**Visvesvaraya Technological University**

**Jnana Sangama, Belagavi – 590018, Karnataka**



**Python Application Programming**

# **A mini project report on**

### Emotion Predictor & Therapist

**Submitted By:**

**Rai Sakshi Rajkumar 1GA17CS115**

**Aneeq Ul Waseeq 1GA17CS014**

**Under the Guidance of**

**Sushmitha R**

**Faculty Dept. Of CSE**



**Department of Computer Science and Engineering**

**(Accredited by NBA 2019-2022)**

**Global Academy of Technology**

**Rajarajeshwarinagar, Bangalore – 560098**

**2019-2020**

**Global Academy of Technology**

**Department of Computer Science and Engineering**



Certificate

This is to certify that the project entitled **”Emotion Predictor & Therapist”** is a bonafide work carried out by **Rai Sakshi Rajkumar(1GA17CS115), Aneeq Ul Waseeq(1GA17CS014)** as a part of assignment in Python Applications Programming in Computer Science and Engineering during the year 2019-2020.

|  |  |  |
| --- | --- | --- |
| **Max. marks** | **Marks obtained** | **Faculty Name and Signature** |
| **07** |  |  |

### ABSTRACT

This app is all about predicting a user's emotion based on the feelings user expresses to the app. According to the emotion experienced by the user, the app tries to help them with their emotions by displaying relative and helpful content. To predict the emotions of the user we used natural language processing (NLP) and trained our predictor model using ISEAR data set. Our model can predict up to seven human emotions, which gives us a better prediction and understanding of emotions.

**TABLE OF CONTENTS**

**TOPIC PAGE NO**

**1. INTRODUCTION**

1.1 Python Programming Language

1.2 Applications of Python

**2. SYSTEM REQUIREMENTS**

2.1 Software Requirements

2.2 Hardware Requirements

**3. IMPLEMENTATION AND RESULTS**

3.1 Project Code

3.2. Results

**CONCLUSION**

**REFERENCES**

**ORGANIZATION OF THE REPORT**

The report is divided into various chapters and is organized as follows:

**Chapter 1: Introduction**

This chapter includes brief introduction to Python Programming Language and its applications.

**Chapter 2: System requirements**

This chapter includes details of hardware and software requirements necessary for the execution of the project.

**Chapter 3: Implementation and Results**

This chapter includes the program code of the project and the results of successful runs of the code.

**Conclusion**

This section includes the conclusion about the project.

**References**

This section includes the bibliographical references used for the development of the project.

**CHAPTER 1**

**INTRODUCTION**

* 1. **Python Programming language**

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code as compared to other programming languages. It provides very high-level dynamic data types and supports dynamic type checking.

The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative and functional programming or procedural styles. Python is dynamically typed and garbage collected.

Python was conceived in the late 1980s, and its implementation was started in December 1989 by Guido Van Rossum at CWI in the Netherlands as a successor to the ABC language (itself inspired by SETL) capable of exception handling and interfacing with the Amoeba operating system.

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).

Python 2.0, released in 2000, introduced features like [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a garbage collection system capable of collecting [reference cycles](https://en.wikipedia.org/wiki/Reference_cycle). Python 3.0, released in 2008, was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility), and much Python 2 code does not run unmodified on Python 3.

An interesting fact about python is that it is named after the comedy television show Monty Python’s Flying Circus. It is not named after the Python snake.

* 1. **Applications of Python programming language**

Python supports cross-platform operating systems which makes building applications with it very convenient. Some of the globally known applications such as YouTube, BitTorrent, DropBox etc use Python to achieve their functionality.

Here are few applications of Python:

1. Web Development: Python can be used to make web applications at rapid rate because of the frameworks provided by Python. Some of the most well-known frameworks are Django, Flask, Pyramid. These frameworks provide better scalability and security.
2. Game Development: Python is also used in the development of interactive games. There are libraries such as PySoy which is a 3D game engine supporting Python 3. PyGame which provides functionality and a library for Game Development. Games such as Civilization-IV, Disney’s Toontown Online, Vega Strike etc. have been using Python.
3. Machine Learning and Artificial Intelligence: These fields are the most trending and also most promising careers for the future. It is basically training computers using algorithms. Libraries used for it are Pandas, Scikit-learn, NumPy etc.
4. Data Science and Data Visualization: Data Science involves analyzing data, performing operations and extracting calculated risks, increase in profits and other relevant information. NumPy, Pandas libraries can be used. For data visualization that is plotting related graphs, libraries such as Matplotlib, Seaborn etc.
5. Desktop GUI: Python can be used to program desktop applications. It provides Tkinter library that can be used to develop User Interfaces. Some other toolkits like wxWidgets, Kivy, PYQT can be used to create applications on several platforms
6. Web Scraping Applications: Python can be used to pull a large amount of data from websites which can be helpful in various real-world processes such as price comparison, job listings, research and development. BeautifulSoup is the library used to pull such data and use accordingly

**CHAPTER 2**

**SYSTEM REQUIREMENTS**

**2.1 Software Requirements**

Operating System: Windows 10 or any compatible operating system

Tools used: PyCharms IDE

**2.2 Hardware Requirements**

Processor: Any Processor above 500 MHz

RAM: 4GB

Hard Disk: 2GB free space

Input device: Keyboard, Mouse

Output device: Monitor

**CHAPTER 3**

**IMPLEMENTATION AND RESULTS**

3**.1 About Project**

‘Emotion Predictor and Therapist’ takes text as user input and predicts what you are feeling. Natural Language Processing is used to process the text data provided by the user and then classify it into 7 different emotion categories Joy, Sadness, Disgust, Anger, Shame and Guilt. Multinomial Naïve Bayes Algorithm is used for classification. Our Model is trained using ISEAR dataset and accuracy score is approximately 89% (highest among all the other classification algorithms that were tried like SVC and Random Forest). Based on this prediction, it provides users with content which can help them deal with what they’re feeling.

We've used Flask to deploy our model onto web. Flask is a lightweight WSGI (Web Server Gateway Interface- a simple calling convention for web servers to forward requests to web applications), basically a micro web framework written in python.

HTML (Hyper Text Markup Language) and CSS is used for front-end development.

**3.2 Project code**

**Python code**

#importing neccessary lib

#used flask for server

import numpy as np

from flask import Flask, request, jsonify, render\_template

import pickle

from sklearn.externals import joblib

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.naive\_bayes import MultinomialNB

app = Flask(\_\_name\_\_)

#redirects to the main page when app is started

@app.route('/') def home():

return render\_template('index.html')

#this function predicts the emotion based on input

@app.route('/predict',methods=['POST'])

def predict():

data = pd.read\_csv("ISEAR2.csv")

#95% data is used to train the model

def simple\_split(data, y, length, split\_mark=0.95):

if split\_mark > 0. and split\_mark < 1.0:

n = int(split\_mark \* length)

else:

n = int(split\_mark)

xtrain = data[:n].copy()

xtest = data[n:].copy()

ytrain = y[:n].copy()

ytest = y[n:].copy()

return xtrain, xtest, ytrain, ytest

vectorizer = CountVectorizer()

xtrain, xtest, ytrain, ytest = simple\_split(data.text, data.emotion,len(data))

xtrain = vectorizer.fit\_transform(xtrain)

xtest = vectorizer.transform(xtest)

mnb = MultinomialNB()

mnb.fit(xtrain, ytrain)

if request.method == 'POST':

emotion = request.form['emotion']

data1 = [emotion]

vect = vectorizer.transform(data1)

#here the predicted emotion is stored

my\_prediction = mnb.predict(vect)

#these if statements redirect user to web pages according to their predicted emotions.

if my\_prediction == 'sadness':

return render\_template('sadness.html', prediction=my\_prediction)

if my\_prediction == 'joy':

return render\_template('joy.html', prediction=my\_prediction)

if my\_prediction == 'anger':

return render\_template('anger.html', prediction=my\_prediction)

if my\_prediction == 'disgust':

return render\_template('anger.html', prediction=my\_prediction)

if my\_prediction == 'fear':

return render\_template('fear.html', prediction=my\_prediction)

if my\_prediction == 'guilt':

return render\_template('shame.html', prediction=my\_prediction)

if my\_prediction == 'shame':

return render\_template('shame.html', prediction=my\_prediction)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

import numpy as np #importing neccessary lib  
from flask import Flask, request, jsonify, render\_template #used flask for server  
import pickle  
from sklearn.externals import joblib  
app = Flask(\_\_name\_\_)  
@app.route('/') #redirects to the main page when app is started  
def home():  
 return render\_template('index.html')  
  
@app.route('/predict',methods=['POST']) #this function predicts the emotion based on input  
def predict():  
 import pandas as pd  
 from sklearn.feature\_extraction.text import CountVectorizer  
 from sklearn.naive\_bayes import MultinomialNB  
 data = pd.read\_csv("ISEAR2.csv") #training dataset is loaded  
 def simple\_split(data, y, length, split\_mark=0.95): #95% data is used to train the model  
 if split\_mark > 0. and split\_mark < 1.0:  
 n = int(split\_mark \* length)  
 else:  
 n = int(split\_mark)  
 xtrain = data[:n].copy()  
 xtest = data[n:].copy()  
 ytrain = y[:n].copy()  
 ytest = y[n:].copy()  
 return xtrain, xtest, ytrain, ytest  
  
 vectorizer = CountVectorizer()  
 xtrain, xtest, ytrain, ytest = simple\_split(data.text, data.emotion, len(data))  
 xtrain = vectorizer.fit\_transform(xtrain)  
 xtest = vectorizer.transform(xtest)  
 mnb = MultinomialNB()  
 mnb.fit(xtrain, ytrain)  
 if request.method == 'POST':  
 emotion = request.form['emotion']  
 data1 = [emotion]  
 vect = vectorizer.transform(data1)  
 my\_prediction = mnb.predict(vect) #here the predicted emotion is stored  
 if my\_prediction == 'sadness': #these if statements redirect user to web pages according to their predicted emotions.  
 return render\_template('sadness.html', prediction=my\_prediction)  
 if my\_prediction == 'joy':  
 return render\_template('joy.html', prediction=my\_prediction)  
 if my\_prediction == 'anger':  
 return render\_template('anger.html', prediction=my\_prediction)  
 if my\_prediction == 'disgust':  
 return render\_template('anger.html', prediction=my\_prediction)  
 if my\_prediction == 'fear':  
 return render\_template('fear.html', prediction=my\_prediction)  
 if my\_prediction == 'guilt':  
 return render\_template('shame.html', prediction=my\_prediction)  
 if my\_prediction == 'shame':  
 return render\_template('shame.html', prediction=my\_prediction)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

**index.html**

<html>  
<head>  
<title>Student form</title>  
<style type="text/css">  
body {  
margin: 0;  
background :url('http://www.gat.ac.in/wp-content/uploads/2016/11/3-2.jpg');  
background-size:cover;  
}  
h1 {  
font-family: cursive;  
background-color: #000;  
color:white;  
text-align: center;  
padding: 20px;  
}  
h3 {  
font-family: cursive;  
background-color: #000;  
color:white;  
text-align: center;  
padding: 20px;  
}  
div {  
width: 400px;  
margin: 10px auto;  
box-shadow: 0px 1px 10px #404040;  
background: rgba(255, 255, 255, 0.2);  
border-color:double;  
border-size:100px;  
border-radius: 15px 50px;  
padding-bottom: 10px;  
}  
a {  
padding: 30px;  
margin: 30px;  
border-radius: 10px;  
background-color: black;  
text-decoration: none;  
display: inline-block;  
color: #fff;  
font-family: cursive;  
}  
a:hover {  
background-color: #cccccc;  
transition: all 0.3s  
}  
input[type=Submit]{  
background-color:black ;  
border: none;  
color: white;  
padding: 16px 32px;  
text-decoration: none;  
margin: 4px 2px;  
cursor: pointer;  
font-family: cursive;  
}  
</style>  
</head>  
<body>  
<form action="{{ url\_for('predict')}}" method="post">  
<h1>EMOTION PREDICTOR & THERAPIST</h1>  
<h3>This app helps you with your feelings, you can express yourself and it helps you to be yourself.</h3>  
<div align="center">  
<a>Tell us what's going on your mind?<input type="text" name ="emotion"></a></br>  
<input type="Submit" value="Go">  
 <div>{{ prediction }}</div>  
</div>  
</form>  
</div>  
</body>  
</html>

**Joy.html**

<html>  
<head>  
 <title>Student form</title>  
 <style type="text/css">  
 body {  
 margin: 0;  
 background: linear-gradient(rgba(255, 255, 255, 0.3), rgba(255, 255, 255, 0.3));  
 background-size: cover;  
 }  
  
 p {  
 font-family: cursive;  
 color: black;  
 text-align: center;  
 padding: 2px;  
 font-weight: bold;  
 font-size: 25px;  
 }  
  
 h1 {  
 font-family: cursive;  
 background-color: #000;  
 color: white;  
 text-align: center;  
 padding: 20px;  
 }  
  
 div {  
 width: 400px;  
 margin: 10px auto;  
 }  
  
 a {  
 padding: 30px;  
 margin: 30px;  
 border-radius: 10px;  
 background-color: black;  
 text-decoration: none;  
 display: inline-block;  
 color: #fff;  
 font-family: cursive;  
 }  
  
 a:hover {  
 background-color: #cccccc;  
 transition: all 0.3s  
 }  
  
 input[type=Submit] {  
 background-color: black;  
 border: none;  
 color: white;  
 padding: 16px 32px;  
 text-decoration: none;  
 margin: 4px 2px;  
 cursor: pointer;  
 font-family: cursive;  
 }  
  
 td {  
 padding: 10px;  
  
 }  
 </style>  
</head>  
<body>  
 <form action="{{ url\_for('predict')}}" method="post">  
 <h1>EMOTION PREDICTOR & THERAPIST</h1>  
  
 </form></br></br>  
 <table align="center" border=2>  
 <div align="center">  
 <p>You are feeling :{{ prediction }}</p>  
 </div>  
 </table>  
 <p>Here have a look at these:</p>  
  
 <table align="center">  
 <tr>  
 <td><img src="https://messages.365greetings.com/wp-content/uploads/2018/03/feel-good-quotes-01.jpg  
"  
 width="300" height="240"></td>  
 <td><img src="https://winkgo.com/wp-content/uploads/2019/03/happy-memes-make-you-smile-more-31-720x680.jpg"  
 width="300" height="240"></td>  
 <td><img src="https://images-na.ssl-images-amazon.com/images/I/81sPhCNa79L.jpg"  
 width="300" height="240"></td>  
 </tr>  
 </table>  
 <table align="center">  
 <tr>  
 <td><img src="https://i.pinimg.com/736x/a0/69/2c/a0692c97b88872640a433e6fad5ecd45.jpg"  
 width="300" height="240"></td>  
  
 <td><img src="https://aknextphase.com/wp-content/uploads/2017/11/Share-Your-Joy.jpg"  
 width="300" height="240"></td>  
 </tr>  
 </table>  
 <table align="center">  
 <tr>  
  
 <td><img src="https://image.freepik.com/free-vector/cute-rabbit-bunny-lover-couple-greeting-cartoon-doodle-wallpaper\_40564-338.jpg  
"  
 width="300" height="240"></td>  
 </tr>  
 </table>  
  
 <div>  
 <p>You can also have a look at these following videos:</p>  
 <p><a href="https://www.youtube.com/watch?v=71hqRT9U0wg">If You're Happy and you know it!!!</a></p>  
 <p><a href="https://www.youtube.com/watch?v=8KkKuTCFvzI">Here's something inspiring!!</a></p>  
  
  
 </div>  
 <p><a href="/">Let's see how you are feeling now.</a></p>  
 </div>  
</body>  
  
</html>

**3.3 Results**

Add snapshots taken from successful runs of the project(min. 4 and max. 6 snapshots).

Every snapshot to be labeled as follows

**Snapshot 3.1: \_\_\_\_\_\_\_**

**Snapshot 3.2: \_\_\_\_\_\_\_**

Snapshot label should be bold and center aligned.

Font: Times New Roman, 10pt.

**CONCLUSION**

We learned about many concepts and application of python language namely, NLP using NLTK library, Tensorflow library for deep learning, web development for front end. Generating model to train our data set using AI. Time management is one of the main skills we used to plan and develop our project. We designated team members with deadlines for each module to be developed. This helped us to complete the project on time.

**REFERENCES**

**Bibliography:**

* 1. Charles R. Severance, “Python for Everybody: Exploring Data Using Python 3”, 1stEdition, CreateSpace Independent Publishing Platform, 2016.
  2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist”,2nd Edition, Green Tea Press, 2015.
  3. Charles Dierbach, "Introduction to Computer Science Using Python", 1st Edition,Wiley India Pvt Ltd. ISBN-13: 978-8126556014

**Websites:**

* 1. www.medium.com
  2. www.towardsdatascience.com
  3. www.kaggle.com