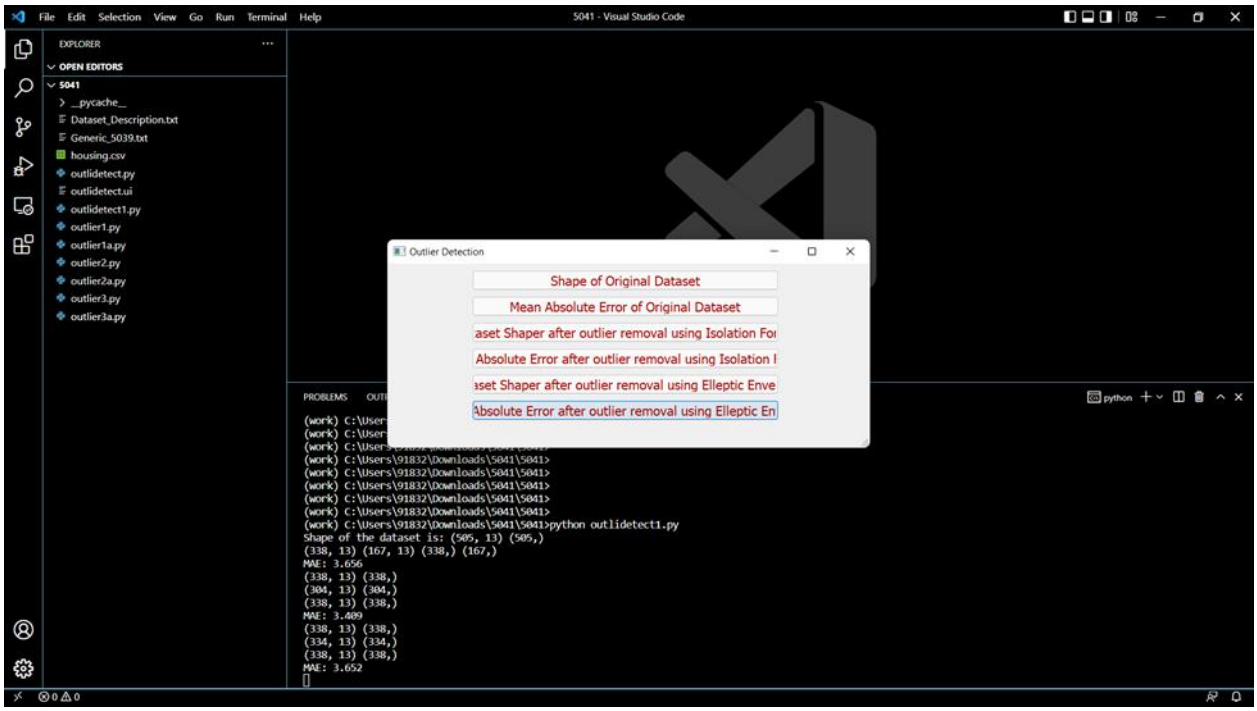
6.1.5 Outlier2a.py



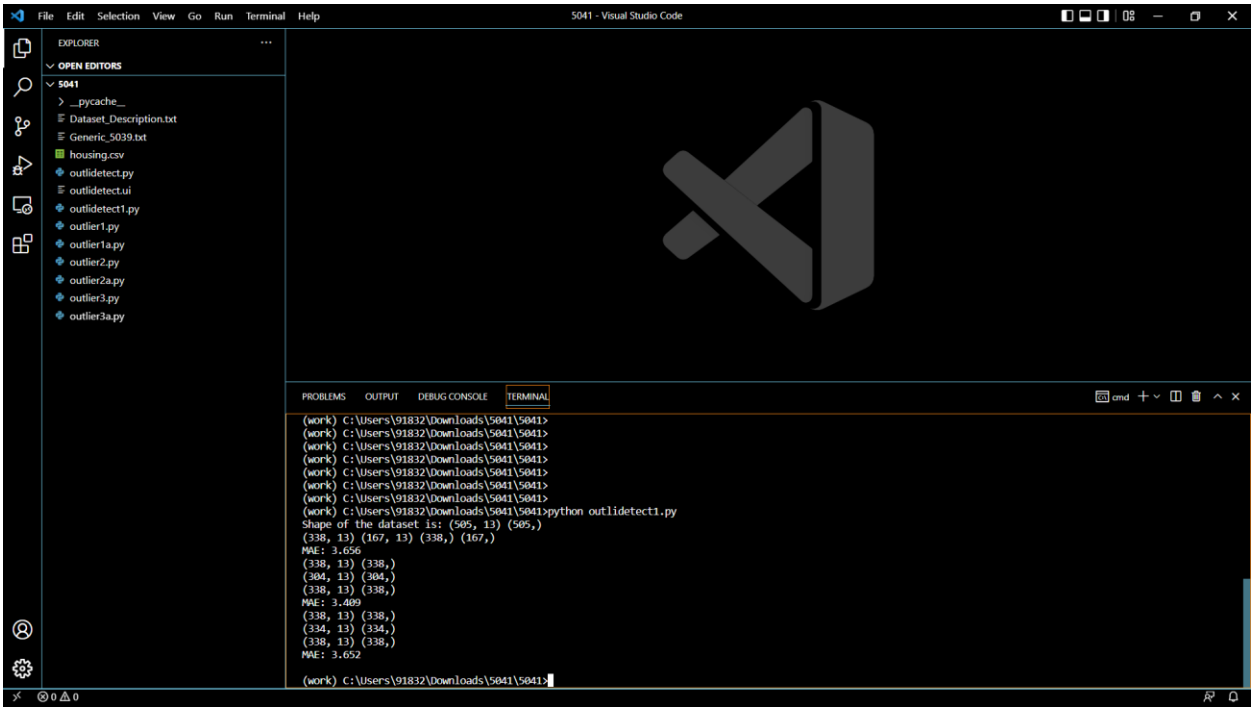
6.1.6 Outlier3.py



6.1.7 Outlier3a.py



6.1.8 Final Output



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

This project entitled **"Dataset Improvement using Outlier Removal."** is uesful to get accurate data analysis results by removing the outliers from the analysis process.

The project is useful to the data analysts to understand more about isolation forest and elleptic envelope methods.

The project is useful to reduce the mean absolute error in the data by removing the outliers.

7.2 FUTURE SCOPE:

As of now, the project is implemeted by using the dataset of real estate sector. The applicability of this project to the other sectors need to be further explored by using the datasets of corresponding sectors.

CHAPTER 8

BIBLIOGRAPHY

[1] YANG MA AND XUJUN ZHAO:POD: A Parallel Outlier Detection Algorithm Using Weighted kNN, IEEE Access

[2] S. Ramaswamy, R. Rastogi, and K. Shim, ``Effcient algorithms for mining outliers from large data sets," in Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD), New York, NY, USA, pp. 427-438**.**

[3] F. Angiulli and C. Pizzuti, ``Outlier mining in large high-dimensional data sets," IEEE Trans. Knowl. Data Eng., vol. 17, no. 2, pp. 203-215.

[4] https://www.python.org/

[5]https://github.com/baoboa/pyqt5/blob/master/pyuic/uic/pyuic.py

[6] https://www.numpy.org/

[7] https://riverbankcomputing.com/software/pyqt/intro