Major Project Report

A Dissertation submitted to the RGUKT-AP in partial fulfillment of the degree of
Bachelor of Technology
in
Computer Science
By

SUNKARA BHAVANA REDDY (R170909)

Under the guidance of :
Ms.HIMA BINDHU
Assistant professor
Computer Science Department

Rajiv Gandhi University of Knowledge Technologies AP-IIIT, Rk Valley, Idupulapaya, Kadapa - 516 330 Andhra Pradesh, India

CERTIFICATE

This is to certify that the dissertation of Major project Report entitled "Nonverbal Communication" submitted by SUNKARA BHAVANA REDDY, bearing Id.No.R170909, in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science is a bonafide work carried out by her under my supervision and guidance.

The dissertation has not been submitted previously in part or infull to this or any other University or Institution for the award of any degree or diploma.

Mr Satyanandram,
Assistant Professor,
Head of the Department,
Computer Science Dept,
RGUKT, RK VALLEY.

Ms. Hima Bindu
Project Internal Guide
Assistant Professor
Computer Science Dept,
RGUKT, RK VALLEY

DECLARATION

I SUNKARA BHAVANA REDDY hereby declare that this Dissertation of Major Project Report entitled "Nonverbal Communication" submitted by me under the guidance and supervision of Asst Prof. Ms.Hima Bindu is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

Date: -

Place: - RGUKT, RK VALLEY.

S Bhavana Reddy

(R170909)

Nonverbal communication

ABSTRACT

Nonverbal communication shows differences between cultures which occur because of how different people around the world interpret actions in social interaction. Understanding the cultural differences in nonverbal communication is important for those with a goal to work in international business.

In every country there are some common non verbal communication which are used widely which help people while travelling to other countries for work or for tourism.

Therefore it is used to interpret what people are communicating.

Non verbal communication gestures are interpreted using machine learning which can help people while travelling to other countries.

INDEX

Sl.no	Contents	Page No.
1	Introduction	4
2	Machine learning	4
3	Data visualization	7
3.1	Types	7
3.2	Tools	9
3.3	Techniques	10
4	Datasets	10
4.1	Column data	10
4.2	Features	12
5	Libraries	12
5.1	Matplotlib	12
5.2	Seaborn	13
5.3	Tensorflow	13
5.4	Keras	13
6	Methodology	13
6.1	One hot encoding	13
6.2	Heatmap	14
7	Implementation	14
8	Conclusion	25

1. INTRODUCTION

Machine learning plays an important role in computer science and artificial intelligence.
The field of nonverbal communication (NVC) has a long history involving many cue modalities, including face, voice, body, touch, and interpersonal space; different levels of analysis, including normative, group, and individual differences; and many substantive themes that cross from psychology into other disciplines.
The NVC field is advancing rapidly. Technological advances such as automatic measurement, brain imaging, and affective computing offer new possibilities for research. In addition, due to the calls for more measurement of actual social behavior, as opposed to self-reports and measures of nonsocial behaviors such as reaction times there is renewed interest in NVC as a compelling behavioral window into psychological processes.

2. Machine Learning

Machine learning is a subset of AI, which enables the machine to automatically learn from data, improve performance from past experiences, and make predictions. Machine learning contains a set of algorithms that work on a huge amount of data. Data is fed to these algorithms to train them, and on the basis of training, they build the model & perform a specific task.
machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning

2.1 Supervised Machine Learning

As its name suggests, supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the

training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

Supervised machine learning can be classified into two types of problems, which are given below:

- Classification
- Regression

2.2 Unsupervised Machine Learning

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.

Unsupervised Learning can be further classified into two types, which are given below:

- Clustering
- Association

2.3 Semi-Supervised Learning

Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate ground between Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabeled datasets during the training period.

2.4 Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded

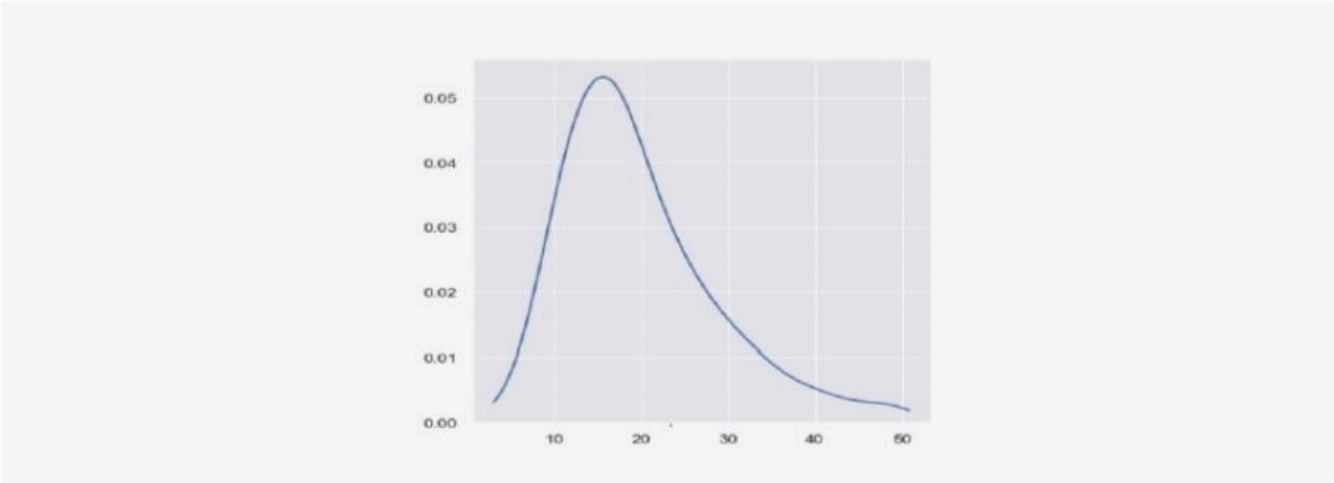
for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

3. Data visualization

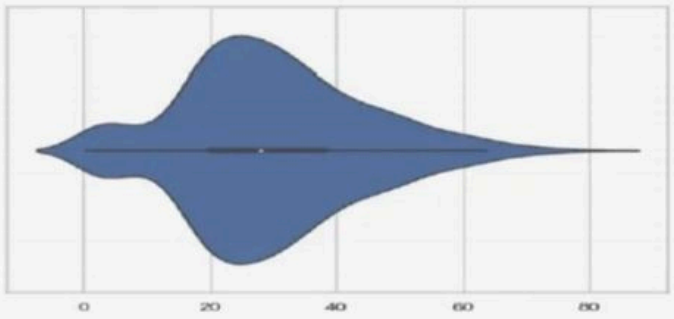
Data Visualization is the process of generating graphical representations of data for various purposes. These graphical representations are commonly known as plots or charts in data science terminology. Data visualization benefits include communicating your results or findings, monitoring the model's performance at the evaluation stage, hyperparameter tuning, identifying trends, patterns and correlation between dataset features, data cleaning such as outlier detection, and validating model assumptions.

3.1 Types

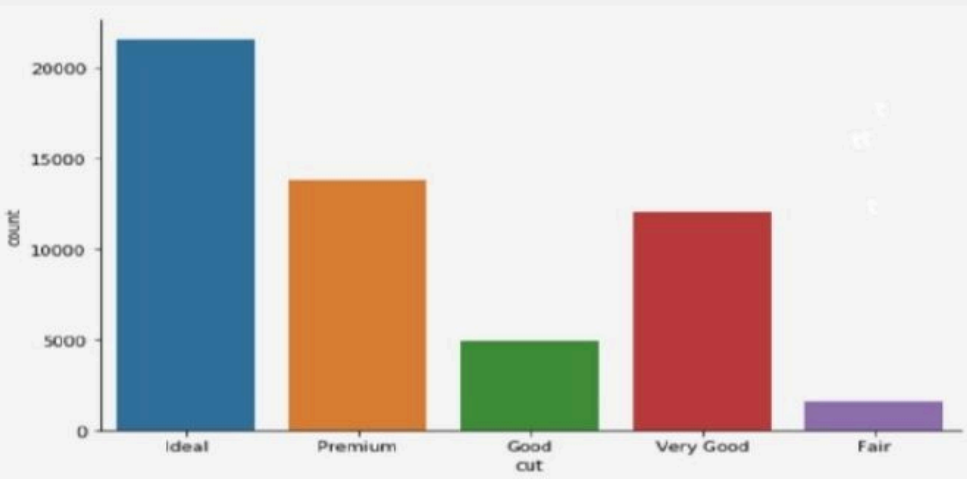
a)Distribution plot: A distribution plot is used to visualize data distribution. Example: Probability distribution plot or density curve.



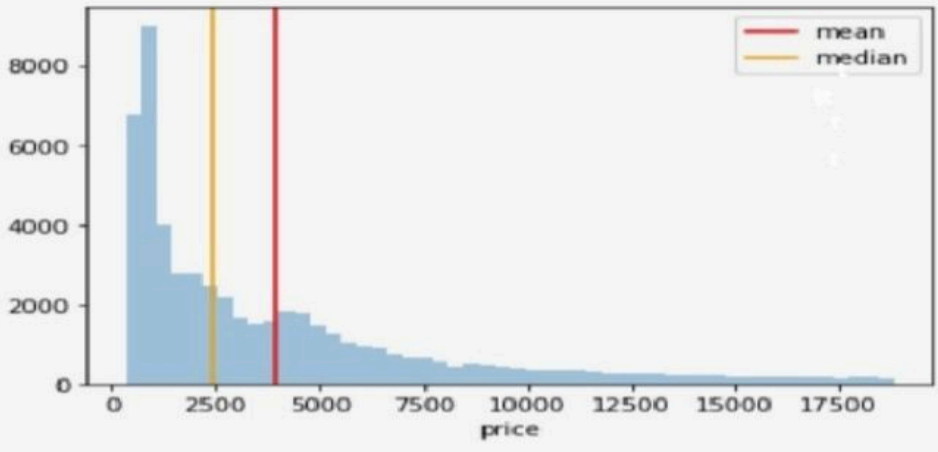
b)Violin plot: Similar to the box and whisker plot, the violin plot is used to plot the variation of a numerical feature. But it contains a kernel density curve in addition to the box plot. The kernel density curve estimates the underlying distribution of data.



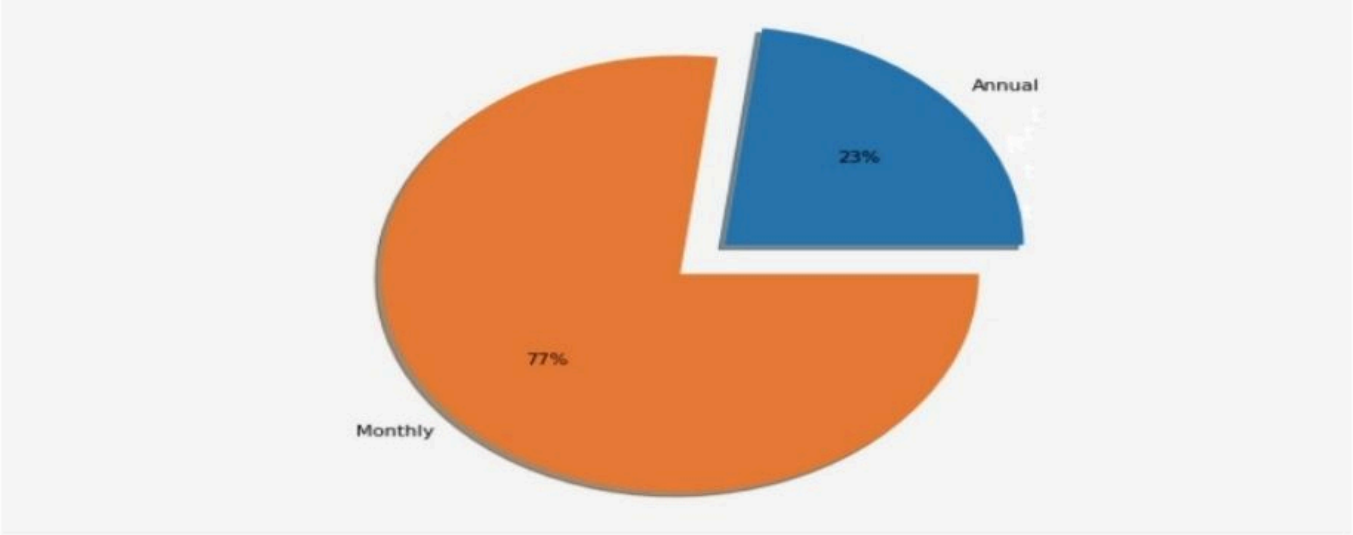
c) Bar plot: A bar plot is used to plot the frequency of occurring categorical data. Each category is represented by a bar. The bars can be created vertically or horizontally. Their heights or lengths are proportional to the values they represent



d)Histogram: A histogram represents the distribution of numerical data. Looking at a histogram, we can decide whether the values are normally distributed (a bell-shaped curve), skewed to the right or skewed left. A histogram of residuals is useful to validate important assumptions in regression analysis.



e)Pie chart: A categorical variable pie chart includes each category's values as slices whose sizes are proportional to the quantity they represent. It is a circular graph made with slices equal to the number of categories.



3.2 Tools and Software for Data Visualization

a) Python provides open-source libraries such as

- Matplotlib
- Seaborn
- Plotty
- Bokeh
- Altair

b) R provides open-source libraries such as

- Ggplot2
- Lattice

c) Other data visualization libraries

- IBM SPSS
- Minitab
- Matlab for data visualization
- Tableau
- Microsoft Power BI are popular among data scientists.

3.3 Techniques

a) Univariate Analysis

In univariate analysis, as the name suggest, we analyze only one variable at a time. In other words, we analyze each variable separately. Bar charts, pie charts, box plots and histograms are common examples of univariate data visualization. Bar charts and pie charts are created for categorical variables, while box plots and histograms are created for numerical variables.

b) Bivariate Analysis

In bivariate analysis, we analyze two variables at a time. Often, we see whether there is a relationship between the two variables. The scatter plot is a classic example of bivariate data visualization.

c) Multivariate Analysis

In multivariate analysis, we analyze more than two variables simultaneously. The heatmap is a classic example of multivariate data visualization. Other examples are cluster analysis and principal component analysis (PCA).

4. DATASET

A total of 73 customers, aged between 24 and 81 years old, were surveyed. Of the customers surveyed, 38 were returning customers, and 35 were new customers. The variables chosen are the essential ones that make up the non-verbal communication system. In addition to being the most feasible to evaluate in clients. The non-verbal system is made up of subsystems such as kinesic, paralanguage, proxemic, chronic, and others. In the design of the questionnaire, the indicators that make up these subsystems were taken into account to be explored as part of the client's communication preferences, as well as being feasible to evaluate in clients. The 22 variables analyzed were considered feasible to evaluate by the hotel's clientele.

4.1 Column data :-

country

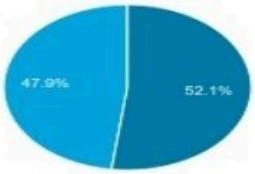
Country of the Client

england	34%
canada	25%
Other (30)	41%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	16	
Most Common	england	34%

returning

If the Client is returning



true
38 52%

false
35 48%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
True	38	52%
False	35	48%

GImg1

Handshakes (Indifferent, likes, dislikes)

likes	68%
indiferent	23%
Other (6)	8%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	3	
Most Common	likes	68%

GImg2

Hug (Indifferent, likes, dislikes)

likes	53%
indiferent	30%
Other (12)	16%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	3	
Most Common	likes	53%

GImg3

Kiss (Indifferent, likes, dislikes)

likes	56%
indiferent	26%
Other (13)	18%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	4	
Most Common	likes	56%

PImg1

Consent posture (Indifferent, likes, dislikes)

likes	96%
indiferent	4%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	2	
Most Common	likes	96%

PImg2

Interest posture (Indifferent, likes, dislikes)

1
unique value

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	1	
Most Common	likes	100%

PImg3

Neutral posture (Indifferent, likes, dislikes)

likes	62%
dislikes	25%
Other (10)	14%

Valid	73	100%
Mismatched	0	0%
Missing	0	0%
Unique	3	
Most Common	likes	62%

4.2 The features are :-

'sex':Sex of the Client
'age':Age of the Client
'country':Country of the Client
'returning':If the Client is returning
'GImg1':Handshakes (Indifferent, likes, dislikes)
'GImg2':Hug (Indifferent, likes, dislikes)
'GImg3':Kiss (Indifferent, likes, dislikes)
'PImg1':Consent posture (Indifferent, likes, dislikes)
'PImg2':Interest posture (Indifferent, likes, dislikes)
'PImg3':Neutral posture (Indifferent, likes, dislikes)
'PImg4':Reflexive posture (Indifferent, likes, dislikes)
'PImg5':Negative posture (Indifferent, likes, dislikes)
'Tense - relaxed':Relaxed observed emotional clime (1 is too tensed & 10 is too relaxed)
'Authoritative -anarchic ':Anarchic observed emotional clime (1 is too authoritative & 10 is too anarchic)
'Hostile - friendly':Observed emotional clime (1 is too hostile & 10 is too friendly)
'TAudio1':Authoritative (Indifferent, likes, dislikes)
'TAudio2':Sarcastic (Indifferent, likes, dislikes)
'TAudio3':Friendly (Indifferent, likes, dislikes)
'QAudio1':Spitting (Indifferent, likes, dislikes)
'QAudio2':Hum (Indifferent, likes, dislikes)
'QAudio3':Sigh (Indifferent, likes, dislikes)
'Proxemics':Physical distance preferred for the client A, B, C, D, ? (A. intimate: 15cm-45cm; B. per-sonal: 46cm-122cm; C. social:
'Type of Client':Class type of Client

5. LIBRARIES

5.1 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

5.2 Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas.
Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

5.3 Tensor Flow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks

5.4 Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

6. METHODOLOGY

6.1 One Hot Encoding

In machine learning, one-hot encoding is a frequently used method to deal with categorical data. Because many machine learning models need their input variables to be numeric, categorical variables need to be transformed in the pre-processing part.One-hot encoding is often applied to nominal variables, in order to improve the performance of the algorithm. For each unique value in the original categorical column, a new column is created in this method. These dummy variables are then filled up with zeros and ones (1 meaning TRUE, 0 meaning FALSE).

6.2 HeatMap

A heat map is a data visualisation technique that shows magnitude of a phenomenon as color in two dimensions. The variation in color may be by hue or intensity, giving obvious visual cues to the reader about how the phenomenon is clustered or varies over space. There are two fundamentally different categories of heat maps: the cluster heat map and the spatial heat map.

7. IMPLEMENTATION

Importing

```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df=pd.read_csv('/home/student/Videos/tourist/non-verbal tourist data.csv')
```

Data Outline

```
df.head()
```

	sex	age	country	returning	GImg1	GImg2	GImg3	PImg1	PImg2	PImg3	...	Authoritative -anarchic	Hostile - friendly	TAudio1	TAudio2	TAudio3	QAudio1	QAudio2	QAudio3	Proxemics	Type of Client
0	F	42	uruguay	no	likes	indiferent	indiferent	likes	likes	indiferent	...	8	3	C	dislikes	dislikes	indiferent	dislikes	dislikes	dislikes	0
1	M	60	brasil	no	likes	indiferent	indiferent	likes	likes	likes	...	9	1	B	dislikes	dislikes	likes	likes	dislikes	dislikes	0
2	F	25	england	no	indiferent	indiferent	indiferent	likes	likes	indiferent	...	10	5	C	dislikes	dislikes	likes	dislikes	dislikes	dislikes	0
3	M	43	canada	no	likes	dislikes	dislikes	likes	likes	indiferent	...	2	1	C	dislikes	dislikes	likes	dislikes	indiferent	dislikes	0
4	M	30	hungary	no	indiferent	indiferent	indiferent	likes	likes	indiferent	...	9	3	C	dislikes	dislikes	likes	indiferent	dislikes	dislikes	0

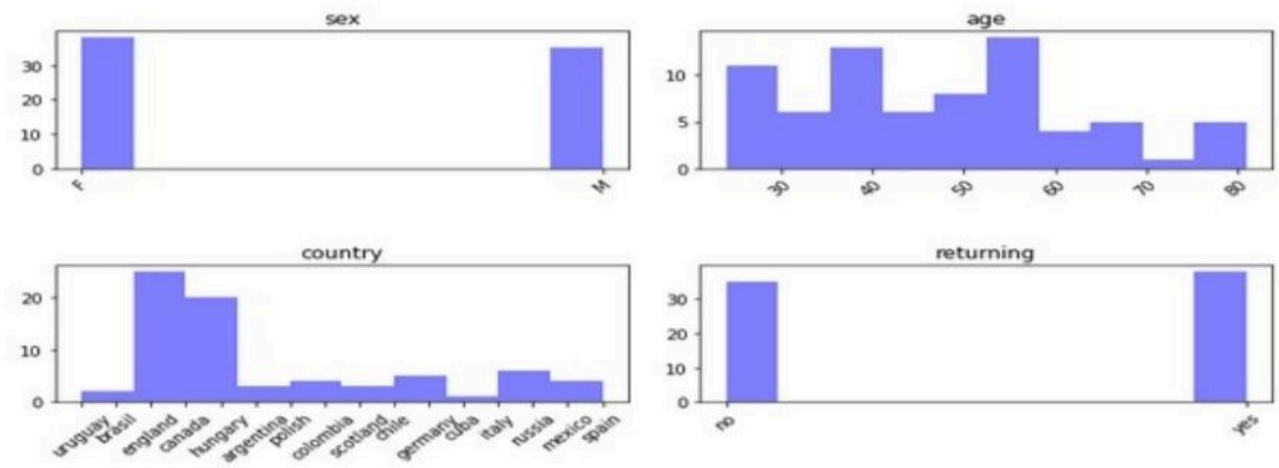
5 rows × 23 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 73 entries, 0 to 72
Data columns (total 23 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   sex                        73 non-null     object
 1   age                        73 non-null     int64
 2   country                    73 non-null     object
 3   returning                  73 non-null     object
 4   GImg1                      73 non-null     object
 5   GImg2                      73 non-null     object
 6   GImg3                      73 non-null     object
 7   PImg1                      73 non-null     object
 8   PImg2                      73 non-null     object
 9   PImg3                      73 non-null     object
 10  PImg4                      73 non-null     object
 11  PImg5                      73 non-null     object
 12  Tense - relaxed            73 non-null     int64
 13  Authoritative -anarchic    73 non-null     object
 14  Hostile - friendly         73 non-null     int64
 15  TAudio1                    73 non-null     object
 16  TAudio2                    73 non-null     object
 17  TAudio3                    73 non-null     object
 18  QAudio1                    73 non-null     object
 19  QAudio2                    73 non-null     object
 20  QAudio3                    73 non-null     object
 21  Proxemics                  73 non-null     object
 22  Type of Client             73 non-null     int64
dtypes: int64(4), object(19)
memory usage: 13.2+ KB
```

#Histogram by features

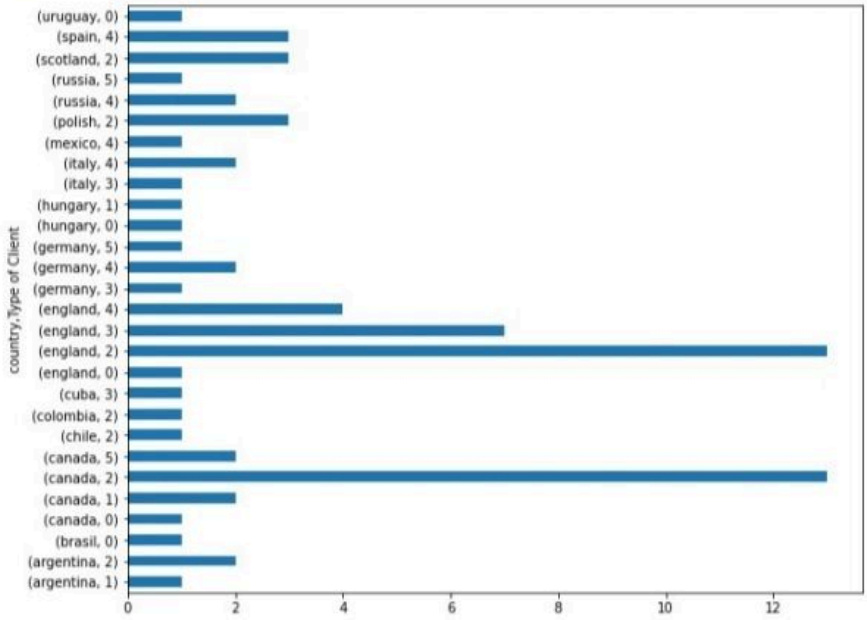
```
num_list = list(df.columns)
fig = plt.figure(figsize=(10,30))
for i in range(len(num_list)):
    plt.subplot(12,2,i+1)
    plt.title(num_list[i])
    plt.xticks(rotation=45)
    plt.hist(df[num_list[i]],color='blue',alpha=0.5)

for i in range(len(num_list)):
    plt.subplot(12,2,i+1)
    plt.title(num_list[i])
    plt.xticks(rotation=45)

plt.tight_layout()
```



#Type of client by countries

```
df.groupby(['country','Type of Client'])
['country'].count().plot.barh(figsize=(10,8))
```

Out[10]: <AxesSubplot:ylabel='country,Type of Client'>



#One Hot Encoding

```
df_new = df.drop(['sex','age','country','returning'],axis=1)
df_new = pd.get_dummies(df_new)
df_new['sex']=df['sex']
df_new['age']=df['age']
df_new['returning']=df['returning']
df_new['country']=df['country']
df_new.T
```

Out[12]:

	0	1	2	3	4	5	6	7	8	9	...	63	64	65	66	67	68	69	70	71	72
Tense - relaxed	4	2	5	3	3	1	1	1	4	1	...	5	5	3	2	3	5	5	2	5	4
Hostile - friendly	3	1	5	1	3	1	2	1	3	2	...	6	6	2	2	2	4	5	2	4	5
Type of Client	0	0	0	0	0	1	1	1	1	2	...	4	4	4	4	4	4	5	5	5	5
Gimg1_dislikes	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	1	1
Gimg1_indiferent	0	0	1	0	1	0	1	0	0	0	...	1	1	1	1	1	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Proxemics_likes	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
sex	F	M	F	M	M	F	M	M	F	M	...	F	F	F	F	F	F	M	F	M	M
age	42	60	25	43	30	31	64	51	28	78	...	56	48	25	25	27	36	38	46	39	27
returning	no	no	no	no	no	yes	yes	yes	no	yes	...	no	no	no	no	no	no	no	no	no	no
country	uruguay	brasil	england	canada	hungary	canada	canada	argentina	hungary	england	...	russia	germany	spain	spain	spain	england	russia	canada	germany	canada

61 rows × 73 columns

```
df_new=df_new.replace({'F': 0, 'M': 1})
```

```
df_new.head()
```

	Tense - relaxed	Hostile - friendly	Type of Client	Glmg1_dislikes	Glmg1_indiferent	Glmg1_likes	Glmg2_dislikes	Glmg2_indiferent	Glmg2_likes	Glmg3_?	...	sex	age	returning	country
0	4	3	0	0	0	1	0	1	0	0	...	0	42	no	uruguay
1	2	1	0	0	0	1	0	1	0	0	...	1	60	no	brasil
2	5	5	0	0	1	0	0	1	0	0	...	0	25	no	england
3	3	1	0	0	0	1	1	0	0	0	...	1	43	no	canada
4	3	3	0	0	1	0	0	1	0	0	...	1	30	no	hungary

```
pd.set_option('display.max_rows', 65)
```

#Average of features by countries

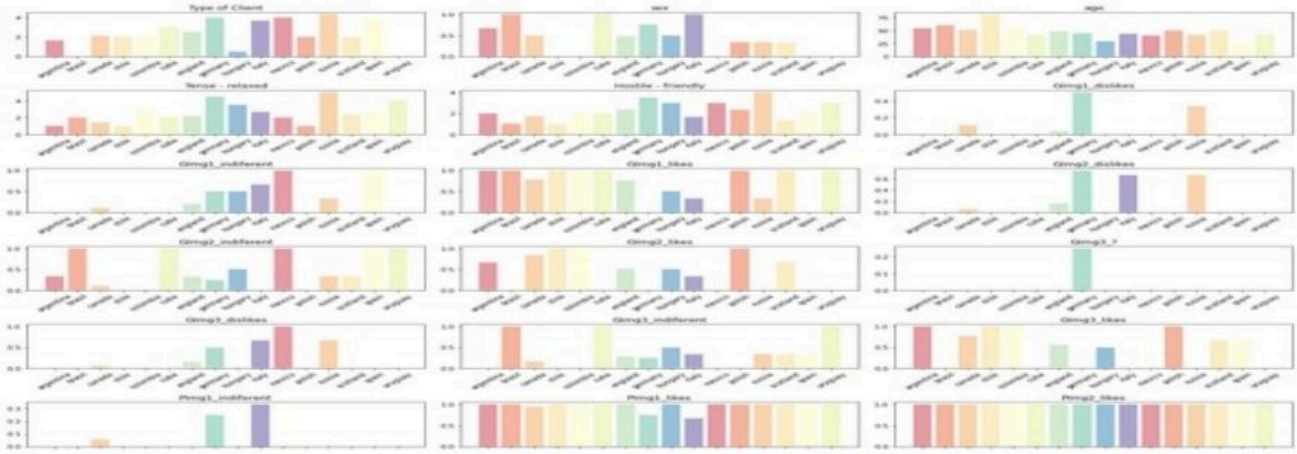
```
df_country_ave=df_new.groupby('country').mean()
df_country_ave.head()
```

country	Tense - relaxed	Hostile - friendly	Type of Client	Glmg1_dislikes	Glmg1_indiferent	Glmg1_likes	Glmg2_dislikes	Glmg2_indiferent	Glmg2_likes	Glmg3_?	...	Proxemics_indiferent	Proxemics_likes	sex	age
argentina	1.000000	2.000000	1.666667	0.000000	0.000000	1.000000	0.000000	0.333333	0.666667	0.0	...	0.000000	0.000000	0.666667	54.333333
brasil	2.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000	1.000000	0.000000	0.0	...	0.000000	0.000000	1.000000	60.000000
canada	1.444444	1.722222	2.111111	0.111111	0.111111	0.777778	0.055556	0.111111	0.833333	0.0	...	0.222222	0.055556	0.500000	50.388889
chile	1.000000	1.000000	2.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.0	...	1.000000	0.000000	0.000000	81.000000
colombia	3.000000	2.000000	2.000000	0.000000	0.000000	1.000000	0.000000	0.000000	1.000000	0.0	...	0.000000	0.000000	0.000000	53.000000

5 rows × 59 columns

```
num_list = list(df_country_ave.columns)
fig = plt.figure(figsize=(20,50))

cm = plt.get_cmap("Spectral")
color_maps = [cm(0.1), cm(0.2), cm(0.3), cm(0.4),
cm(0.5),cm(0.6),cm(0.7),cm(0.8),cm(0.9),cm(1.0)]
for i in range(len(num_list)):
```

```
    plt.subplot(21,3,i+1)
    plt.title(num_list[i])
    plt.xticks(rotation=45)

plt.bar(x=df_country_ave.index,height=df_country_ave[num_list[i
]],color=color_maps,alpha=0.5)

plt.tight_layout()
```



#Average of features by sex

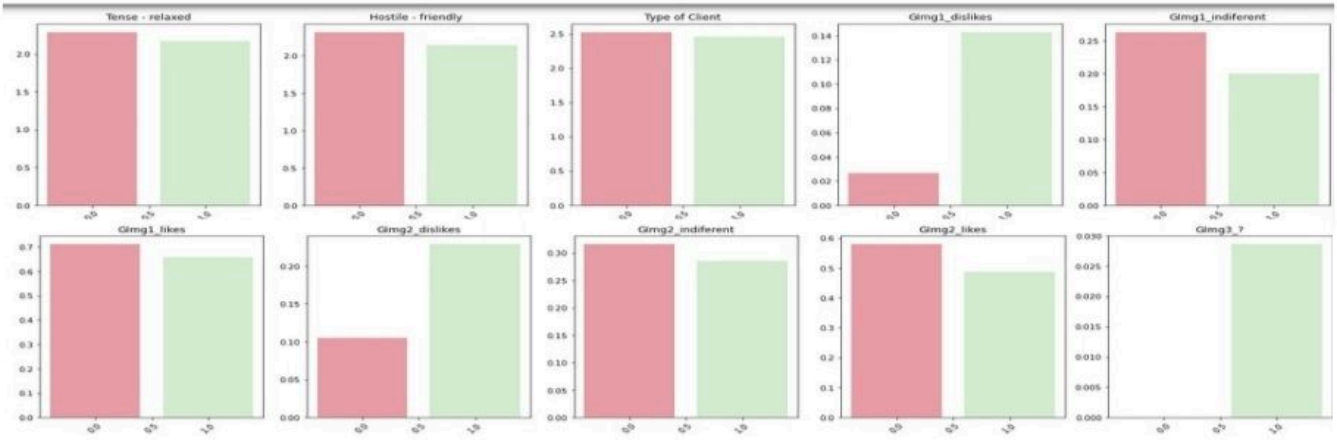
```
df_sex_ave=df_new.groupby('sex').mean()
df_sex_ave.head()
```

sex	Tense - relaxed	Hostile - friendly	Type of Client	Glmg1_dislikes	Glmg1_indiferent	Glmg1_likes	Glmg2_dislikes	Glmg2_indiferent	Glmg2_likes	Glmg3_?	...	QAudio3_likes	Proxemics_dislikes	Proxemics_indiferent	Proxemics_likes	age
0	2.289474	2.315789	2.526316	0.026316	0.263158	0.710526	0.105263	0.315789	0.578947	0.000000	...	0.026316	0.842105	0.131579	0.026316	45.078947
1	2.171429	2.142857	2.457143	0.142857	0.200000	0.657143	0.228571	0.285714	0.485714	0.028571	...	0.028571	0.885714	0.085714	0.028571	49.628571

2 rows × 58 columns

```
num_list2 = list(df_sex_ave.columns)
fig = plt.figure(figsize=(20,60))
cm = plt.get_cmap("Spectral")
color_maps = [cm(0.1),cm(0.7)]
for i in range(len(num_list2)):
    plt.subplot(13,5,i+1)
    plt.title(num_list2[i])
    plt.xticks(rotation=45)
```

```
plt.bar(x=df_sex_ave.index,height=df_sex_ave[num_list2[i]],color
=color_maps,alpha=0.5)
plt.tight_layout()
```



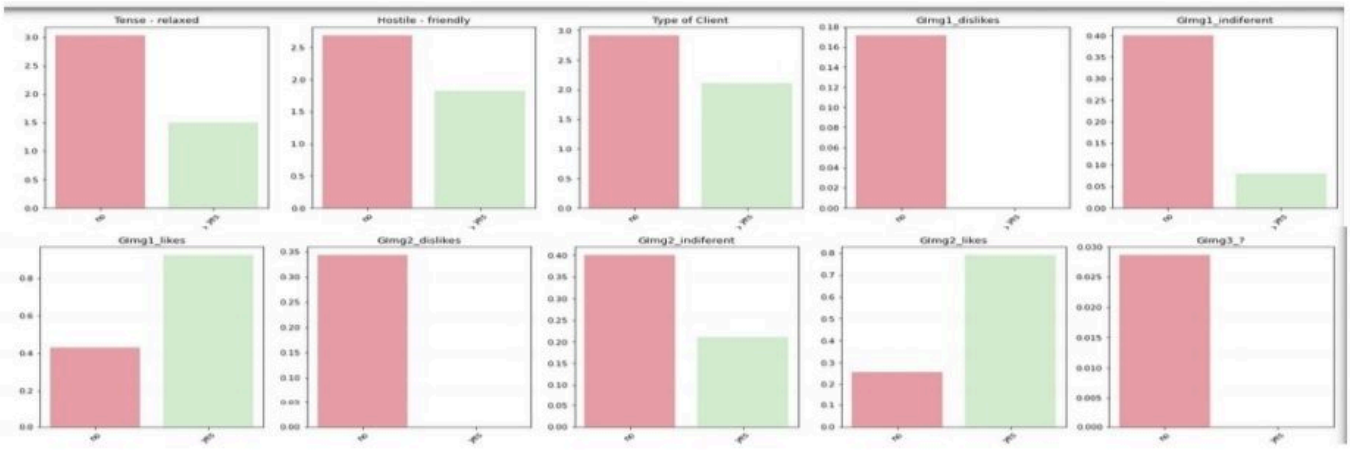
#Average of features by returning

```
df_returning_ave=df_new.groupby('returning').mean()
df_returning_ave.head()
```

returning	Tense - relaxed	Hostile - friendly	Type of Client	Glmg1_dislikes	Glmg1_indiferent	Glmg1_likes	Glmg2_dislikes	Glmg2_indiferent	Glmg2_likes	Glmg3_?	...	Proxemics_dislikes	Proxemics_indiferent	Proxemics_likes	sex	age
no	3.028571	2.685714	2.914286	0.171429	0.400000	0.428571	0.342857	0.400000	0.257143	0.028571	...	0.914286	0.057143	0.028571	0.457143	42.228571
yes	1.500000	1.815789	2.105263	0.000000	0.078947	0.921053	0.000000	0.210526	0.789474	0.000000	...	0.815789	0.157895	0.026316	0.500000	51.894737

2 rows × 59 columns

```
num_list4 = list(df_returning_ave.columns)
fig = plt.figure(figsize=(20,60))
cm = plt.get_cmap("Spectral")
color_maps = [cm(0.1),cm(0.7)]

for i in range(len(num_list4)):
    plt.subplot(13,5,i+1)
    plt.title(num_list4[i])
    plt.xticks(rotation=45)

plt.bar(x=df_returning_ave.index,height=df_returning_ave[num_li
st4[i]],color=color_maps,alpha=0.5)
plt.tight_layout()
```



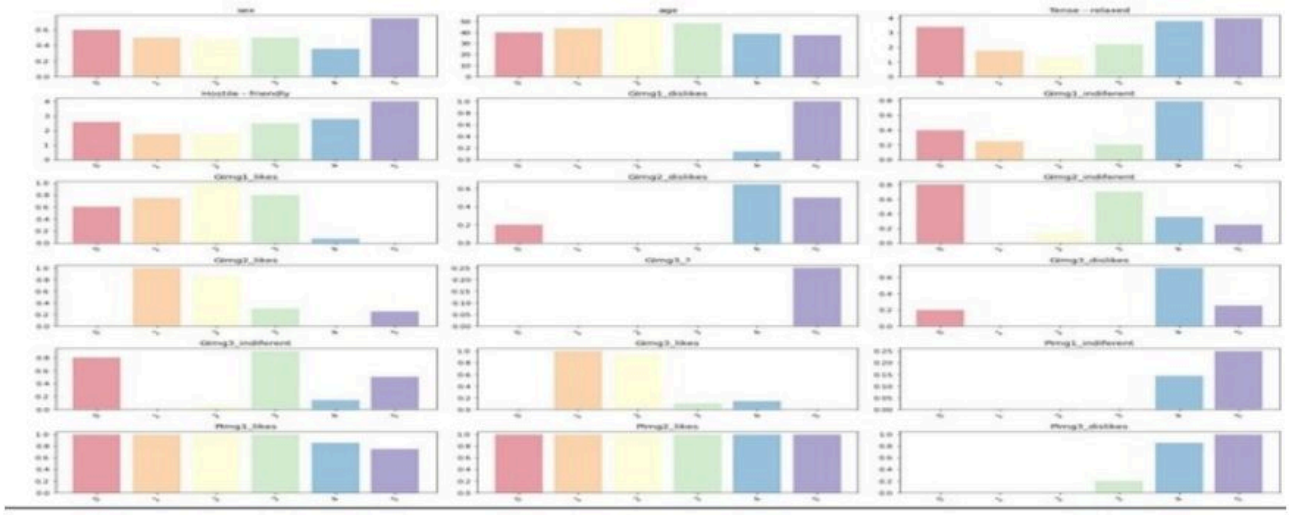
#Average of features by type of client (& majoy in countries)

```
df_client_ave=df_new.groupby('Type of Client').mean()
df_client_ave.head()
```

Type of Client	Tense - relaxed	Hostile - friendly	Glmg1_dislikes	Glmg1_indiferent	Glmg1_likes	Glmg2_dislikes	Glmg2_indiferent	Glmg2_likes	Glmg3_?	Glmg3_dislikes	...	Proxemics_indiferent	Proxemics_likes	sex	age
0	3.400000	2.600000	0.000000	0.400000	0.600000	0.200000	0.800000	0.000000	0.0	0.200000	...	0.000000	0.000000	0.600000	40.000000
1	1.750000	1.750000	0.000000	0.250000	0.750000	0.000000	0.000000	1.000000	0.0	0.000000	...	0.000000	0.000000	0.500000	43.500000
2	1.333333	1.750000	0.000000	0.027778	0.972222	0.000000	0.138889	0.861111	0.0	0.000000	...	0.166667	0.027778	0.472222	52.861111
3	2.200000	2.500000	0.000000	0.200000	0.800000	0.000000	0.700000	0.300000	0.0	0.000000	...	0.100000	0.000000	0.500000	48.200000
4	3.785714	2.785714	0.142857	0.785714	0.071429	0.642857	0.357143	0.000000	0.0	0.714286	...	0.000000	0.000000	0.357143	38.642857

5 rows × 58 columns

```
num_list3 = list(df_client_ave.columns)
fig = plt.figure(figsize=(20,50))
cm = plt.get_cmap("Spectral")
color_maps = [cm(0.1), cm(0.3), cm(0.5), cm(0.7),
cm(0.9),cm(1.1)]

for i in range(len(num_list3)):
    plt.subplot(21,3,i+1)
    plt.title(num_list3[i])
    plt.xticks(rotation=45)

plt.bar(x=df_client_ave.index,height=df_client_ave[num_list3[i]],
color=color_maps,alpha=0.5)
plt.tight_layout()
```



#What type has maximam average of features ?

#Type 0 : Brazil, Hangary, Uruguay

#type 1 : Hungary

#Type 2 : Argentina, Canada, Chile, Colombia,England,Scotland, Polish

#Type 3 : Cuba

#Type 4 : Germany, Spain, Russia, Mexico, Italy

#Type 5 : No Major

```
df_type=pd.DataFrame(df_client_ave.idxmax())
df_type=df_type.reset_index()
df_type=df_type.sort_values(0)
df_type=df_type.rename(columns={'index': 'feature', 0: 'Type of client'})
```

```
def countries(ex):
    if ex==0:
        return 'Brazil, Hangary, Uruguay'
    elif ex==1:
        return 'Hungary'
    elif ex==2:
        return 'Aegentina, Canada, Chile, Colombia,England,Scotland,
Polish'
    elif ex==3:
        return 'Cuba'
    elif ex==4:
        return 'Germany, Spain, Russia, Mexico, Italy'
    else:
        return 'No major'
```

```
df_type['countries which has miximam average']=df_type['Type of client'].apply(countries)
df_type
```

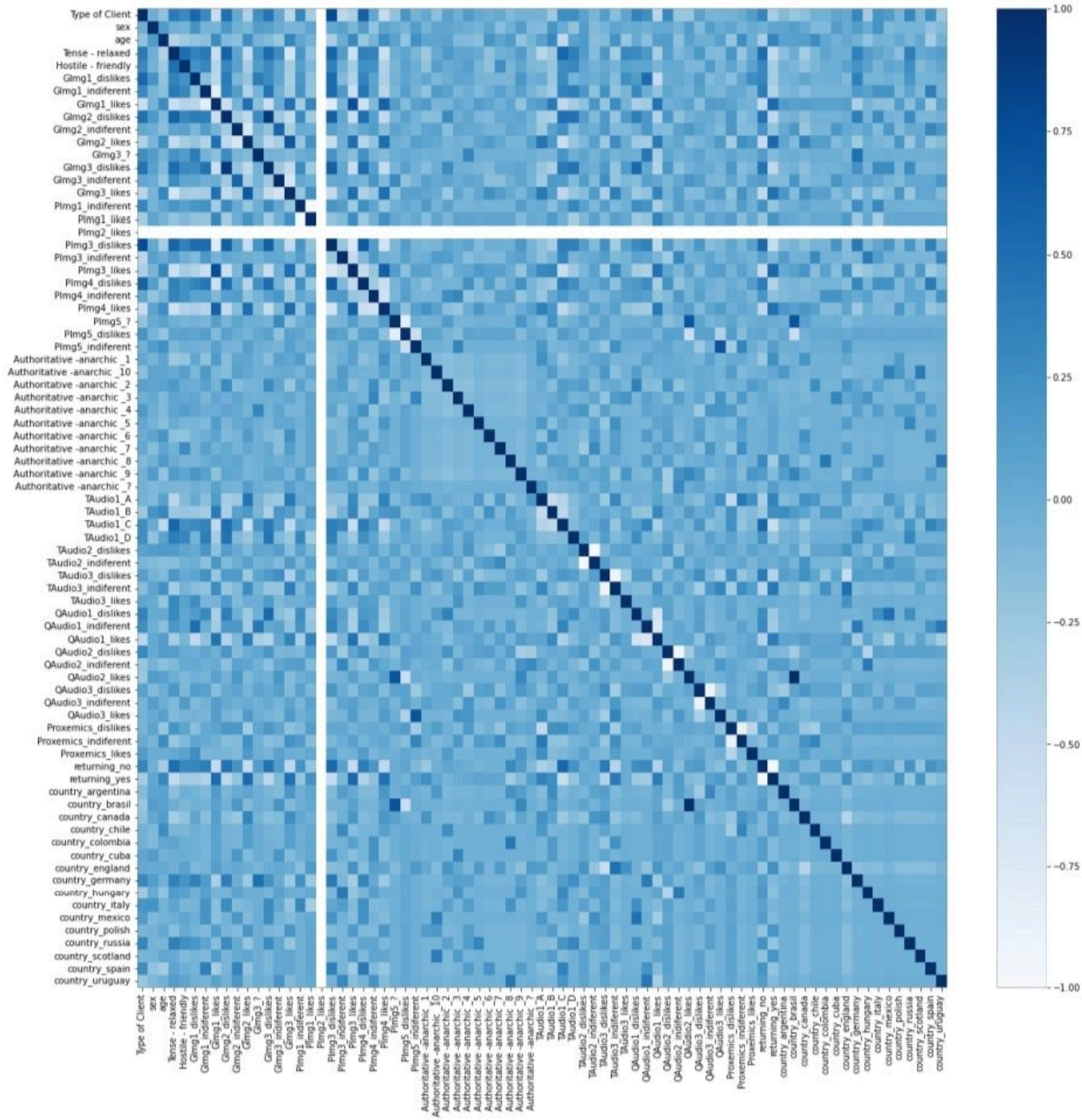
Out[31]:

	feature	Type of client	countries which has miximam average
41	TAudio3_dislikes	0	Brazil, Hangary, Uruguay
21	PImg5_?	0	Brazil, Hangary, Uruguay
39	TAudio2_dislikes	0	Brazil, Hangary, Uruguay
6	GImg2_indiferent	0	Brazil, Hangary, Uruguay
49	QAudio2_likes	0	Brazil, Hangary, Uruguay
53	Proxemics_dislikes	0	Brazil, Hangary, Uruguay
37	TAudio1_C	0	Brazil, Hangary, Uruguay
32	Authoritative -anarchic _8	0	Brazil, Hangary, Uruguay
16	PImg3_indiferent	0	Brazil, Hangary, Uruguay
13	PImg1_likes	0	Brazil, Hangary, Uruguay
14	PImg2_likes	0	Brazil, Hangary, Uruguay
46	QAudio1_likes	1	Hungary
27	Authoritative -anarchic _3	1	Hungary
22	PImg5_dislikes	1	Hungary
33	Authoritative -anarchic _9	1	Hungary
25	Authoritative -anarchic _10	1	Hungary
50	QAudio3_dislikes	1	Hungary
11	GImg3_likes	1	Hungary
36	TAudio1_B	1	Hungary
7	GImg2_likes	1	Hungary
40	TAudio2_indiferent	1	Hungary
48	QAudio2_indiferent	1	Hungary
30	Authoritative -anarchic _6	2	Aegentina, Canada, Chile, Colombia,England,Sco...
42	TAudio3_indiferent	2	Aegentina, Canada, Chile, Colombia,England,Sco...
34	Authoritative -anarchic _?	2	Aegentina, Canada, Chile, Colombia,England,Sco...
35	TAudio1_A	2	Aegentina, Canada, Chile, Colombia,England,Sco...
57	age	2	Aegentina, Canada, Chile, Colombia,England,Sco...
23	PImg5_indiferent	2	Aegentina, Canada, Chile, Colombia,England,Sco...
20	PImg4_likes	2	Aegentina, Canada, Chile, Colombia,England,Sco...
4	GImg1_likes	2	Aegentina, Canada, Chile, Colombia,England,Sco...
52	QAudio3_likes	2	Aegentina, Canada, Chile, Colombia,England,Sco...
17	PImg3_likes	2	Aegentina, Canada, Chile, Colombia,England,Sco...
26	Authoritative -anarchic _2	3	Cuba
51	QAudio3_indiferent	3	Cuba

#Correlation among feature

```
df_new2=pd.get_dummies(df_new)
```

```
plt.figure(figsize = (20,20))
sns.heatmap(df_new2.corr(),cbar=True, cmap='Blues')
```



#Type of Client Classify Model

```
from sklearn.model_selection import train_test_split as split
import tensorflow as tf
import keras
from keras.layers import Dense,Activation
```

```
from sklearn.metrics import classification_report,
confusion_matrix
import itertools
```

```
X=df_new2.drop(['Type of Client'],axis=1)
y=df_new2['Type of Client']
```

```
x_train, x_test, y_train, y_test =
split(X,y,train_size=0.8,test_size=0.2)
```

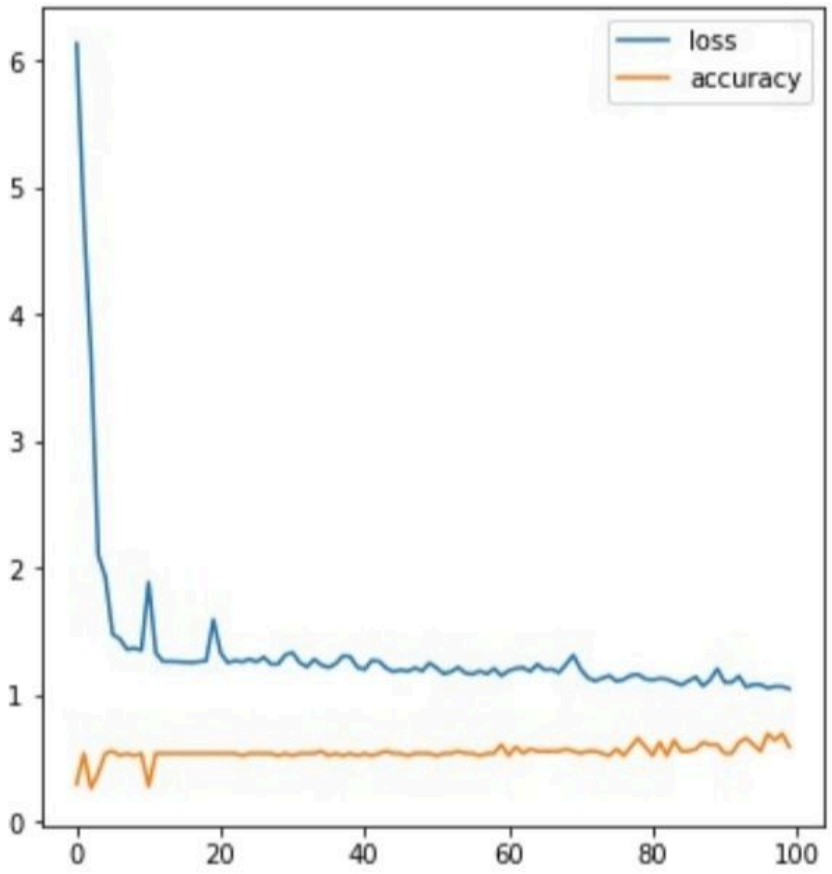
```
model = keras.models.Sequential()
model.add(Dense(units=32,input_dim=76))
model.add(Activation('relu'))
model.add(Dense(units=6))
model.add(Activation('softmax'))
model.compile(loss='sparse_categorical_crossentropy',optimizer
='sgd',metrics=['accuracy'])
```

```
history=model.fit(x_train,y_train,epochs=120)
```

```
Epoch 1/120
2/2 [==============================] - 4s 18ms/step - loss: 9.4108 - accuracy: 0.3103
Epoch 2/120
2/2 [==============================] - 0s 34ms/step - loss: 4.3402 - accuracy: 0.4483
Epoch 3/120
2/2 [==============================] - 0s 17ms/step - loss: 10.6397 - accuracy: 0.1379
Epoch 4/120
2/2 [==============================] - 0s 21ms/step - loss: 6.7905 - accuracy: 0.2586
Epoch 5/120
2/2 [==============================] - 0s 12ms/step - loss: 5.8799 - accuracy: 0.3103
Epoch 6/120
2/2 [==============================] - 0s 65ms/step - loss: 3.4295 - accuracy: 0.3276
Epoch 7/120
2/2 [==============================] - 0s 23ms/step - loss: 3.2459 - accuracy: 0.3103
Epoch 8/120
2/2 [==============================] - 0s 17ms/step - loss: 2.5754 - accuracy: 0.4655
Epoch 9/120
2/2 [==============================] - 0s 9ms/step - loss: 1.7395 - accuracy: 0.4655
Epoch 112/120
2/2 [==============================] - 0s 8ms/step - loss: 1.1316 - accuracy: 0.5345
Epoch 113/120
2/2 [==============================] - 0s 6ms/step - loss: 1.7017 - accuracy: 0.3448
Epoch 114/120
2/2 [==============================] - 0s 6ms/step - loss: 1.2023 - accuracy: 0.5517
Epoch 115/120
2/2 [==============================] - 0s 7ms/step - loss: 1.1497 - accuracy: 0.6379
Epoch 116/120
2/2 [==============================] - 0s 7ms/step - loss: 1.1849 - accuracy: 0.6207
Epoch 117/120
2/2 [==============================] - 0s 6ms/step - loss: 1.0704 - accuracy: 0.6724
Epoch 118/120
2/2 [==============================] - 0s 6ms/step - loss: 1.1305 - accuracy: 0.6034
Epoch 119/120
2/2 [==============================] - 0s 15ms/step - loss: 1.1064 - accuracy: 0.5690
Epoch 120/120
2/2 [==============================] - 0s 12ms/step - loss: 1.2335 - accuracy: 0.5517
```

#Type of Client Classify Model

```
plt.figure(figsize=(19,6))
plt.subplot(131)
plt.plot(history.epoch, history.history['loss'], label="loss")
plt.plot(history.epoch, history.history['accuracy'],
label="accuracy")
plt.legend()
```



8. Conclusion

We practiced machine learning including libraries such as tensorflow, matplotlib,seaborn and keras to obtain proper visualization of data for the tourists who use non verbal language while travelling to foriegn countries.We obtained a stable accuracy and exponentially decreasing rate of loss function. Using heatmap we found out the corelation between features. This allows us to easily visualize the data and use this data to further process it for various other uses such as finding the behavioral window into psychological processes to understand the human behaviour.