In [1]:
```
#Importing Libraries
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from tkinter import *
import numpy as np
import pandas as pd
import os
```

In [2]:
```
#List of the symptoms is listed here in list l1.

l1=['back_pain','constipation','abdominal_pain','diarrhoea','mild_fever','yellow_urine
    'yellowing_of_eyes','acute_liver_failure','fluid_overload','swelling_of_stomach',
    'swelled_lymph_nodes','malaise','blurred_and_distorted_vision','phlegm','throat_ir
    'redness_of_eyes','sinus_pressure','runny_nose','congestion','chest_pain','weaknes
    'fast_heart_rate','pain_during_bowel_movements','pain_in_anal_region','bloody_stoo
    'irritation_in_anus','neck_pain','dizziness','cramps','bruising','obesity','swolle
    'swollen_blood_vessels','puffy_face_and_eyes','enlarged_thyroid','brittle_nails',
    'swollen_extremeties','excessive_hunger','extra_marital_contacts','drying_and_ting
    'slurred_speech','knee_pain','hip_joint_pain','muscle_weakness','stiff_neck','swel
    'movement_stiffness','spinning_movements','loss_of_balance','unsteadiness',
    'weakness_of_one_body_side','loss_of_smell','bladder_discomfort','foul_smell_of ur
    'continuous_feel_of_urine','passage_of_gases','internal_itching','toxic_look_(typh
    'depression','irritability','muscle_pain','altered_sensorium','red_spots_over_body
    'abnormal_menstruation','dischromic _patches','watering_from_eyes','increased_appe
    'rusty_sputum','lack_of_concentration','visual_disturbances','receiving_blood_tran
    'receiving_unsterile_injections','coma','stomach_bleeding','distention_of_abdomen'
    'history_of_alcohol_consumption','fluid_overload','blood_in_sputum','prominent_vei
    'palpitations','painful_walking','pus_filled_pimples','blackheads','scurring','ski
    'silver_like_dusting','small_dents_in_nails','inflammatory_nails','blister','red_s
    'yellow_crust_ooze']
```

In [3]:
```
#List of Diseases is listed in list disease.

disease=['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes ',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension ', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
       'Dimorphic hemmorhoids(piles)', 'Heart attack', 'Varicose veins',
       'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo']
```

In [4]:
```
l2=[]
for i in range(0,len(l1)):
    l2.append(0)
print(l2)
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [6]:
```
#Reading the training .csv file
df=pd.read_csv("training.csv")
DF= pd.read_csv('training.csv', index_col='prognosis')
#Replace the values in the imported file by pandas by the inbuilt function replace in

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Input In [6], in <cell line: 2>()
      1 #Reading the training .csv file
----> 2 df=pd.read_csv("training.csv")
      3 DF= pd.read_csv('training.csv', index_col='prognosis')
      4 #Replace the values in the imported file by pandas by the inbuilt function re
place in pandas.

File ~\anaconda3\lib\site-packages\pandas\util\_decorators.py:311, in deprecate_nonke
yword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    305 if len(args) > num_allow_args:
    306     warnings.warn(
    307         msg.format(arguments=arguments),
    308         FutureWarning,
    309         stacklevel=stacklevel,
    310     )
--> 311 return func(*args, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:680, in read_csv(file
path_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, m
angle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspa
ce, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip
_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirs
t, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator,
quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dial
ect, error_bad_lines, warn_bad_lines, on_bad_lines, delim_whitespace, low_memory, mem
ory_map, float_precision, storage_options)
    665 kwds_defaults = _refine_defaults_read(
    666     dialect,
    667     delimiter,
   (...)
    676     defaults={"delimiter": ","},
    677 )
    678 kwds.update(kwds_defaults)
--> 680 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:575, in _read(filepat
h_or_buffer, kwds)
    572 _validate_names(kwds.get("names", None))
    574 # Create the parser.
--> 575 parser = TextFileReader(filepath_or_buffer, **kwds)
    577 if chunksize or iterator:
    578     return parser

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:933, in TextFileReade
r.__init__(self, f, engine, **kwds)
    930     self.options["has_index_names"] = kwds["has_index_names"]
    932 self.handles: IOHandles | None = None
--> 933 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1217, in TextFileRead
er._make_engine(self, f, engine)
   1213     mode = "rb"
   1214 # error: No overload variant of "get_handle" matches argument types
   1215 # "Union[str, PathLike[str], ReadCsvBuffer[bytes], ReadCsvBuffer[str]]"
   1216 # , "str", "bool", "Any", "Any", "Any", "Any", "Any"
-> 1217 self.handles = get_handle(  # type: ignore[call-overload]
   1218     f,
   1219     mode,
```

```
   1220     encoding=self.options.get("encoding", None),
   1221     compression=self.options.get("compression", None),
   1222     memory_map=self.options.get("memory_map", False),
   1223     is_text=is_text,
   1224     errors=self.options.get("encoding_errors", "strict"),
   1225     storage_options=self.options.get("storage_options", None),
   1226 )
   1227 assert self.handles is not None
   1228 f = self.handles.handle

File ~\anaconda3\lib\site-packages\pandas\io\common.py:789, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    784 elif isinstance(handle, str):
    785     # Check whether the filename is to be opened in binary mode.
    786     # Binary mode does not support 'encoding' and 'newline'.
    787     if ioargs.encoding and "b" not in ioargs.mode:
    788         # Encoding
--> 789         handle = open(
    790             handle,
    791             ioargs.mode,
    792             encoding=ioargs.encoding,
    793             errors=errors,
    794             newline="",
    795         )
    796     else:
    797         # Binary mode
    798         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'training.csv'
```

In [ ]:

In [7]:
```
#Reading the training .csv file
df=pd.read_csv("training.csv")
DF= pd.read_csv('training.csv', index_col='prognosis')
#Replace the values in the imported file by pandas by the inbuilt function replace in

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Input In [7], in <cell line: 2>()
      1 #Reading the training .csv file
----> 2 df=pd.read_csv("training.csv")
      3 DF= pd.read_csv('training.csv', index_col='prognosis')
      4 #Replace the values in the imported file by pandas by the inbuilt function re
place in pandas.

File ~\anaconda3\lib\site-packages\pandas\util\_decorators.py:311, in deprecate_nonke
yword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    305 if len(args) > num_allow_args:
    306     warnings.warn(
    307         msg.format(arguments=arguments),
    308         FutureWarning,
    309         stacklevel=stacklevel,
    310     )
--> 311 return func(*args, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:680, in read_csv(file
path_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, m
angle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspa
ce, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip
_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirs
t, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator,
quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dial
ect, error_bad_lines, warn_bad_lines, on_bad_lines, delim_whitespace, low_memory, mem
ory_map, float_precision, storage_options)
    665 kwds_defaults = _refine_defaults_read(
    666     dialect,
    667     delimiter,
   (...)
    676     defaults={"delimiter": ","},
    677 )
    678 kwds.update(kwds_defaults)
--> 680 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:575, in _read(filepat
h_or_buffer, kwds)
    572 _validate_names(kwds.get("names", None))
    574 # Create the parser.
--> 575 parser = TextFileReader(filepath_or_buffer, **kwds)
    577 if chunksize or iterator:
    578     return parser

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:933, in TextFileReade
r.__init__(self, f, engine, **kwds)
    930     self.options["has_index_names"] = kwds["has_index_names"]
    932 self.handles: IOHandles | None = None
--> 933 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1217, in TextFileRead
er._make_engine(self, f, engine)
   1213     mode = "rb"
   1214 # error: No overload variant of "get_handle" matches argument types
   1215 # "Union[str, PathLike[str], ReadCsvBuffer[bytes], ReadCsvBuffer[str]]"
   1216 # , "str", "bool", "Any", "Any", "Any", "Any", "Any"
-> 1217 self.handles = get_handle(  # type: ignore[call-overload]
   1218     f,
   1219     mode,
```

```
   1220     encoding=self.options.get("encoding", None),
   1221     compression=self.options.get("compression", None),
   1222     memory_map=self.options.get("memory_map", False),
   1223     is_text=is_text,
   1224     errors=self.options.get("encoding_errors", "strict"),
   1225     storage_options=self.options.get("storage_options", None),
   1226 )
   1227 assert self.handles is not None
   1228 f = self.handles.handle

File ~\anaconda3\lib\site-packages\pandas\io\common.py:789, in get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    784 elif isinstance(handle, str):
    785     # Check whether the filename is to be opened in binary mode.
    786     # Binary mode does not support 'encoding' and 'newline'.
    787     if ioargs.encoding and "b" not in ioargs.mode:
    788         # Encoding
--> 789         handle = open(
    790             handle,
    791             ioargs.mode,
    792             encoding=ioargs.encoding,
    793             errors=errors,
    794             newline="",
    795         )
    796     else:
    797         # Binary mode
    798         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'training.csv'
```

In [10]:

```
#Reading the training .csv file
df=pd.read_csv("C:/Users/Sazna/OneDrive/Desktop/training.csv")
DF= pd.read_csv( 'C:\Users\Sazna\OneDrive\Desktop\training.csv', index_col='prognosis'
#Replace the values in the imported file by pandas by the inbuilt function replace in

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

```
  Input In [10]
    DF= pd.read_csv( 'C:\Users\Sazna\OneDrive\Desktop\training.csv', index_col='progn
osis')
                                                                                    ^

SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-
3: truncated \UXXXXXXXX escape
```

In [11]:

```
#Reading the training .csv file
df=pd.read_csv("C:/Users/Sazna/OneDrive/Desktop/training.csv")
DF= pd.read_csv( 'C:/Users/Sazna/OneDrive/Desktop/training.csv', index_col='prognosis'
#Replace the values in the imported file by pandas by the inbuilt function replace in

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
```

```
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

```
  Input In [11]
    DF= pd.read_csv( 'C:\Users\Sazna\OneDrive\Desktop\training.csv', index_col='progn
osis')
                                                                                    ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-
3: truncated \UXXXXXXXX escape
```

In [12]:
```
#Reading the training .csv file
df=pd.read_csv("C:/Users/Sazna/OneDrive/Desktop/training.csv")
DF= pd.read_csv( 'C:/Users/Sazna/OneDrive/Desktop/training.csv', index_col='prognosis'
#Replace the values in the imported file by pandas by the inbuilt function replace in

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
#df.head()
DF.head()
```

Out[12]:

prognosis	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain
Fungal infection	1	1	1	0	0	0	0
Fungal infection	0	1	1	0	0	0	0
Fungal infection	1	0	1	0	0	0	0
Fungal infection	1	1	0	0	0	0	0
Fungal infection	1	1	1	0	0	0	0

5 rows × 132 columns

In [13]:
```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow):
    nunique = df1.nunique()
```

```
    df1 = df1[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For dis
    nRow, nCol = df1.shape
    columnNames = list(df1)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, face
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

In [14]:

```
# Scatter and density plots
def plotScatterMatrix(df1, plotSize, textSize):
    df1 = df1.select_dtypes(include =[np.number]) # keep only numerical columns
    # Remove rows and columns that would lead to df being singular
    df1 = df1.dropna('columns')
    df1 = df1[[col for col in df if df[col].nunique() > 1]] # keep columns where there
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion of k
        columnNames = columnNames[:10]
    df1 = df1[columnNames]
    ax = pd.plotting.scatter_matrix(df1, alpha=0.75, figsize=[plotSize, plotSize], dia
    corrs = df1.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axe
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

In [15]:

```
plotPerColumnDistribution(df, 10, 5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [15], in <cell line: 1>()
----> 1 plotPerColumnDistribution(df, 10, 5)

Input In [13], in plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow)
      8 plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
facecolor = 'w', edgecolor = 'k')
      9 for i in range(min(nCol, nGraphShown)):
---> 10     plt.subplot(nGraphRow, nGraphPerRow, i + 1)
     11     columnDf = df.iloc[:, i]
     12     if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):

File ~\anaconda3\lib\site-packages\matplotlib\pyplot.py:1268, in subplot(*args, **kwa
rgs)
   1265 fig = gcf()
   1267 # First, search for an existing subplot with a matching spec.
-> 1268 key = SubplotSpec._from_subplot_args(fig, args)
   1270 for ax in fig.axes:
   1271     # if we found an axes at the position sort out if we can re-use it
   1272     if hasattr(ax, 'get_subplotspec') and ax.get_subplotspec() == key:
   1273         # if the user passed no kwargs, re-use

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:597, in SubplotSpec._from_s
ubplot_args(figure, args)
    593 else:
    594     raise TypeError(f"subplot() takes 1 or 3 positional arguments but "
    595                     f"{len(args)} were given")
--> 597 gs = GridSpec._check_gridspec_exists(figure, rows, cols)
    598 if gs is None:
    599     gs = GridSpec(rows, cols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:225, in GridSpecBase._check
_gridspec_exists(figure, nrows, ncols)
    223             return gs
    224 # else gridspec not found:
--> 225 return GridSpec(nrows, ncols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:385, in GridSpec.__init__(s
elf, nrows, ncols, figure, left, bottom, right, top, wspace, hspace, width_ratios, he
ight_ratios)
    382 self.hspace = hspace
    383 self.figure = figure
--> 385 super().__init__(nrows, ncols,
    386                  width_ratios=width_ratios,
    387                  height_ratios=height_ratios)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:49, in GridSpecBase.__init_
_(self, nrows, ncols, height_ratios, width_ratios)
     34 """
     35 Parameters
     36 ----------
   (...)
     46     If not given, all rows will have the same height.
     47 """
     48 if not isinstance(nrows, Integral) or nrows <= 0:
---> 49     raise ValueError(
     50         f"Number of rows must be a positive integer, not {nrows!r}")
     51 if not isinstance(ncols, Integral) or ncols <= 0:
     52     raise ValueError(
```

```
     53         f"Number of columns must be a positive integer, not {ncols!r}")

ValueError: Number of rows must be a positive integer, not 27.2
```

<Figure size 2400x17408 with 0 Axes>

In [16]:
```
plotPerColumnDistribution(df, 10, 5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [16], in <cell line: 1>()
----> 1 plotPerColumnDistribution(df, 10, 5)

Input In [13], in plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow)
      8 plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
facecolor = 'w', edgecolor = 'k')
      9 for i in range(min(nCol, nGraphShown)):
---> 10     plt.subplot(nGraphRow, nGraphPerRow, i + 1)
     11     columnDf = df.iloc[:, i]
     12     if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):

File ~\anaconda3\lib\site-packages\matplotlib\pyplot.py:1268, in subplot(*args, **kwa
rgs)
   1265 fig = gcf()
   1267 # First, search for an existing subplot with a matching spec.
-> 1268 key = SubplotSpec._from_subplot_args(fig, args)
   1270 for ax in fig.axes:
   1271     # if we found an axes at the position sort out if we can re-use it
   1272     if hasattr(ax, 'get_subplotspec') and ax.get_subplotspec() == key:
   1273         # if the user passed no kwargs, re-use

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:597, in SubplotSpec._from_s
ubplot_args(figure, args)
    593 else:
    594     raise TypeError(f"subplot() takes 1 or 3 positional arguments but "
    595                     f"{len(args)} were given")
--> 597 gs = GridSpec._check_gridspec_exists(figure, rows, cols)
    598 if gs is None:
    599     gs = GridSpec(rows, cols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:225, in GridSpecBase._check
_gridspec_exists(figure, nrows, ncols)
    223             return gs
    224 # else gridspec not found:
--> 225 return GridSpec(nrows, ncols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:385, in GridSpec.__init__(s
elf, nrows, ncols, figure, left, bottom, right, top, wspace, hspace, width_ratios, he
ight_ratios)
    382 self.hspace = hspace
    383 self.figure = figure
--> 385 super().__init__(nrows, ncols,
    386                  width_ratios=width_ratios,
    387                  height_ratios=height_ratios)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:49, in GridSpecBase.__init_
_(self, nrows, ncols, height_ratios, width_ratios)
     34 """
     35 Parameters
     36 ----------
   (...)
     46     If not given, all rows will have the same height.
     47 """
     48 if not isinstance(nrows, Integral) or nrows <= 0:
---> 49     raise ValueError(
     50         f"Number of rows must be a positive integer, not {nrows!r}")
     51 if not isinstance(ncols, Integral) or ncols <= 0:
     52     raise ValueError(
```

```
     53         f"Number of columns must be a positive integer, not {ncols!r}")

ValueError: Number of rows must be a positive integer, not 27.2
```

```
<Figure size 2400x17408 with 0 Axes>
```

In [17]:
```
# Scatter and density plots
def plotScatterMatrix(df1, plotSize, textSize):
    df1 = df1.select_dtypes(include =[np.number]) # keep only numerical columns
    # Remove rows and columns that would lead to df being singular
    df1 = df1.dropna('columns')
    df1 = df1[[col for col in df if df[col].nunique() > 1]] # keep columns where there
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion of k
        columnNames = columnNames[:10]
    df1 = df1[columnNames]
    ax = pd.plotting.scatter_matrix(df1, alpha=0.75, figsize=[plotSize, plotSize], dia
    corrs = df1.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axe
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

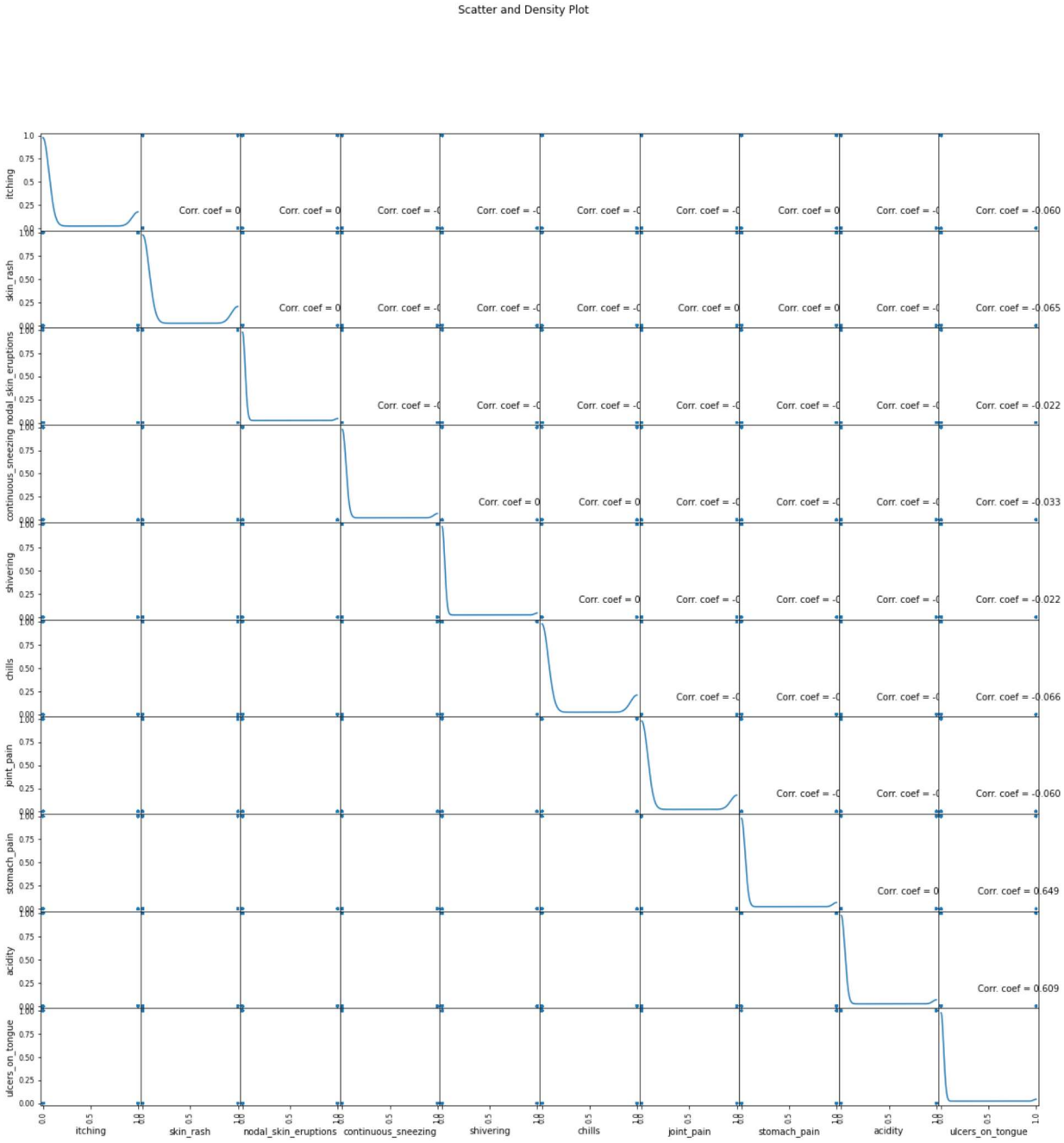
In [18]:
```
plotPerColumnDistribution(df, 10, 5)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [18], in <cell line: 1>()
----> 1 plotPerColumnDistribution(df, 10, 5)

Input In [13], in plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow)
      8 plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
facecolor = 'w', edgecolor = 'k')
      9 for i in range(min(nCol, nGraphShown)):
---> 10     plt.subplot(nGraphRow, nGraphPerRow, i + 1)
     11     columnDf = df.iloc[:, i]
     12     if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):

File ~\anaconda3\lib\site-packages\matplotlib\pyplot.py:1268, in subplot(*args, **kwa
rgs)
   1265 fig = gcf()
   1267 # First, search for an existing subplot with a matching spec.
-> 1268 key = SubplotSpec._from_subplot_args(fig, args)
   1270 for ax in fig.axes:
   1271     # if we found an axes at the position sort out if we can re-use it
   1272     if hasattr(ax, 'get_subplotspec') and ax.get_subplotspec() == key:
   1273         # if the user passed no kwargs, re-use

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:597, in SubplotSpec._from_s
ubplot_args(figure, args)
    593 else:
    594     raise TypeError(f"subplot() takes 1 or 3 positional arguments but "
    595                     f"{len(args)} were given")
--> 597 gs = GridSpec._check_gridspec_exists(figure, rows, cols)
    598 if gs is None:
    599     gs = GridSpec(rows, cols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:225, in GridSpecBase._check
_gridspec_exists(figure, nrows, ncols)
    223             return gs
    224 # else gridspec not found:
--> 225 return GridSpec(nrows, ncols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:385, in GridSpec.__init__(s
elf, nrows, ncols, figure, left, bottom, right, top, wspace, hspace, width_ratios, he
ight_ratios)
    382 self.hspace = hspace
    383 self.figure = figure
--> 385 super().__init__(nrows, ncols,
    386                  width_ratios=width_ratios,
    387                  height_ratios=height_ratios)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:49, in GridSpecBase.__init_
_(self, nrows, ncols, height_ratios, width_ratios)
     34 """
     35 Parameters
     36 ----------
   (...)
     46     If not given, all rows will have the same height.
     47 """
     48 if not isinstance(nrows, Integral) or nrows <= 0:
---> 49     raise ValueError(
     50         f"Number of rows must be a positive integer, not {nrows!r}")
     51 if not isinstance(ncols, Integral) or ncols <= 0:
     52     raise ValueError(
```

```
     53         f"Number of columns must be a positive integer, not {ncols!r}")

ValueError: Number of rows must be a positive integer, not 27.2
<Figure size 2400x17408 with 0 Axes>
```

In [19]:
```
plotScatterMatrix(df, 20, 10)
```

```
C:\Users\Sazna\AppData\Local\Temp\ipykernel_8196\1304029198.py:5: FutureWarning: In a future version of pandas all arguments of DataFrame.dropna will be keyword-only.
  df1 = df1.dropna('columns')
```



In [20]:
```
X= df[l1]
y = df[["prognosis"]]
np.ravel(y)
print(X)
```

```
      back_pain  constipation  abdominal_pain  diarrhoea  mild_fever  \
0             0             0               0          0           0
1             0             0               0          0           0
2             0             0               0          0           0
3             0             0               0          0           0
4             0             0               0          0           0
...         ...           ...             ...        ...         ...
4915          0             0               0          0           0
4916          0             0               0          0           0
4917          0             0               0          0           0
4918          0             0               0          0           0
4919          0             0               0          0           0

      yellow_urine  yellowing_of_eyes  acute_liver_failure  fluid_overload  \
0                0                  0                    0               0
1                0                  0                    0               0
2                0                  0                    0               0
3                0                  0                    0               0
4                0                  0                    0               0
...            ...                ...                  ...             ...
4915             0                  0                    0               0
4916             0                  0                    0               0
4917             0                  0                    0               0
4918             0                  0                    0               0
4919             0                  0                    0               0

      swelling_of_stomach  ...  pus_filled_pimples  blackheads  scurring  \
0                       0  ...                   0           0         0
1                       0  ...                   0           0         0
2                       0  ...                   0           0         0
3                       0  ...                   0           0         0
4                       0  ...                   0           0         0
...                   ...  ...                 ...         ...       ...
4915                    0  ...                   0           0         0
4916                    0  ...                   1           1         1
4917                    0  ...                   0           0         0
4918                    0  ...                   0           0         0
4919                    0  ...                   0           0         0

      skin_peeling  silver_like_dusting  small_dents_in_nails  \
0                0                    0                     0
1                0                    0                     0
2                0                    0                     0
3                0                    0                     0
4                0                    0                     0
...            ...                  ...                   ...
4915             0                    0                     0
4916             0                    0                     0
4917             0                    0                     0
4918             1                    1                     1
4919             0                    0                     0

      inflammatory_nails  blister  red_sore_around_nose  yellow_crust_ooze
0                      0        0                     0                  0
1                      0        0                     0                  0
2                      0        0                     0                  0
3                      0        0                     0                  0
4                      0        0                     0                  0
...                  ...      ...                   ...                ...
4915                   0        0                     0                  0
```

```
4916                    0        0                    0                    0
4917                    0        0                    0                    0
4918                    1        0                    0                    0
4919                    0        1                    1                    1

[4920 rows x 95 columns]
```

In [21]:
```
print(y)
```

```
      prognosis
0             0
1             0
2             0
3             0
4             0
...         ...
4915         36
4916         37
4917         38
4918         39
4919         40

[4920 rows x 1 columns]
```

In [22]:
```
#Reading the  testing.csv file
tr=pd.read_csv("C:/Users/Sazna/OneDrive/Desktop/testing.csv")

#Using inbuilt function replace in pandas for replacing the values

tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
tr.head()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Input In [22], in <cell line: 2>()
      1 #Reading the  testing.csv file
----> 2 tr=pd.read_csv("testing.csv")
      4 #Using inbuilt function replace in pandas for replacing the values
      6 tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic c
holestasis':3,'Drug Reaction':4,
      7     'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronc
hial Asthma':9,'Hypertension ':10,
      8     'Migraine':11,'Cervical spondylosis':12,
   (...)
     13     '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract inf
ection':38,'Psoriasis':39,
     14     'Impetigo':40}},inplace=True)

File ~\anaconda3\lib\site-packages\pandas\util\_decorators.py:311, in deprecate_nonke
yword_arguments.<locals>.decorate.<locals>.wrapper(*args, **kwargs)
    305 if len(args) > num_allow_args:
    306     warnings.warn(
    307         msg.format(arguments=arguments),
    308         FutureWarning,
    309         stacklevel=stacklevel,
    310     )
--> 311 return func(*args, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:680, in read_csv(file
path_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, m
angle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspa
ce, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip
_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirs
t, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator,
quotechar, quoting, doublequote, escapechar, comment, encoding, encoding_errors, dial
ect, error_bad_lines, warn_bad_lines, on_bad_lines, delim_whitespace, low_memory, mem
ory_map, float_precision, storage_options)
    665 kwds_defaults = _refine_defaults_read(
    666     dialect,
    667     delimiter,
   (...)
    676     defaults={"delimiter": ","},
    677 )
    678 kwds.update(kwds_defaults)
--> 680 return _read(filepath_or_buffer, kwds)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:575, in _read(filepat
h_or_buffer, kwds)
    572 _validate_names(kwds.get("names", None))
    574 # Create the parser.
--> 575 parser = TextFileReader(filepath_or_buffer, **kwds)
    577 if chunksize or iterator:
    578     return parser

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:933, in TextFileReade
r.__init__(self, f, engine, **kwds)
    930     self.options["has_index_names"] = kwds["has_index_names"]
    932 self.handles: IOHandles | None = None
--> 933 self._engine = self._make_engine(f, self.engine)

File ~\anaconda3\lib\site-packages\pandas\io\parsers\readers.py:1217, in TextFileRead
er._make_engine(self, f, engine)
```

```
   1213     mode = "rb"
   1214 # error: No overload variant of "get_handle" matches argument types
   1215 # "Union[str, PathLike[str], ReadCsvBuffer[bytes], ReadCsvBuffer[str]]"
   1216 # , "str", "bool", "Any", "Any", "Any", "Any", "Any"
-> 1217 self.handles = get_handle(  # type: ignore[call-overload]
   1218     f,
   1219     mode,
   1220     encoding=self.options.get("encoding", None),
   1221     compression=self.options.get("compression", None),
   1222     memory_map=self.options.get("memory_map", False),
   1223     is_text=is_text,
   1224     errors=self.options.get("encoding_errors", "strict"),
   1225     storage_options=self.options.get("storage_options", None),
   1226 )
   1227 assert self.handles is not None
   1228 f = self.handles.handle

File ~\anaconda3\lib\site-packages\pandas\io\common.py:789, in get_handle(path_or_bu
f, mode, encoding, compression, memory_map, is_text, errors, storage_options)
    784 elif isinstance(handle, str):
    785     # Check whether the filename is to be opened in binary mode.
    786     # Binary mode does not support 'encoding' and 'newline'.
    787     if ioargs.encoding and "b" not in ioargs.mode:
    788         # Encoding
--> 789         handle = open(
    790             handle,
    791             ioargs.mode,
    792             encoding=ioargs.encoding,
    793             errors=errors,
    794             newline="",
    795         )
    796     else:
    797         # Binary mode
    798         handle = open(handle, ioargs.mode)

FileNotFoundError: [Errno 2] No such file or directory: 'testing.csv'
```

In [23]:

```
#Reading the  testing.csv file
tr=pd.read_csv("C:/Users/Sazna/OneDrive/Desktop/testing.csv")

#Using inbuilt function replace in pandas for replacing the values

tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasi
    'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asth
    'Migraine':11,'Cervical spondylosis':12,
    'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Der
    'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hep
    'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':2
    'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,
    '(vertigo) Paroymsal  Positional Vertigo':36,'Acne':37,'Urinary tract infection':3
    'Impetigo':40}},inplace=True)
tr.head()
```

Out[23]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach
0	1	1	1	0	0	0	0	
1	0	0	0	1	1	1	0	
2	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	
4	1	1	0	0	0	0	0	

5 rows × 133 columns

In [25]:
```
plotPerColumnDistribution(tr, 10, 6)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [25], in <cell line: 1>()
----> 1 plotPerColumnDistribution(tr, 10, 6)

Input In [13], in plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow)
      8 plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
facecolor = 'w', edgecolor = 'k')
      9 for i in range(min(nCol, nGraphShown)):
---> 10     plt.subplot(nGraphRow, nGraphPerRow, i + 1)
     11     columnDf = df.iloc[:, i]
     12     if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):

File ~\anaconda3\lib\site-packages\matplotlib\pyplot.py:1268, in subplot(*args, **kwa
rgs)
   1265 fig = gcf()
   1267 # First, search for an existing subplot with a matching spec.
-> 1268 key = SubplotSpec._from_subplot_args(fig, args)
   1270 for ax in fig.axes:
   1271     # if we found an axes at the position sort out if we can re-use it
   1272     if hasattr(ax, 'get_subplotspec') and ax.get_subplotspec() == key:
   1273         # if the user passed no kwargs, re-use

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:597, in SubplotSpec._from_s
ubplot_args(figure, args)
    593 else:
    594     raise TypeError(f"subplot() takes 1 or 3 positional arguments but "
    595                     f"{len(args)} were given")
--> 597 gs = GridSpec._check_gridspec_exists(figure, rows, cols)
    598 if gs is None:
    599     gs = GridSpec(rows, cols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:225, in GridSpecBase._check
_gridspec_exists(figure, nrows, ncols)
    223             return gs
    224 # else gridspec not found:
--> 225 return GridSpec(nrows, ncols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:385, in GridSpec.__init__(s
elf, nrows, ncols, figure, left, bottom, right, top, wspace, hspace, width_ratios, he
ight_ratios)
    382 self.hspace = hspace
    383 self.figure = figure
--> 385 super().__init__(nrows, ncols,
    386                  width_ratios=width_ratios,
    387                  height_ratios=height_ratios)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:49, in GridSpecBase.__init_
_(self, nrows, ncols, height_ratios, width_ratios)
     34 """
     35 Parameters
     36 ----------
   (...)
     46     If not given, all rows will have the same height.
     47 """
     48 if not isinstance(nrows, Integral) or nrows <= 0:
---> 49     raise ValueError(
     50         f"Number of rows must be a positive integer, not {nrows!r}")
     51 if not isinstance(ncols, Integral) or ncols <= 0:
     52     raise ValueError(
```

```
     53         f"Number of columns must be a positive integer, not {ncols!r}")

ValueError: Number of rows must be a positive integer, not 22.833333333333332
```

<Figure size 2880x14613.3 with 0 Axes>

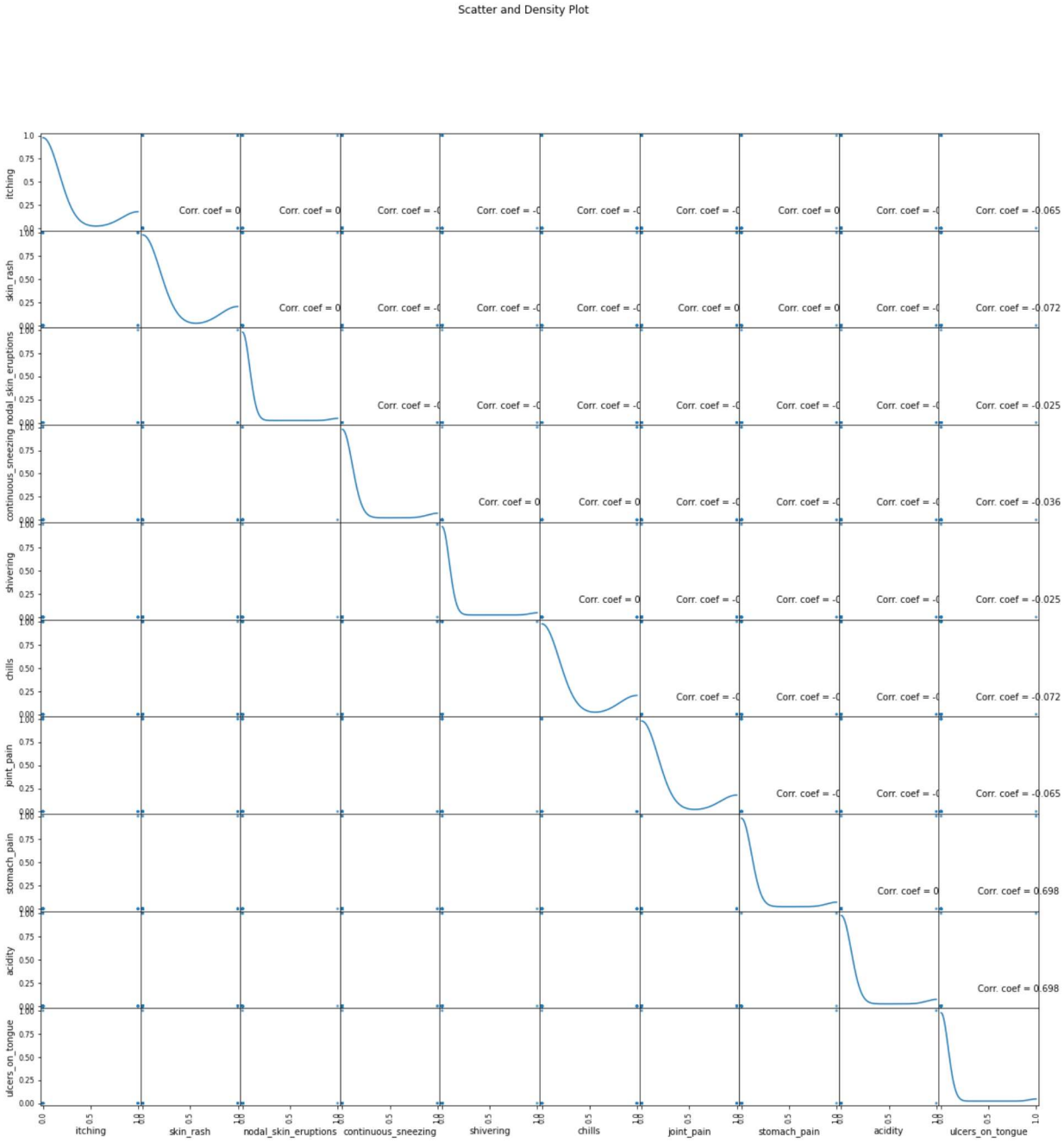
In [26]:
```
plotPerColumnDistribution(tr, 10, 6)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Input In [26], in <cell line: 1>()
----> 1 plotPerColumnDistribution(tr, 10, 6)

Input In [13], in plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow)
      8 plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80,
facecolor = 'w', edgecolor = 'k')
      9 for i in range(min(nCol, nGraphShown)):
---> 10     plt.subplot(nGraphRow, nGraphPerRow, i + 1)
     11     columnDf = df.iloc[:, i]
     12     if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):

File ~\anaconda3\lib\site-packages\matplotlib\pyplot.py:1268, in subplot(*args, **kwa
rgs)
   1265 fig = gcf()
   1267 # First, search for an existing subplot with a matching spec.
-> 1268 key = SubplotSpec._from_subplot_args(fig, args)
   1270 for ax in fig.axes:
   1271     # if we found an axes at the position sort out if we can re-use it
   1272     if hasattr(ax, 'get_subplotspec') and ax.get_subplotspec() == key:
   1273         # if the user passed no kwargs, re-use

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:597, in SubplotSpec._from_s
ubplot_args(figure, args)
    593 else:
    594     raise TypeError(f"subplot() takes 1 or 3 positional arguments but "
    595                     f"{len(args)} were given")
--> 597 gs = GridSpec._check_gridspec_exists(figure, rows, cols)
    598 if gs is None:
    599     gs = GridSpec(rows, cols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:225, in GridSpecBase._check
_gridspec_exists(figure, nrows, ncols)
    223             return gs
    224 # else gridspec not found:
--> 225 return GridSpec(nrows, ncols, figure=figure)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:385, in GridSpec.__init__(s
elf, nrows, ncols, figure, left, bottom, right, top, wspace, hspace, width_ratios, he
ight_ratios)
    382 self.hspace = hspace
    383 self.figure = figure
--> 385 super().__init__(nrows, ncols,
    386                  width_ratios=width_ratios,
    387                  height_ratios=height_ratios)

File ~\anaconda3\lib\site-packages\matplotlib\gridspec.py:49, in GridSpecBase.__init_
_(self, nrows, ncols, height_ratios, width_ratios)
     34 """
     35 Parameters
     36 ----------
   (...)
     46     If not given, all rows will have the same height.
     47 """
     48 if not isinstance(nrows, Integral) or nrows <= 0:
---> 49     raise ValueError(
     50         f"Number of rows must be a positive integer, not {nrows!r}")
     51 if not isinstance(ncols, Integral) or ncols <= 0:
     52     raise ValueError(
```

```
     53         f"Number of columns must be a positive integer, not {ncols!r}")

ValueError: Number of rows must be a positive integer, not 22.833333333333332
<Figure size 2880x14613.3 with 0 Axes>
```

In [27]:
```
plotScatterMatrix(tr, 20, 10)
```

```
C:\Users\Sazna\AppData\Local\Temp\ipykernel_8196\1304029198.py:5: FutureWarning: In a future version of pandas all arguments of DataFrame.dropna will be keyword-only.
  df1 = df1.dropna('columns')
```



In [28]:
```
X_test= tr[l1]
y_test = tr[["prognosis"]]
np.ravel(y_test)
print(X_test)
```

	back_pain	constipation	abdominal_pain	diarrhoea	mild_fever
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	1	0	0
4	0	0	0	0	0
5	0	0	1	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	1	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	1	0	0	0	0
13	0	0	0	0	0
14	0	0	1	0	0
15	0	0	0	1	0
16	0	0	0	0	1
17	1	0	0	0	0
18	0	1	1	1	0
19	0	0	1	1	1
20	0	0	1	0	0
21	0	0	0	0	0
22	0	0	1	0	0
23	0	0	1	0	0
24	0	0	1	0	0
25	0	0	0	0	1
26	0	0	0	0	0
27	0	0	0	0	0
28	0	1	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	0	0	0	0	0
32	0	0	0	1	0
33	0	0	0	0	0
34	0	0	0	0	0
35	0	0	0	0	0
36	0	0	0	0	0
37	0	0	0	0	0
38	0	0	0	0	0
39	0	0	0	0	0
40	0	0	0	0	0

	yellow_urine	yellowing_of_eyes	acute_liver_failure	fluid_overload
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	1	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0

16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	1	0	0
20	1	1	0	0
21	0	1	0	0
22	0	1	0	0
23	0	1	1	0
24	0	0	0	0
25	0	1	0	0
26	0	0	0	0
27	0	0	0	0
28	0	0	0	0
29	0	0	0	0
30	0	0	0	0
31	0	0	0	0
32	0	0	0	0
33	0	0	0	0
34	0	0	0	0
35	0	0	0	0
36	0	0	0	0
37	0	0	0	0
38	0	0	0	0
39	0	0	0	0
40	0	0	0	0

	swelling_of_stomach	...	pus_filled_pimples	blackheads	scurring	\
0	0	...	0	0	0	
1	0	...	0	0	0	
2	0	...	0	0	0	
3	0	...	0	0	0	
4	0	...	0	0	0	
5	0	...	0	0	0	
6	0	...	0	0	0	
7	0	...	0	0	0	
8	0	...	0	0	0	
9	0	...	0	0	0	
10	0	...	0	0	0	
11	0	...	0	0	0	
12	0	...	0	0	0	
13	0	...	0	0	0	
14	0	...	0	0	0	
15	0	...	0	0	0	
16	0	...	0	0	0	
17	0	...	0	0	0	
18	0	...	0	0	0	
19	0	...	0	0	0	
20	0	...	0	0	0	
21	0	...	0	0	0	
22	0	...	0	0	0	
23	0	...	0	0	0	
24	1	...	0	0	0	
25	0	...	0	0	0	
26	0	...	0	0	0	
27	0	...	0	0	0	
28	0	...	0	0	0	
29	0	...	0	0	0	
30	0	...	0	0	0	
31	0	...	0	0	0	
32	0	...	0	0	0	

```
33                   0  ...                   0          0         0
34                   0  ...                   0          0         0
35                   0  ...                   0          0         0
36                   0  ...                   0          0         0
37                   0  ...                   1          1         1
38                   0  ...                   0          0         0
39                   0  ...                   0          0         0
40                   0  ...                   0          0         0

    skin_peeling  silver_like_dusting  small_dents_in_nails  \
0              0                    0                     0
1              0                    0                     0
2              0                    0                     0
3              0                    0                     0
4              0                    0                     0
5              0                    0                     0
6              0                    0                     0
7              0                    0                     0
8              0                    0                     0
9              0                    0                     0
10             0                    0                     0
11             0                    0                     0
12             0                    0                     0
13             0                    0                     0
14             0                    0                     0
15             0                    0                     0
16             0                    0                     0
17             0                    0                     0
18             0                    0                     0
19             0                    0                     0
20             0                    0                     0
21             0                    0                     0
22             0                    0                     0
23             0                    0                     0
24             0                    0                     0
25             0                    0                     0
26             0                    0                     0
27             0                    0                     0
28             0                    0                     0
29             0                    0                     0
30             0                    0                     0
31             0                    0                     0
32             0                    0                     0
33             0                    0                     0
34             0                    0                     0
35             0                    0                     0
36             0                    0                     0
37             0                    0                     0
38             0                    0                     0
39             1                    1                     1
40             0                    0                     0

    inflammatory_nails  blister  red_sore_around_nose  yellow_crust_ooze
0                    0        0                     0                  0
1                    0        0                     0                  0
2                    0        0                     0                  0
3                    0        0                     0                  0
4                    0        0                     0                  0
5                    0        0                     0                  0
6                    0        0                     0                  0
```

7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
27	0	0	0	0
28	0	0	0	0
29	0	0	0	0
30	0	0	0	0
31	0	0	0	0
32	0	0	0	0
33	0	0	0	0
34	0	0	0	0
35	0	0	0	0
36	0	0	0	0
37	0	0	0	0
38	0	0	0	0
39	1	0	0	0
40	0	1	1	1

[41 rows x 95 columns]

In [29]:
```
print(y_test)
```

	prognosis
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40

In [30]:
```
#list1 = DF['prognosis'].unique()
def scatterplt(disea):
    x = ((DF.loc[disea]).sum())#total sum of symptom reported for given disease
    x.drop(x[x==0].index,inplace=True)#droping symptoms with values 0
    print(x.values)
    y = x.keys()#storing nameof symptoms in y
    print(len(x))
    print(len(y))
    plt.title(disea)
    plt.scatter(y,x.values)
    plt.show()

def scatterinp(sym1,sym2,sym3,sym4,sym5):
    x = [sym1,sym2,sym3,sym4,sym5]#storing input symptoms in y
    y = [0,0,0,0,0]#creating and giving values to the input symptoms
    if(sym1!='Select Here'):
        y[0]=1
```

```
    if(sym2!='Select Here'):
        y[1]=1
    if(sym3!='Select Here'):
        y[2]=1
    if(sym4!='Select Here'):
        y[3]=1
    if(sym5!='Select Here'):
        y[4]=1
    print(x)
    print(y)
    plt.scatter(x,y)
    plt.show()
```

In [33]:

```
root = Tk()
pred1=StringVar()
def DecisionTree():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn import tree

        clf3 = tree.DecisionTreeClassifier()
        clf3 = clf3.fit(X,y)

        from sklearn.metrics import classification_report,confusion_matrix,accuracy_sc
        y_pred=clf3.predict(X_test)
        print("Decision Tree")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Sympt

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = clf3.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
```

```
        if (h=='yes'):
            pred1.set(" ")
            pred1.set(disease[a])
        else:
            pred1.set(" ")
            pred1.set("Not Found")
        #Creating the database if not exists named as database.db and creating table i
        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS DecisionTree(Name StringVar,Symtom1 Stri
        c.execute("INSERT INTO DecisionTree(Name,Symtom1,Symtom2,Symtom3,Symtom4,Symtc
        conn.commit()
        c.close()
        conn.close()

        #printing scatter plot of input symptoms
        #printing scatter plot of disease predicted vs its symptoms
        scatterinp(Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Sympton
        scatterplt(pred1.get())
```

In [34]:

```
pred2=StringVar()
def randomforest():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.ensemble import RandomForestClassifier
        clf4 = RandomForestClassifier(n_estimators=100)
        clf4 = clf4.fit(X,np.ravel(y))

        # calculating accuracy
        from sklearn.metrics import classification_report,confusion_matrix,accuracy_sc
        y_pred=clf4.predict(X_test)
        print("Random Forest")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Sympt

        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
        predict = clf4.predict(inputtest)
```

```
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
        if (h=='yes'):
            pred2.set(" ")
            pred2.set(disease[a])
        else:
            pred2.set(" ")
            pred2.set("Not Found")
         #Creating the database if not exists named as database.db and creating table
        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS RandomForest(Name StringVar,Symtom1 Stri
        c.execute("INSERT INTO RandomForest(Name,Symtom1,Symtom2,Symtom3,Symtom4,Symtc
        conn.commit()
        c.close()
        conn.close()
        #printing scatter plot of disease predicted vs its symptoms
        scatterplt(pred2.get())
```

In [35]:

```
pred3=StringVar()
def NaiveBayes():
    if len(NameEn.get()) == 0:
        pred1.set(" ")
        comp=messagebox.askokcancel("System","Kindly Fill the Name")
        if comp:
            root.mainloop()
    elif((Symptom1.get()=="Select Here") or (Symptom2.get()=="Select Here")):
        pred1.set(" ")
        sym=messagebox.askokcancel("System","Kindly Fill atleast first two Symptoms")
        if sym:
            root.mainloop()
    else:
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb=gnb.fit(X,np.ravel(y))

        from sklearn.metrics import classification_report,confusion_matrix,accuracy_sc
        y_pred=gnb.predict(X_test)
        print("Naive Bayes")
        print("Accuracy")
        print(accuracy_score(y_test, y_pred))
        print(accuracy_score(y_test, y_pred,normalize=False))
        print("Confusion matrix")
        conf_matrix=confusion_matrix(y_test,y_pred)
        print(conf_matrix)

        psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Sympt
        for k in range(0,len(l1)):
            for z in psymptoms:
                if(z==l1[k]):
                    l2[k]=1

        inputtest = [l2]
```

```
        predict = gnb.predict(inputtest)
        predicted=predict[0]

        h='no'
        for a in range(0,len(disease)):
            if(predicted == a):
                h='yes'
                break
        if (h=='yes'):
            pred3.set(" ")
            pred3.set(disease[a])
        else:
            pred3.set(" ")
            pred3.set("Not Found")
         #Creating the database if not exists named as database.db and creating table
        import sqlite3
        conn = sqlite3.connect('database.db')
        c = conn.cursor()
        c.execute("CREATE TABLE IF NOT EXISTS NaiveBayes(Name StringVar,Symtom1 String
        c.execute("INSERT INTO NaiveBayes(Name,Symtom1,Symtom2,Symtom3,Symtom4,Symtom5
        conn.commit()
        c.close()
        conn.close()
        #printing scatter plot of disease predicted vs its symptoms
        scatterplt(pred3.get())
```

In [36]:
```
#Tk class is used to create a root window
root.configure(background='Ivory')
root.title('Smart Disease Predictor System')
root.resizable(0,0)
```

Out[36]:
```
''
```

In [37]:
```
Symptom1 = StringVar()
Symptom1.set("Select Here")

Symptom2 = StringVar()
Symptom2.set("Select Here")

Symptom3 = StringVar()
Symptom3.set("Select Here")

Symptom4 = StringVar()
Symptom4.set("Select Here")

Symptom5 = StringVar()
Symptom5.set("Select Here")
Name = StringVar()
```

In [38]:
```
prev_win=None
def Reset():
    global prev_win

    Symptom1.set("Select Here")
    Symptom2.set("Select Here")
    Symptom3.set("Select Here")
    Symptom4.set("Select Here")
    Symptom5.set("Select Here")
    NameEn.delete(first=0,last=100)
```

```
    pred1.set(" ")
    pred2.set(" ")
    pred3.set(" ")
    pred4.set(" ")
    try:
        prev_win.destroy()
        prev_win=None
    except AttributeError:
        pass
```

In [39]:
```
prev_win=None
def Reset():
    global prev_win

    Symptom1.set("Select Here")
    Symptom2.set("Select Here")
    Symptom3.set("Select Here")
    Symptom4.set("Select Here")
    Symptom5.set("Select Here")
    NameEn.delete(first=0,last=100)
    pred1.set(" ")
    pred2.set(" ")
    pred3.set(" ")
    pred4.set(" ")
    try:
        prev_win.destroy()
        prev_win=None
    except AttributeError:
        pass
```

In [40]:
```
from tkinter import messagebox
def Exit():
    qExit=messagebox.askyesno("System","Do you want to exit the system")

    if qExit:
        root.destroy()
        exit()
```

In [41]:
```
#Headings for the GUI written at the top of GUI
w2 = Label(root, justify=LEFT, text="Disease Predictor using Machine Learning", fg="Re
w2.config(font=("Times",30,"bold italic"))
w2.grid(row=1, column=0, columnspan=2, padx=100)
w2 = Label(root, justify=LEFT, text="Contributors: Sudhanshu,Rohan,Aditya", fg="Pink",
w2.config(font=("Times",30,"bold italic"))
w2.grid(row=2, column=0, columnspan=2, padx=100)
```

In [42]:
```
#Label for the name
NameLb = Label(root, text="Name of the Patient *", fg="Red", bg="Ivory")
NameLb.config(font=("Times",15,"bold italic"))
NameLb.grid(row=6, column=0, pady=15, sticky=W)
```

In [43]:
```
#Creating Labels for the symtoms
S1Lb = Label(root, text="Symptom 1 *", fg="Black", bg="Ivory")
S1Lb.config(font=("Times",15,"bold italic"))
S1Lb.grid(row=7, column=0, pady=10, sticky=W)

S2Lb = Label(root, text="Symptom 2 *", fg="Black", bg="Ivory")
S2Lb.config(font=("Times",15,"bold italic"))
```

```
S2Lb.grid(row=8, column=0, pady=10, sticky=W)

S3Lb = Label(root, text="Symptom 3", fg="Black",bg="Ivory")
S3Lb.config(font=("Times",15,"bold italic"))
S3Lb.grid(row=9, column=0, pady=10, sticky=W)

S4Lb = Label(root, text="Symptom 4", fg="Black", bg="Ivory")
S4Lb.config(font=("Times",15,"bold italic"))
S4Lb.grid(row=10, column=0, pady=10, sticky=W)

S5Lb = Label(root, text="Symptom 5", fg="Black", bg="Ivory")
S5Lb.config(font=("Times",15,"bold italic"))
S5Lb.grid(row=11, column=0, pady=10, sticky=W)
```

In [44]:
```
#Labels for the different algorithms
lrLb = Label(root, text="DecisionTree", fg="white", bg="red", width = 20)
lrLb.config(font=("Times",15,"bold italic"))
lrLb.grid(row=15, column=0, pady=10,sticky=W)

destreeLb = Label(root, text="RandomForest", fg="Red", bg="Orange", width = 20)
destreeLb.config(font=("Times",15,"bold italic"))
destreeLb.grid(row=17, column=0, pady=10, sticky=W)

ranfLb = Label(root, text="NaiveBayes", fg="White", bg="green", width = 20)
ranfLb.config(font=("Times",15,"bold italic"))
ranfLb.grid(row=19, column=0, pady=10, sticky=W)

knnLb = Label(root, text="kNearestNeighbour", fg="Red", bg="Sky Blue", width = 20)
knnLb.config(font=("Times",15,"bold italic"))
knnLb.grid(row=21, column=0, pady=10, sticky=W)
OPTIONS = sorted(l1)
```

In [45]:
```
#Taking name as input from user
NameEn = Entry(root, textvariable=Name)
NameEn.grid(row=6, column=1)

#Taking Symptoms as input from the dropdown from the user
S1 = OptionMenu(root, Symptom1,*OPTIONS)
S1.grid(row=7, column=1)

S2 = OptionMenu(root, Symptom2,*OPTIONS)
S2.grid(row=8, column=1)

S3 = OptionMenu(root, Symptom3,*OPTIONS)
S3.grid(row=9, column=1)

S4 = OptionMenu(root, Symptom4,*OPTIONS)
S4.grid(row=10, column=1)

S5 = OptionMenu(root, Symptom5,*OPTIONS)
S5.grid(row=11, column=1)
```

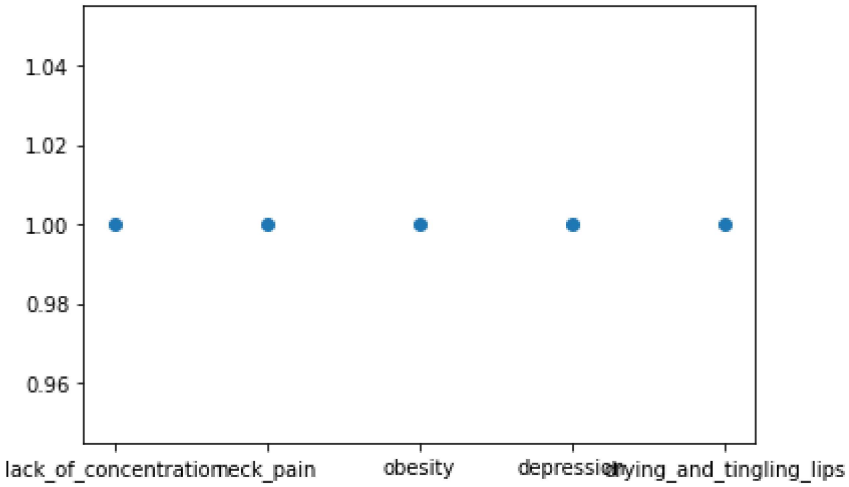
In [47]:
```
#Buttons for predicting the disease using different algorithms
dst = Button(root, text="Prediction 1", command=DecisionTree,bg="Red",fg="yellow")
dst.config(font=("Times",15,"bold italic"))
dst.grid(row=6, column=3,padx=10)

rnf = Button(root, text="Prediction 2", command=randomforest,bg="Light green",fg="red'
rnf.config(font=("Times",15,"bold italic"))
```

```
rnf.grid(row=7, column=3,padx=10)

lr = Button(root, text="Prediction 3", command=NaiveBayes,bg="Blue",fg="white")
lr.config(font=("Times",15,"bold italic"))
lr.grid(row=8, column=3,padx=10)

rs = Button(root,text="Reset Inputs", command=Reset,bg="yellow",fg="purple",width=15)
rs.config(font=("Times",15,"bold italic"))
rs.grid(row=10,column=3,padx=10)

ex = Button(root,text="Exit System", command=Exit,bg="yellow",fg="purple",width=15)
ex.config(font=("Times",15,"bold italic"))
ex.grid(row=11,column=3,padx=10)
```

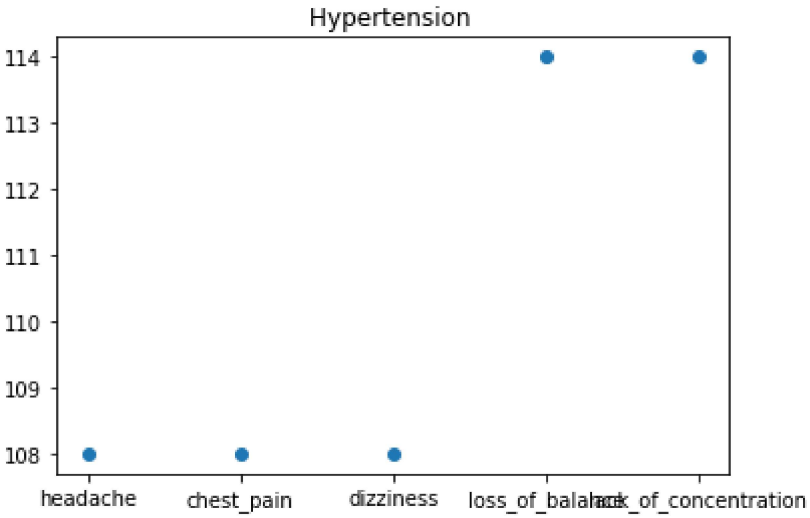
In [48]:
```
#Showing the output of different aldorithms
t1=Label(root,font=("Times",15,"bold italic"),text="Decision Tree",height=1,bg="Light
         ,width=40,fg="red",textvariable=pred1,relief="sunken").grid(row=15, column=1,

t2=Label(root,font=("Times",15,"bold italic"),text="Random Forest",height=1,bg="Purple
         ,width=40,fg="white",textvariable=pred2,relief="sunken").grid(row=17, column=

t3=Label(root,font=("Times",15,"bold italic"),text="Naive Bayes",height=1,bg="red"
         ,width=40,fg="orange",textvariable=pred3,relief="sunken").grid(row=19, columr
```

In [49]:
```
#calling this function because the application is ready to run
root.mainloop()
```

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'neck_pain', 'obesity', 'depression', 'drying_and_tingling_
lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

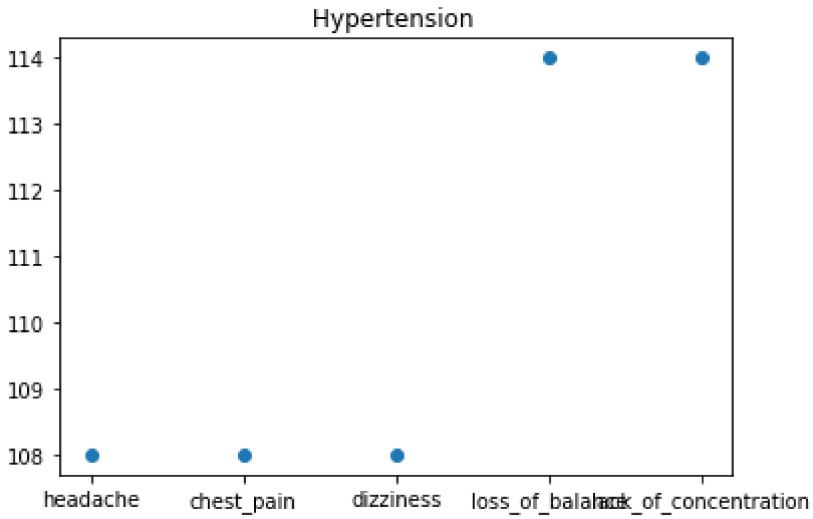



```
[108 108 108 114 114]
5
5
```



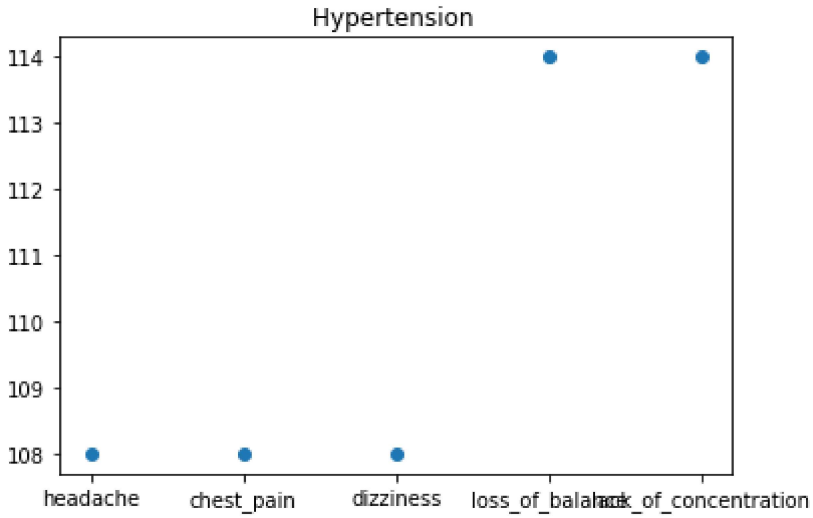

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

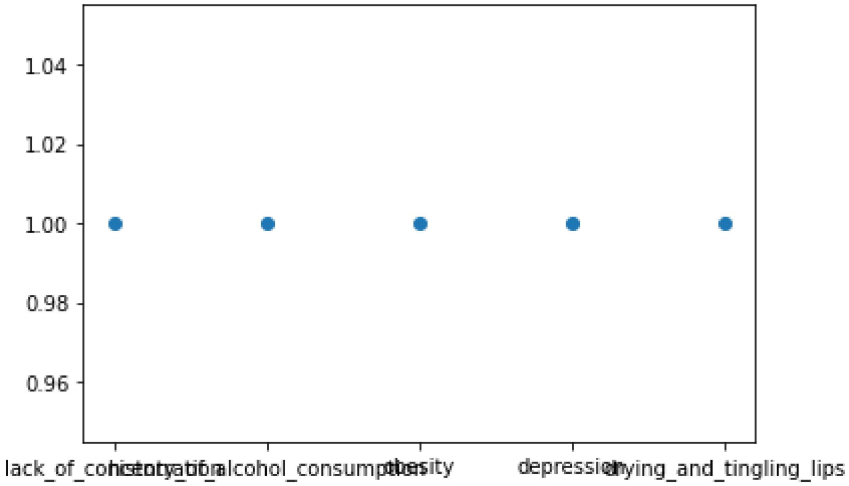



```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

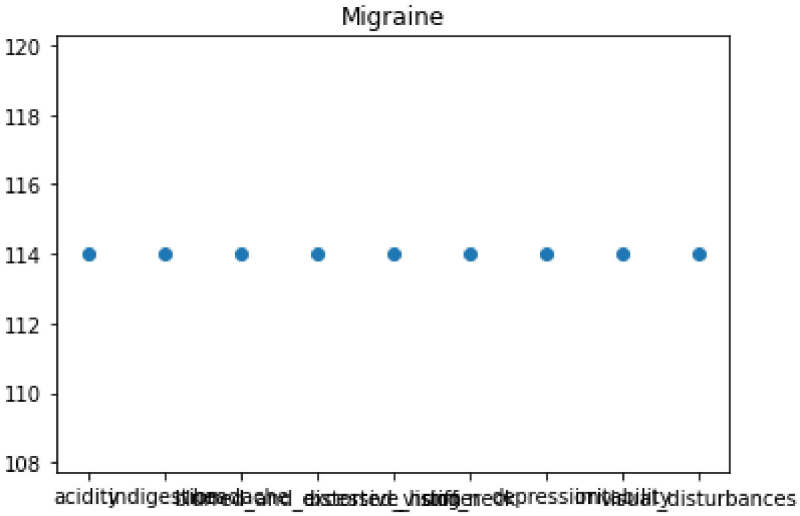
```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

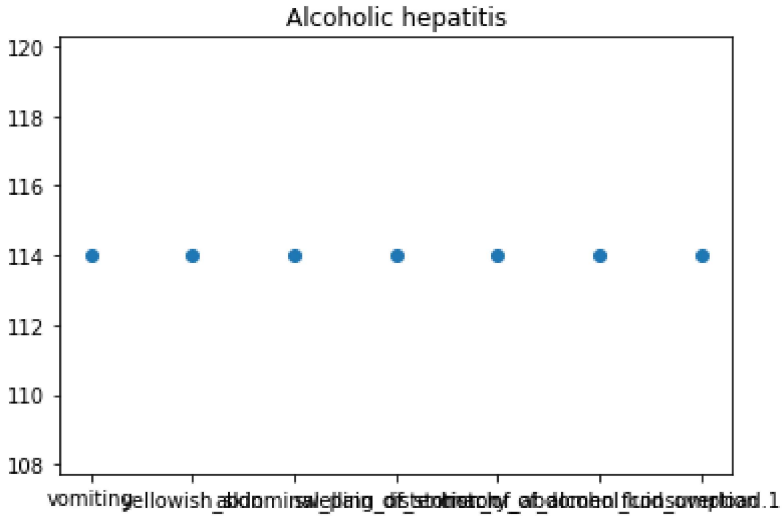



```
[114 114 114 114 114 114 114 114 114]
9
9
```



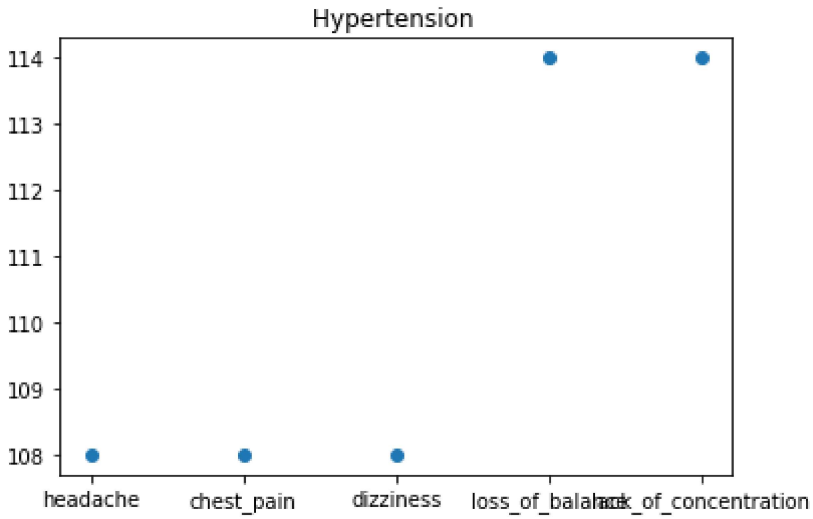
```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114 114]
7
7
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



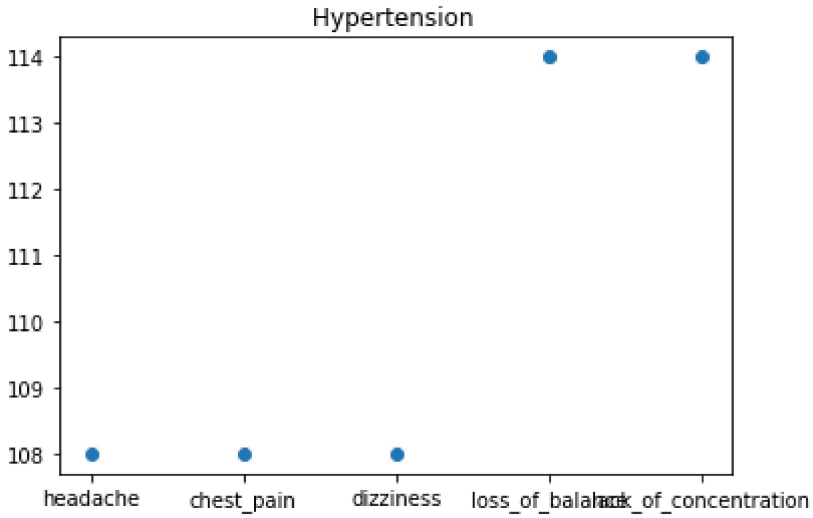

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



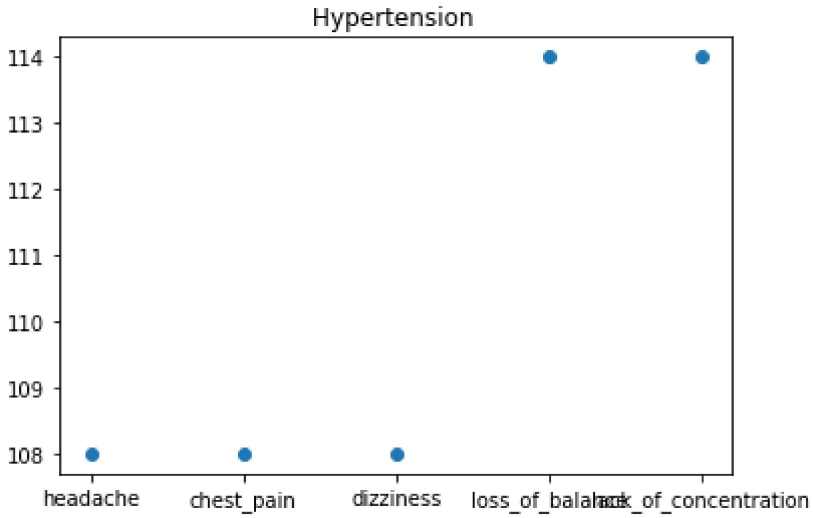

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



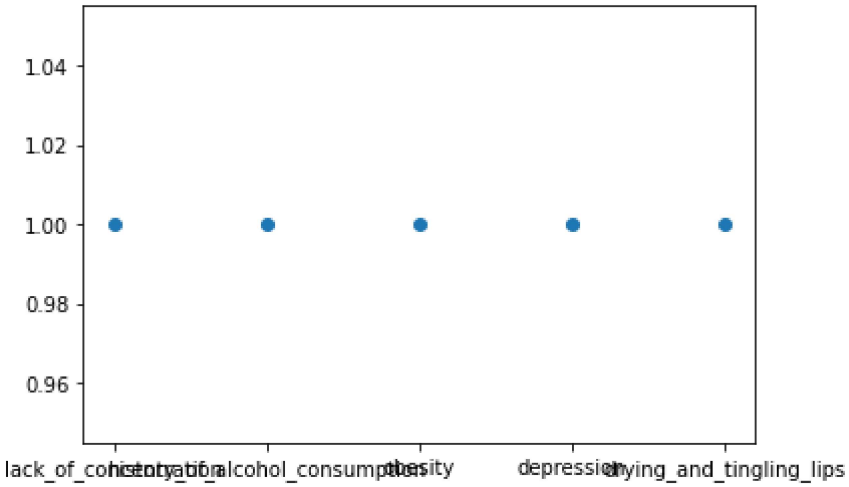
```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```

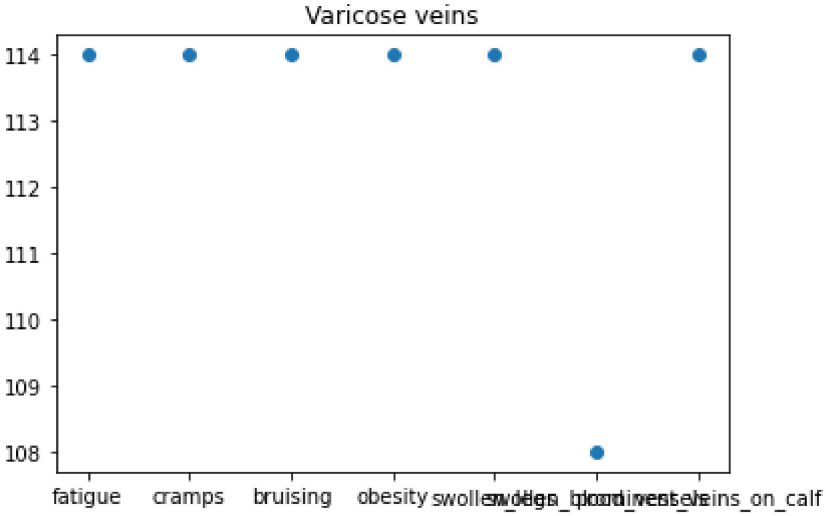



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
```

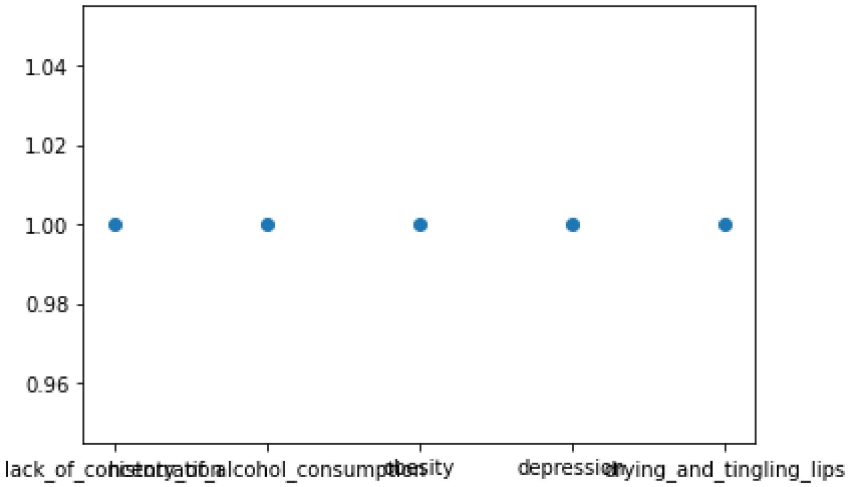
```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



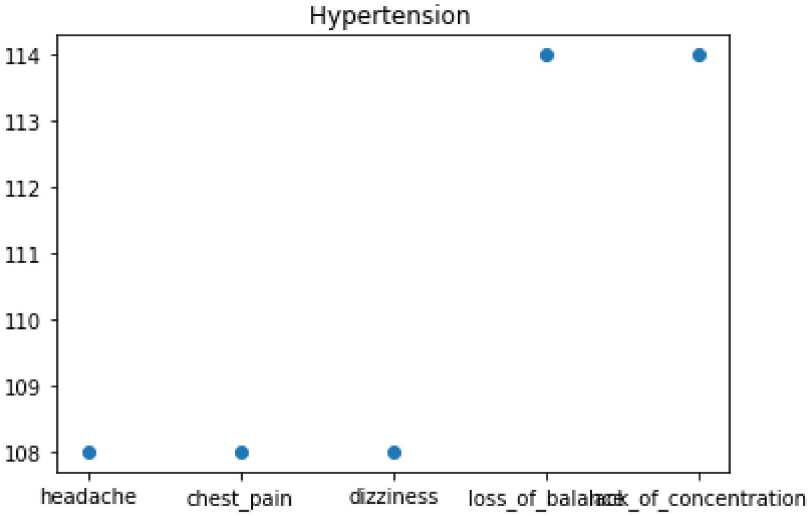

```
[114 114 114 114 114 108 114]
7
7
```



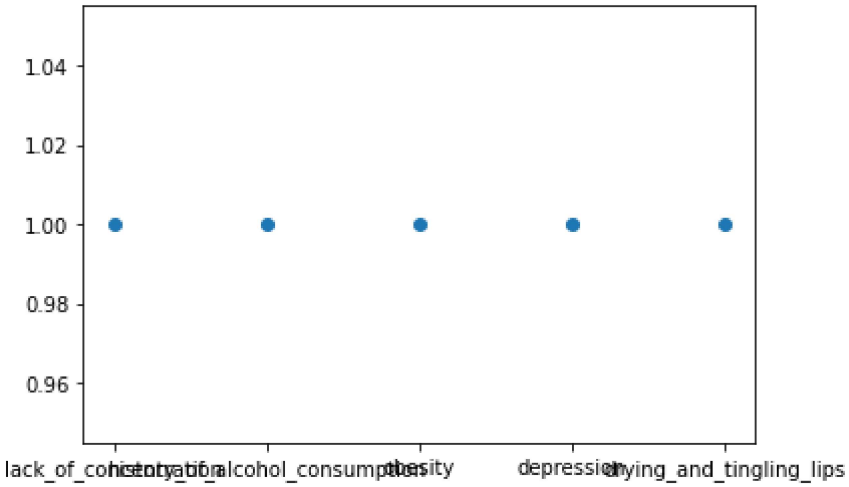

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



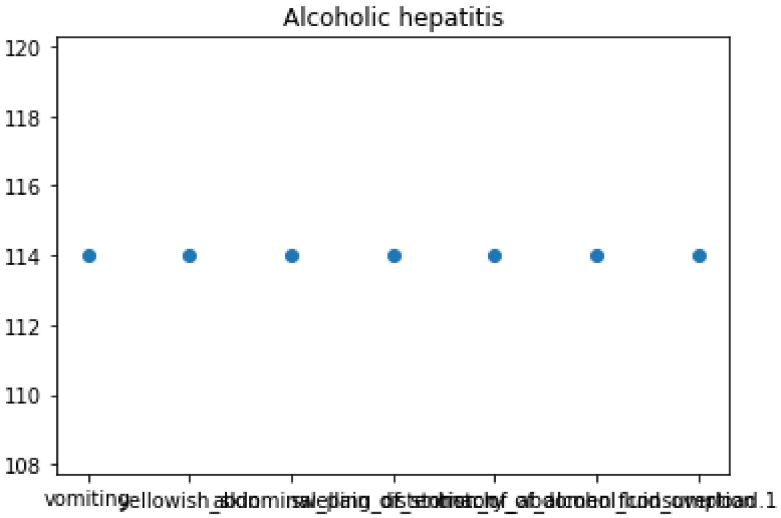

```
[108 108 108 114 114]
5
5
```




```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

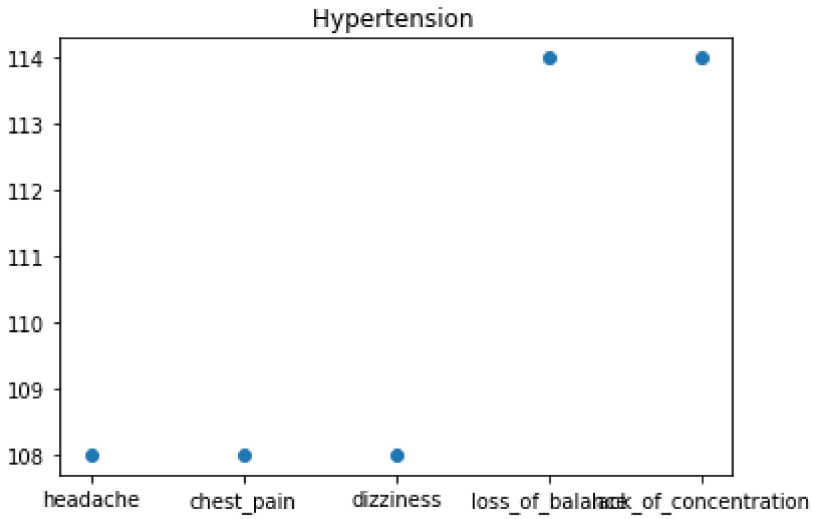



```
[114 114 114 114 114 114 114]
7
7
```



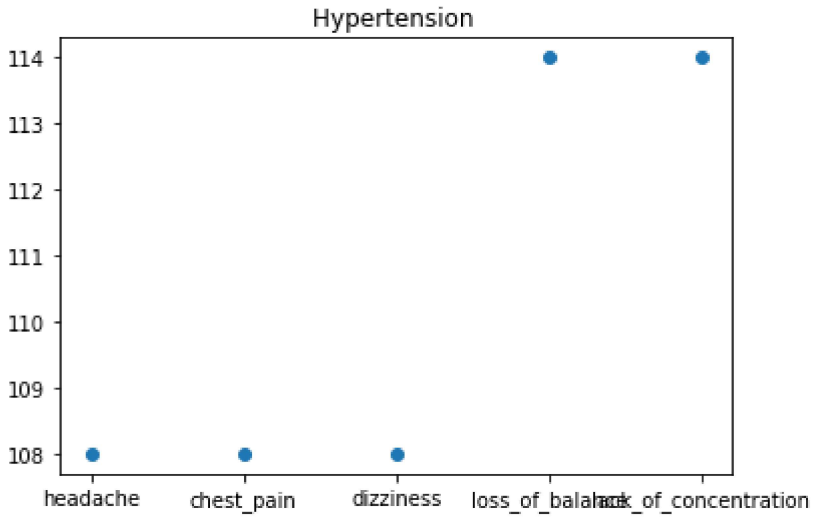

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



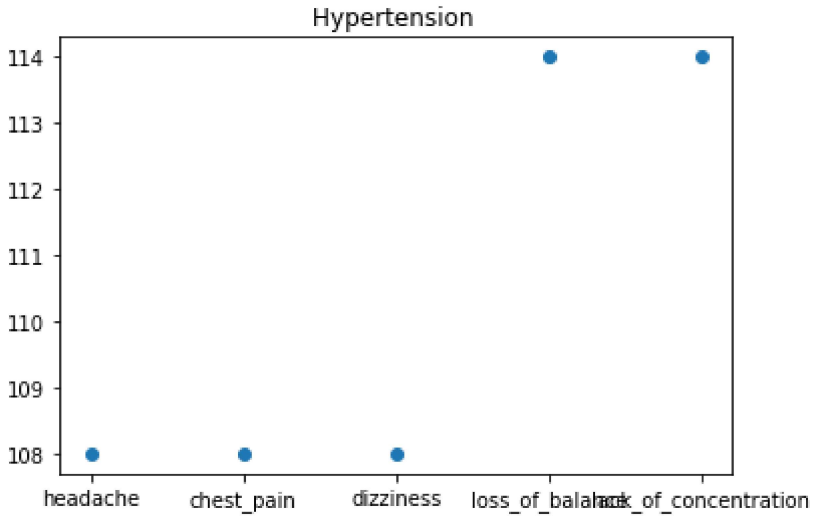

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



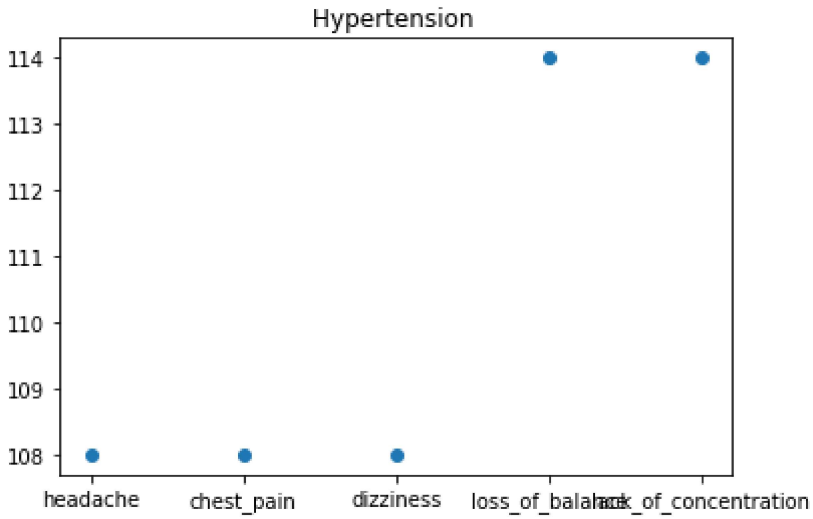
```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



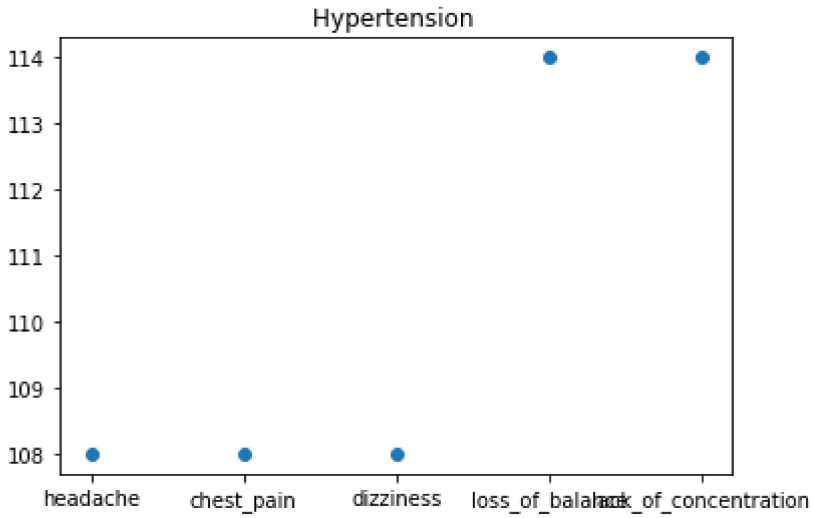

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



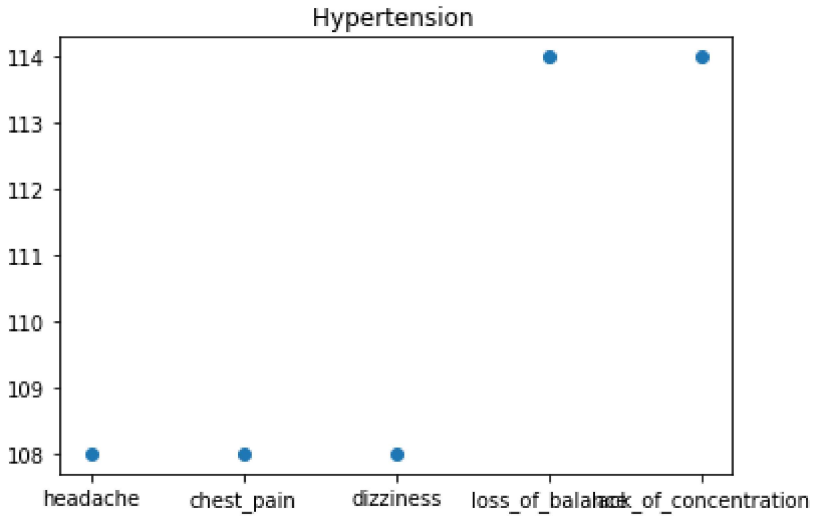

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



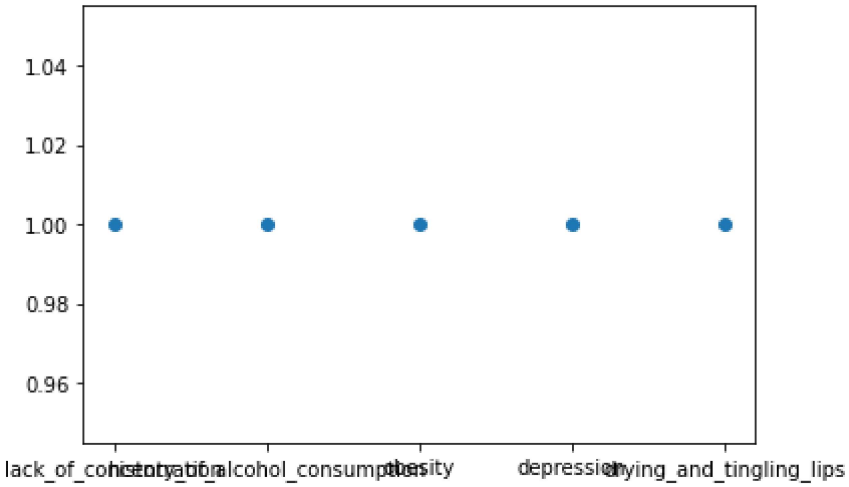
```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```

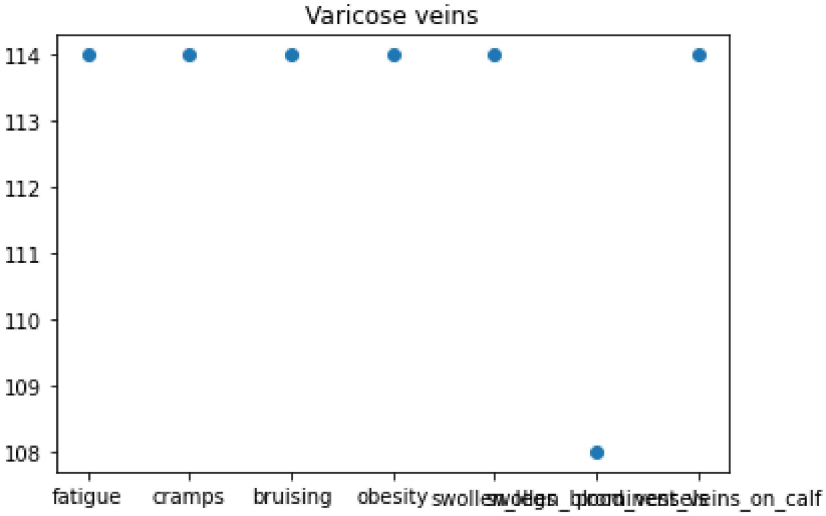



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
```

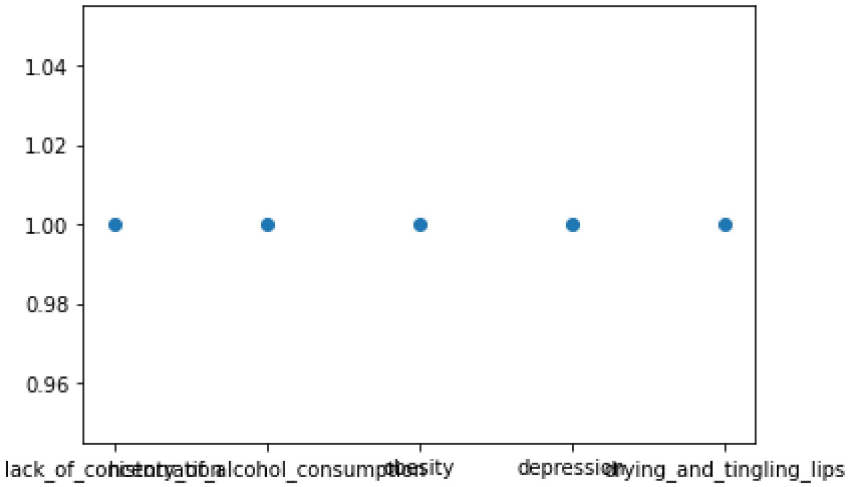
```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



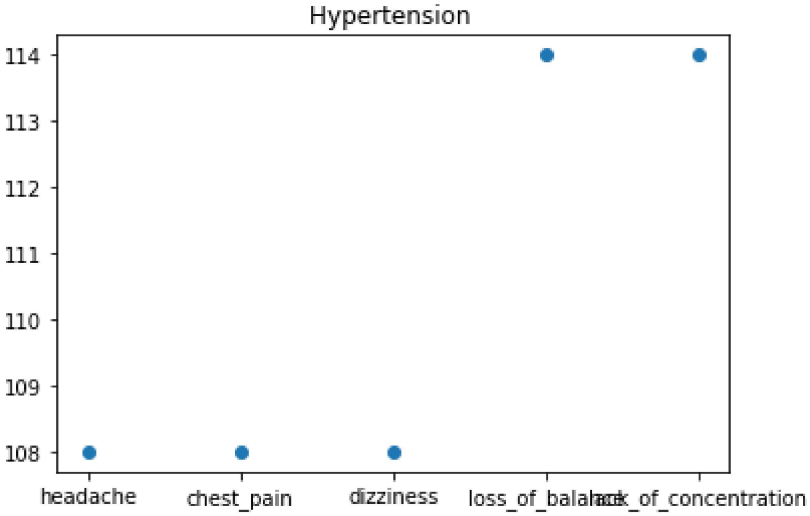

```
[114 114 114 114 114 108 114]
7
7
```



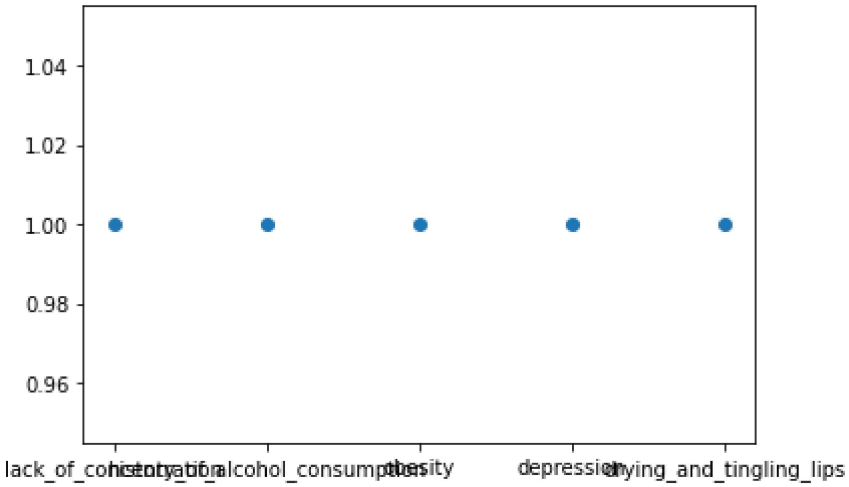

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



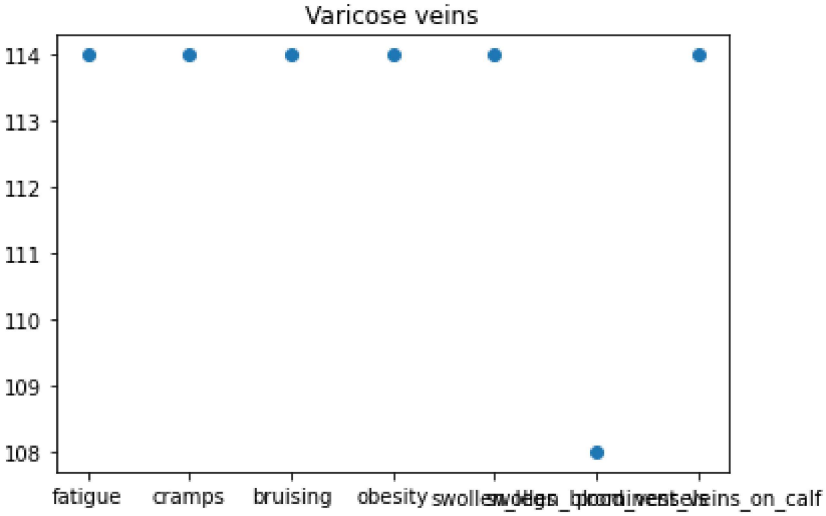

```
[108 108 108 114 114]
5
5
```



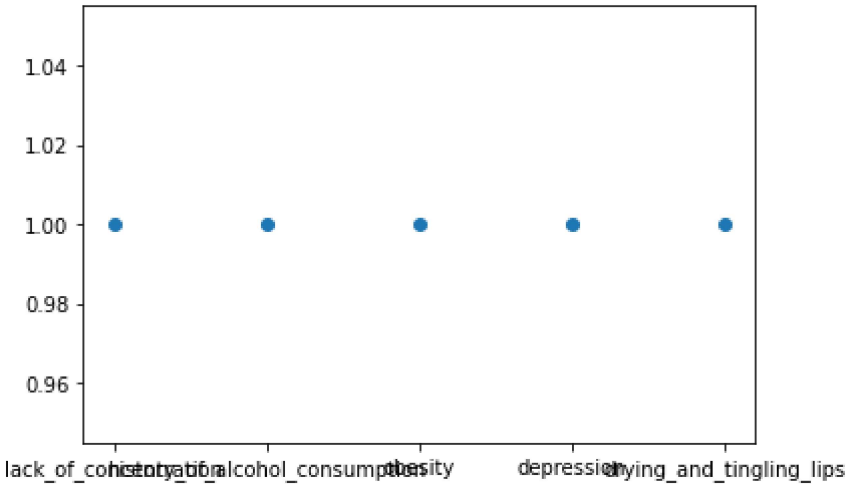

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



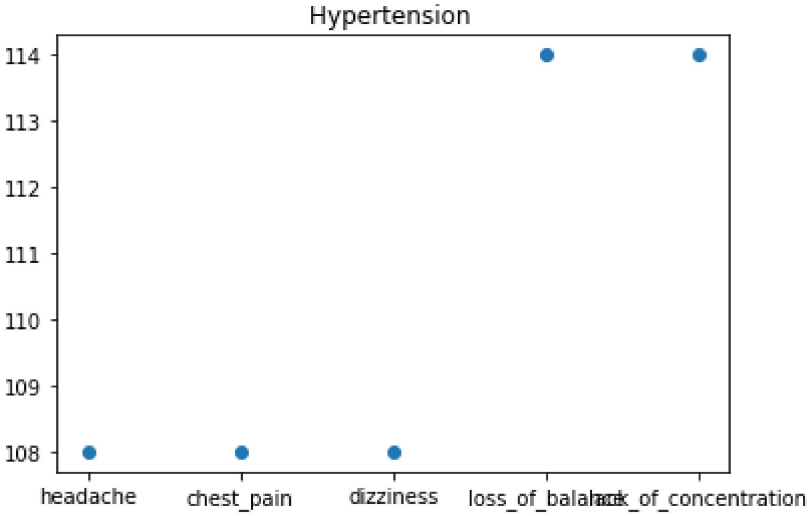

```
[114 114 114 114 114 108 114]
7
7
```



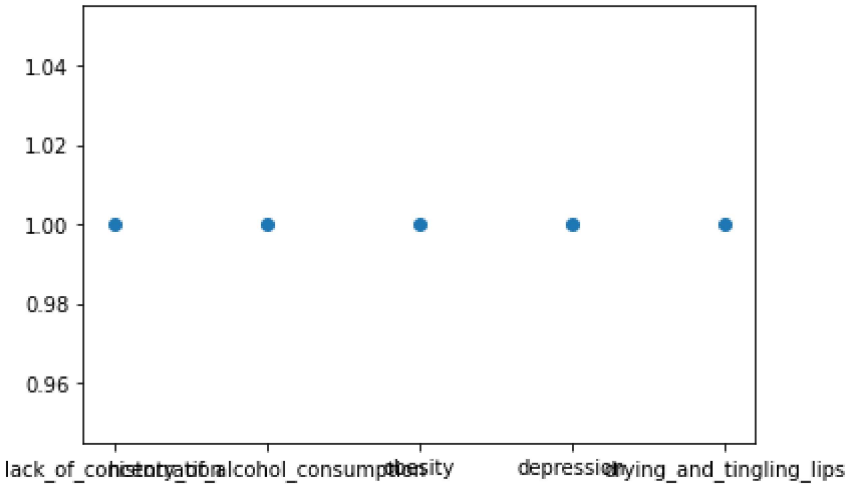

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



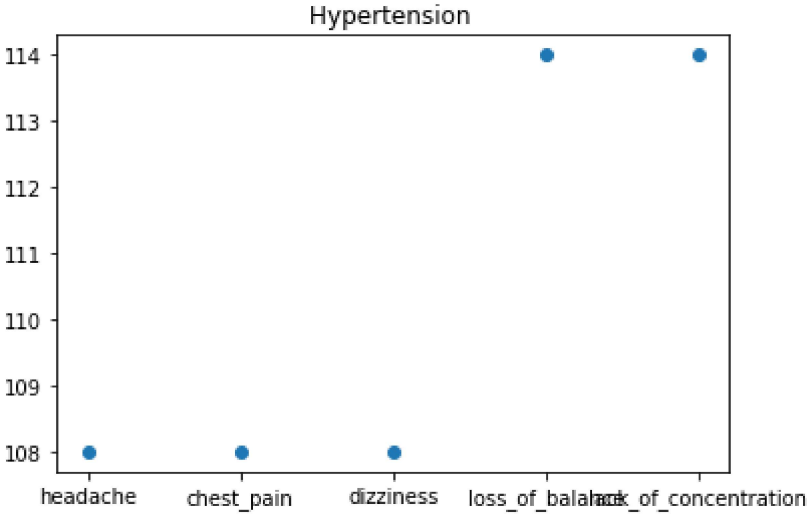

```
[108 108 108 114 114]
5
5
```



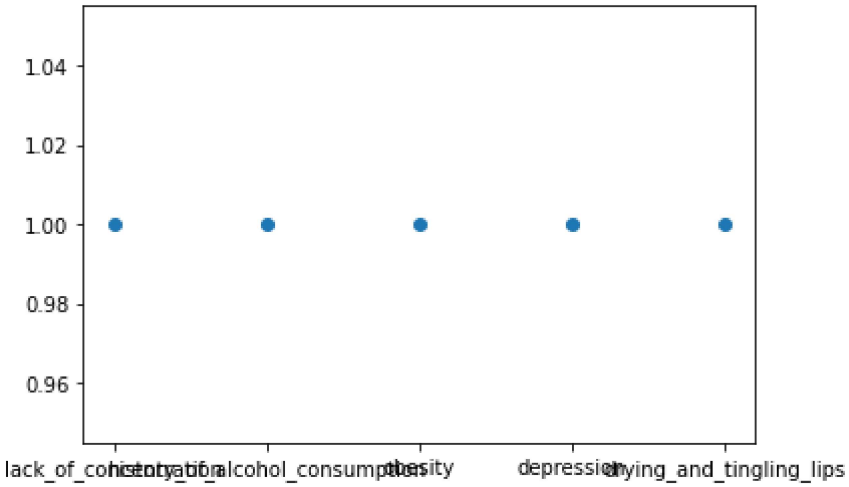

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



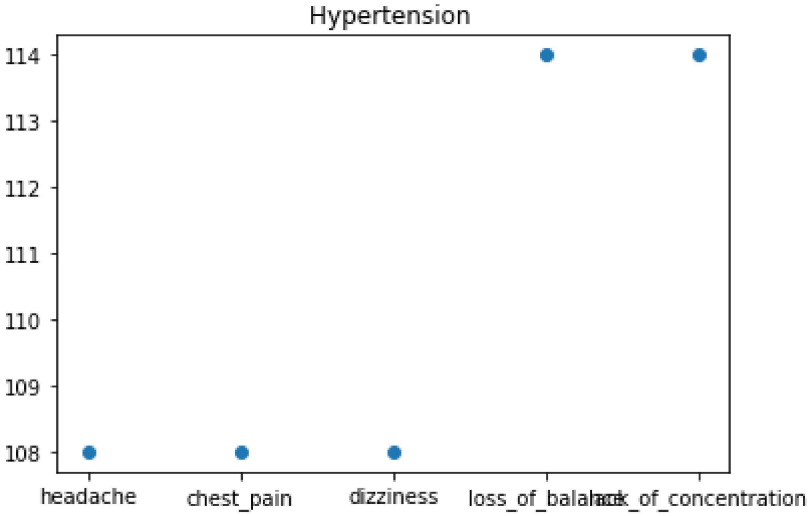

```
[108 108 108 114 114]
5
5
```



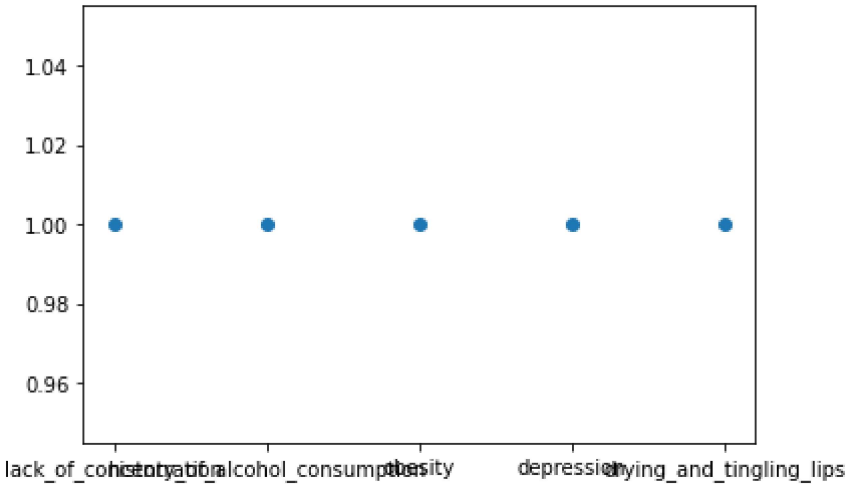

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



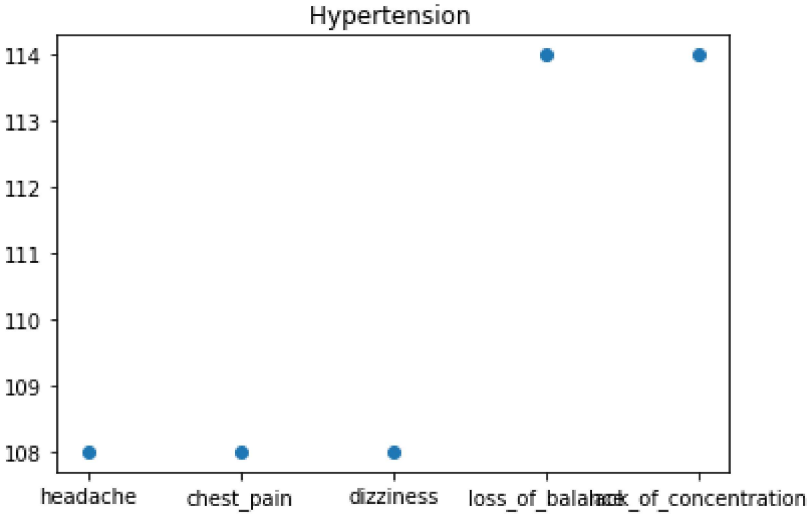

```
[108 108 108 114 114]
5
5
```



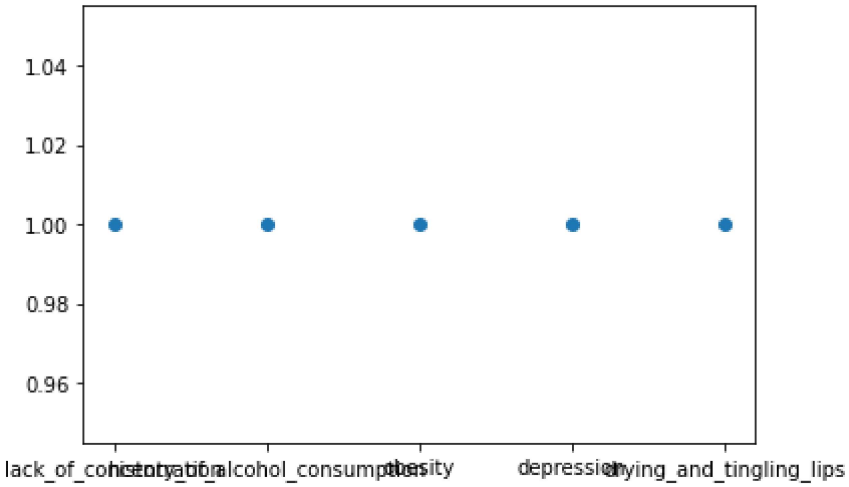

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



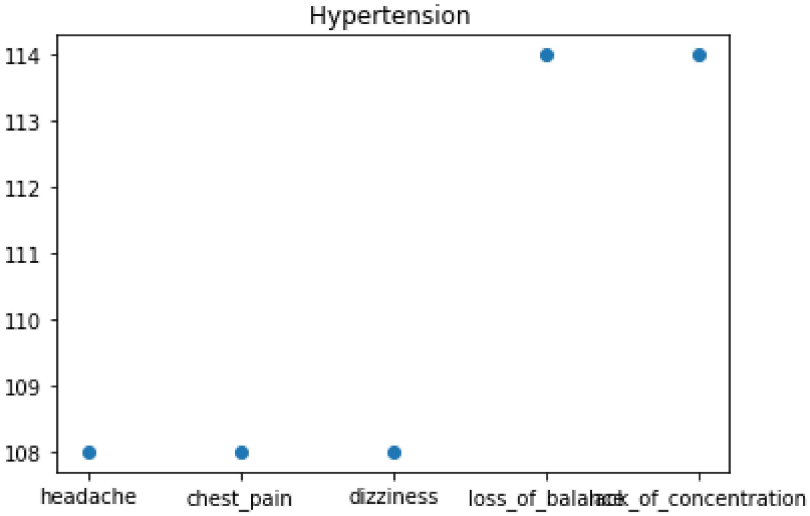

```
[108 108 108 114 114]
5
5
```



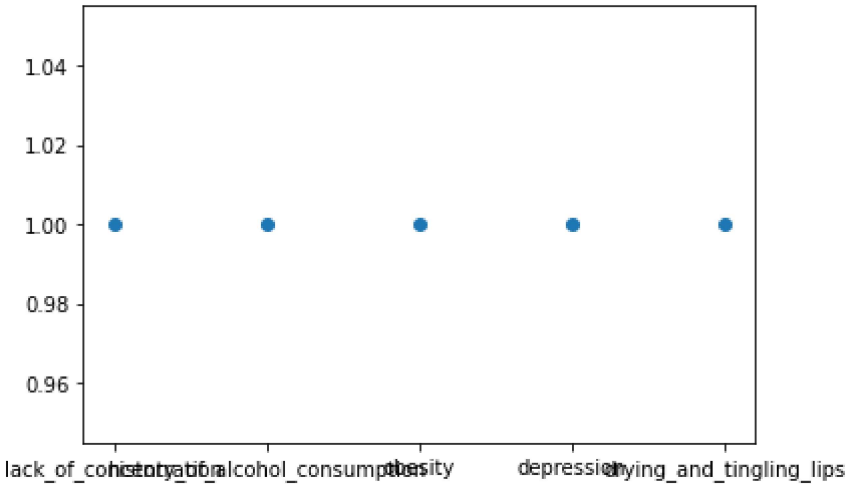

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



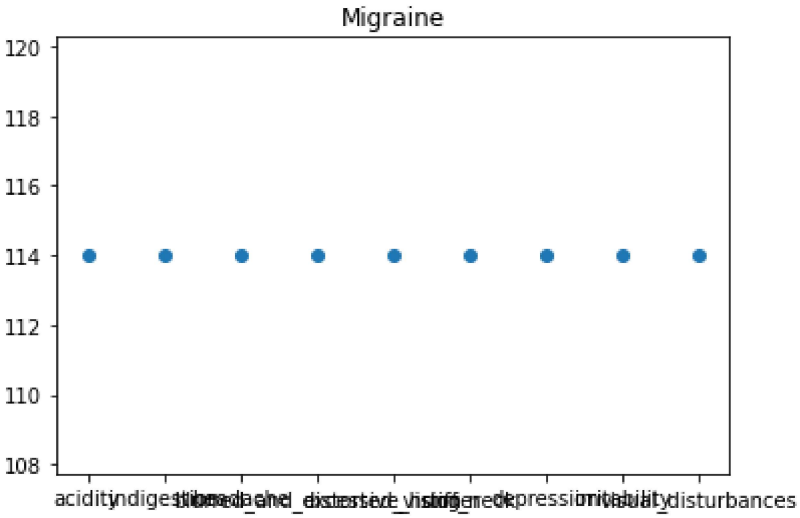

```
[108 108 108 114 114]
5
5
```




```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['lack_of_concentration', 'history_of_alcohol_consumption', 'obesity', 'depression',
'drying_and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

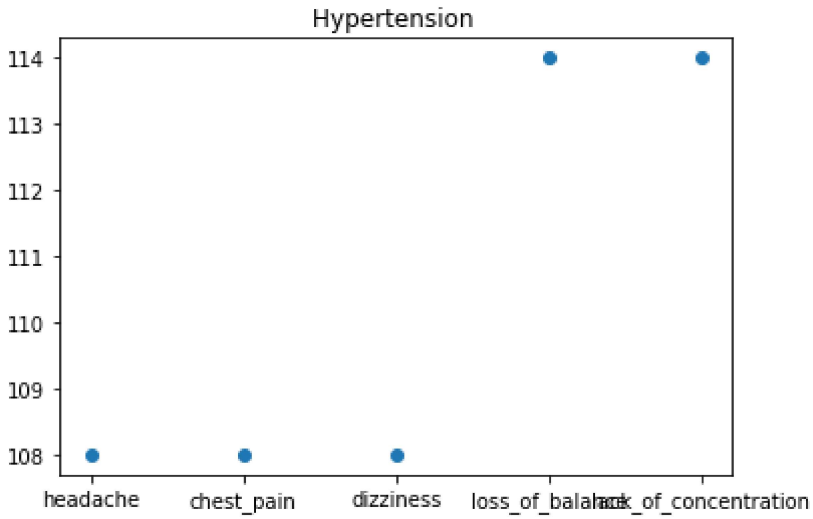



```
[114 114 114 114 114 114 114 114 114]
9
9
```



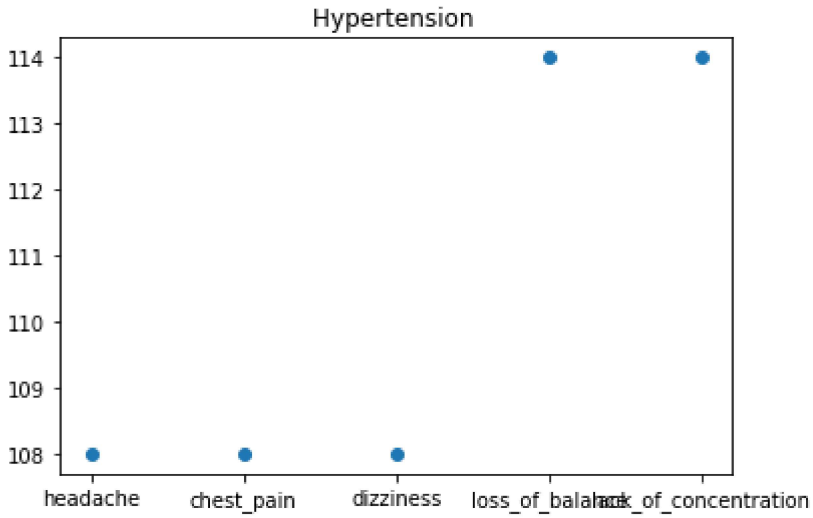

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

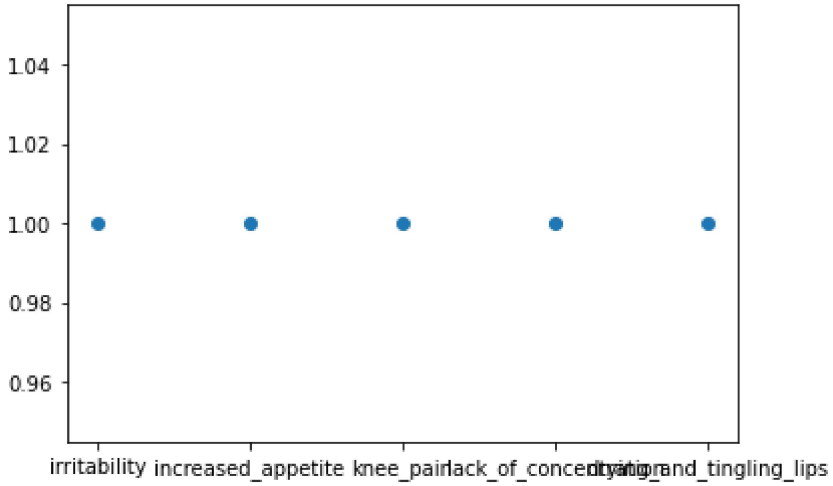



```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

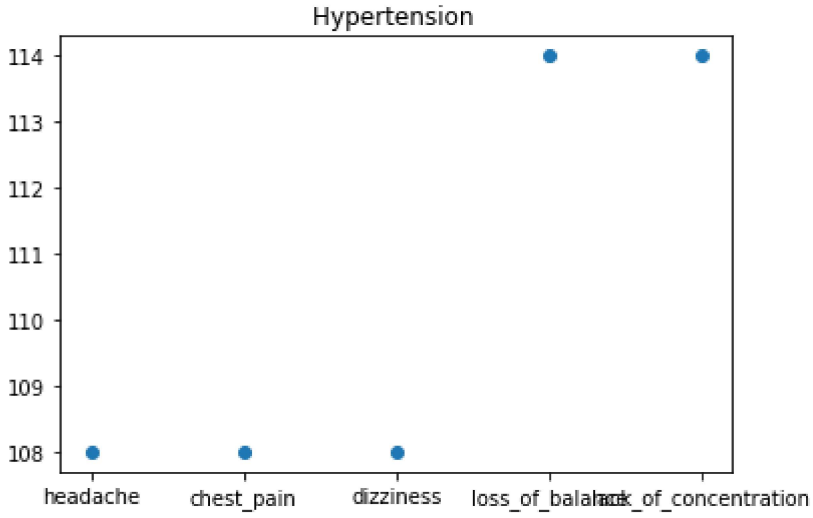
```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

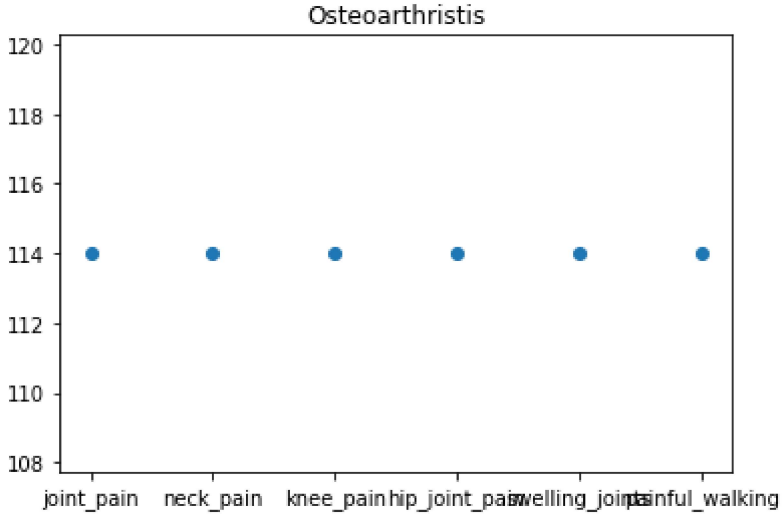



```
[108 108 108 114 114]
5
5
```



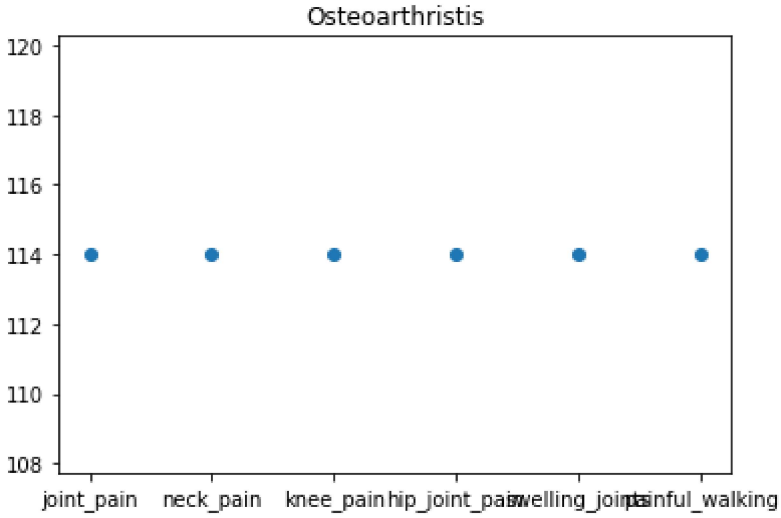
```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



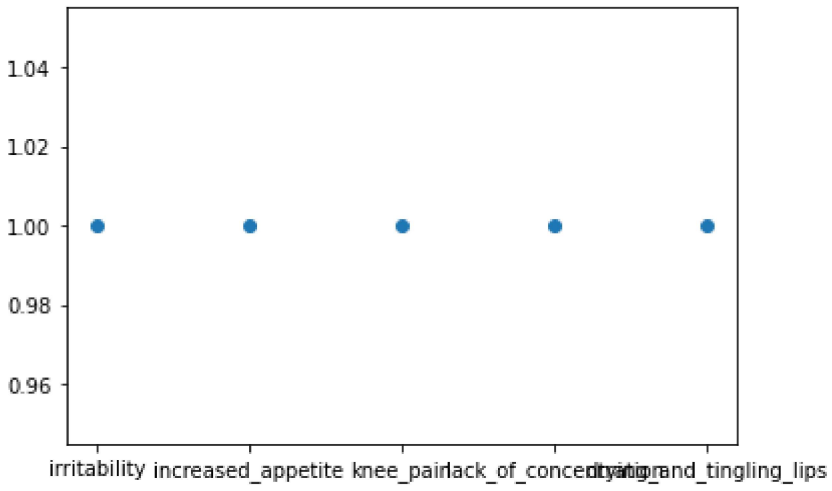

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```

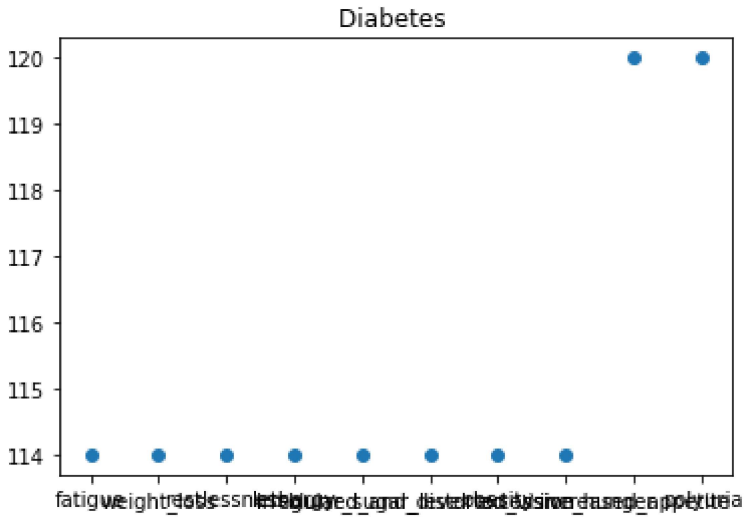



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

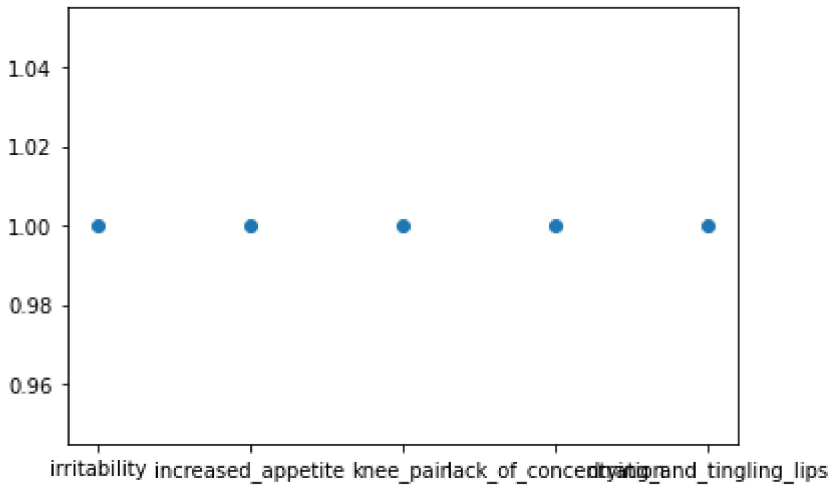



```
[114 114 114 114 114 114 114 114 120 120]
10
10
```

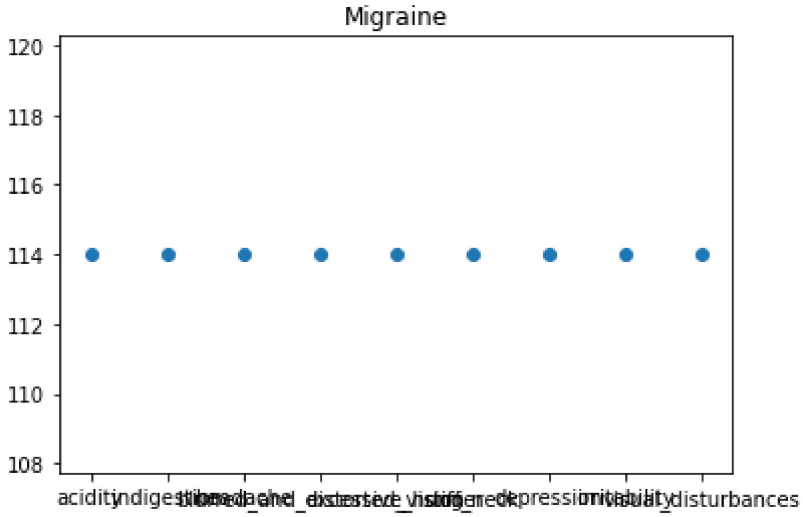



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

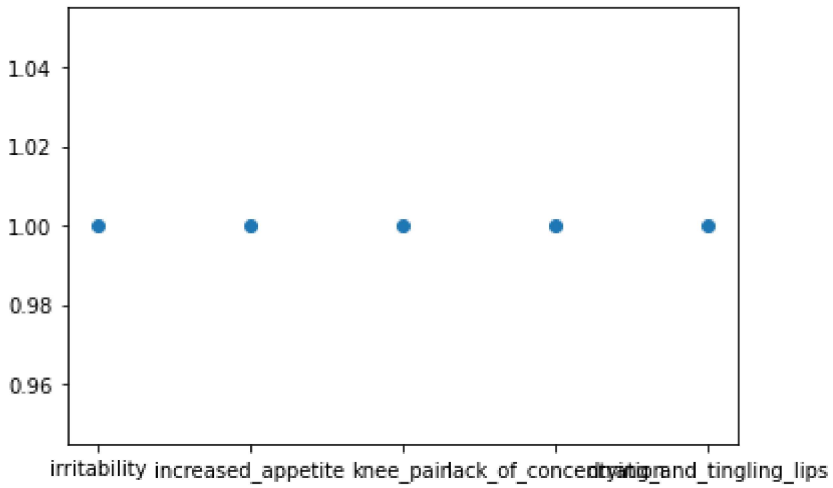



```
[114 114 114 114 114 114 114 114 114]
9
9
```

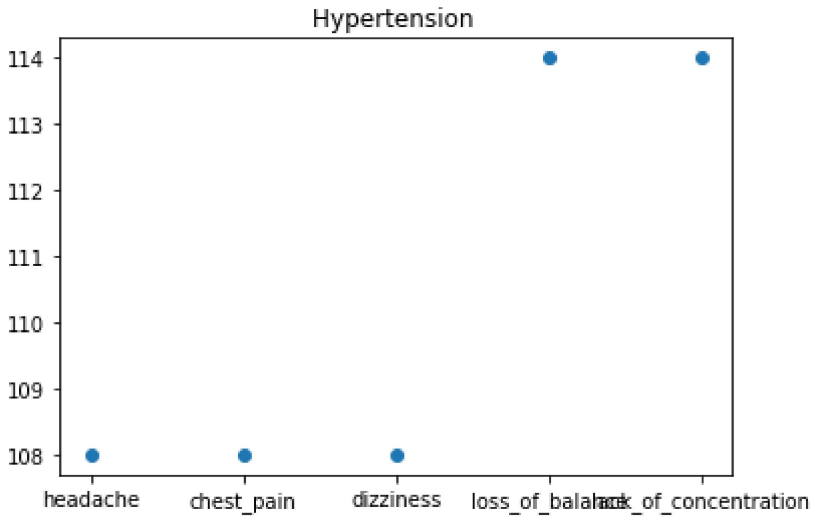



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

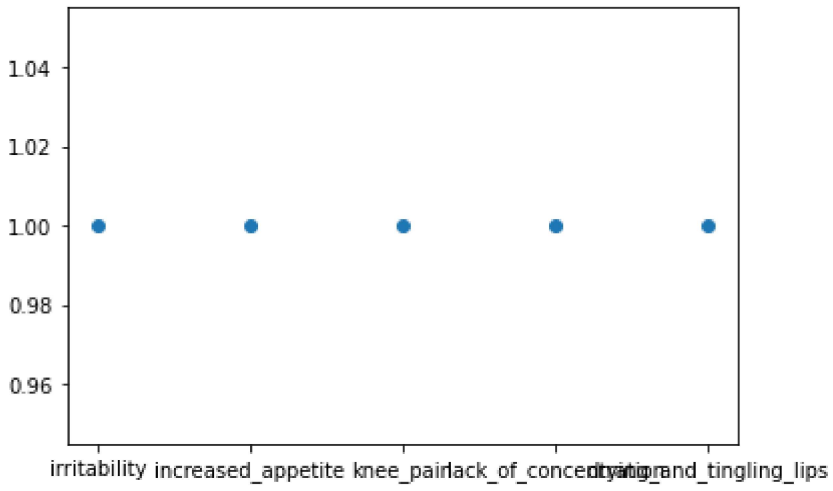



```
[108 108 108 114 114]
5
5
```

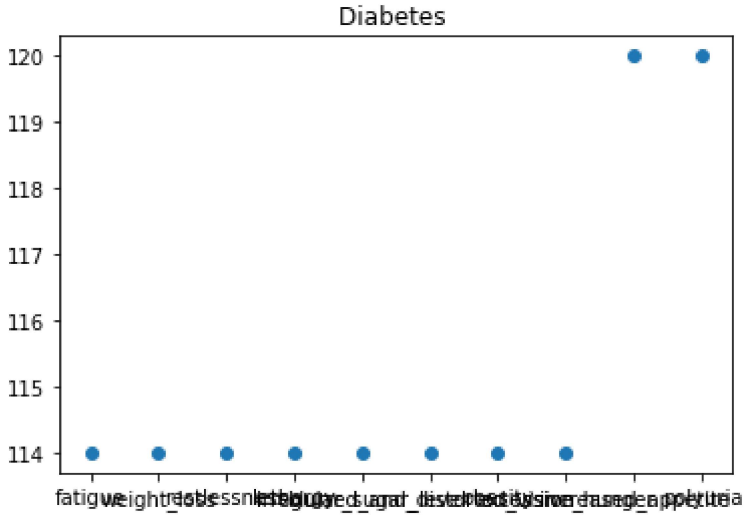



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

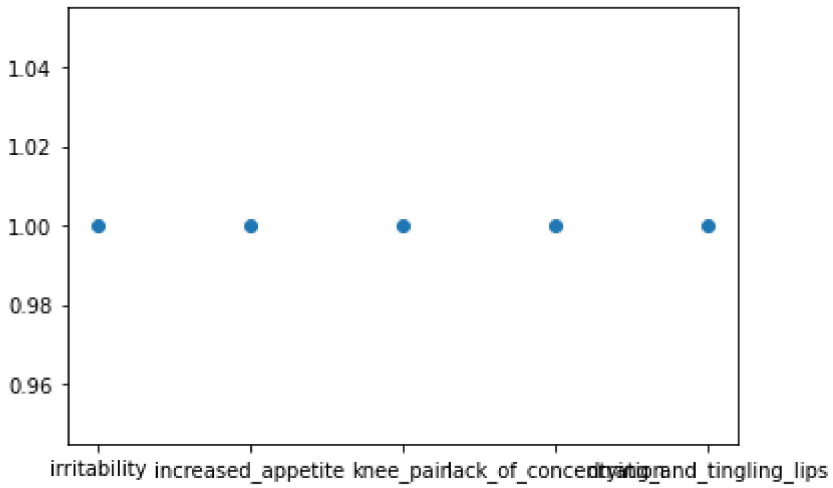



```
[114 114 114 114 114 114 114 114 120 120]
10
10
```

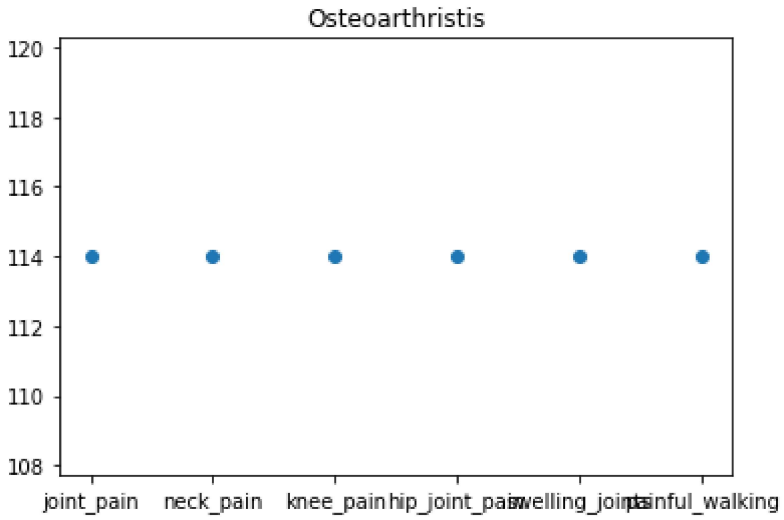



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

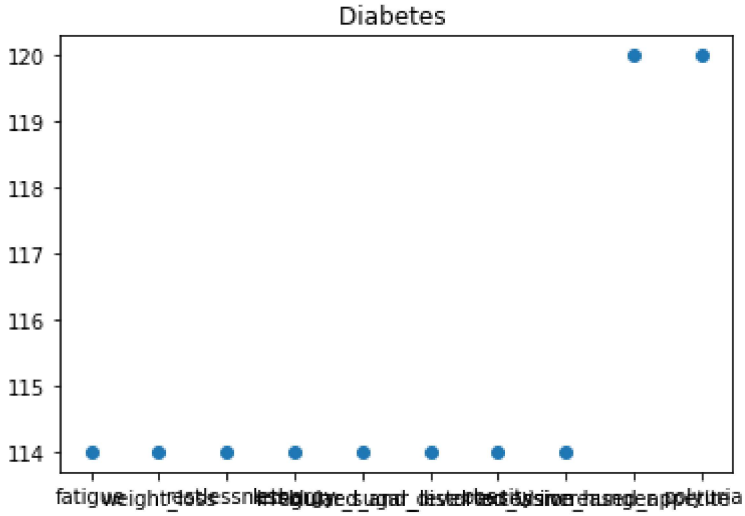



```
[114 114 114 114 114 114]
6
6
```



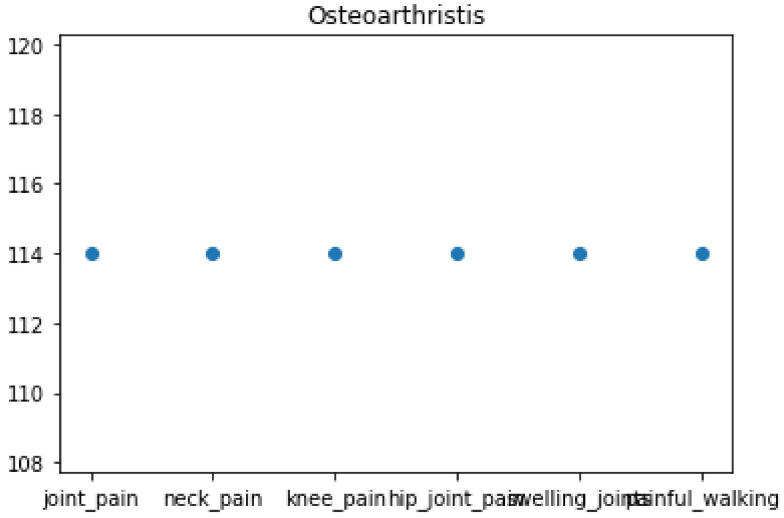

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114 114 114 120 120]
10
10
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



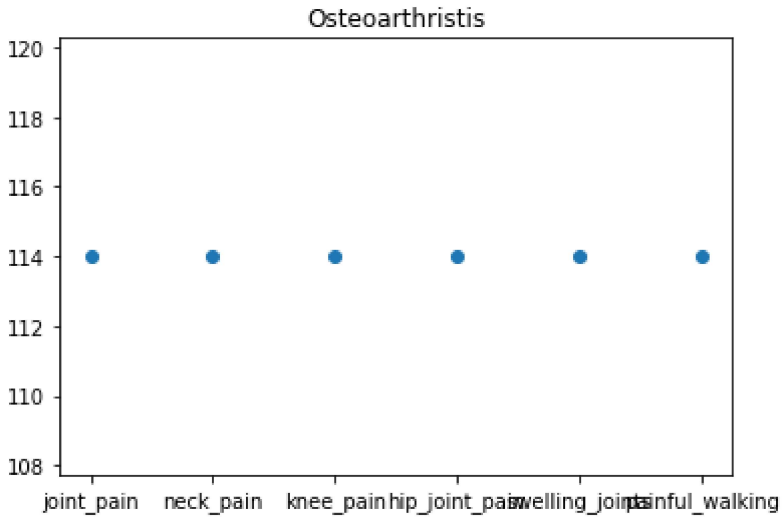
```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



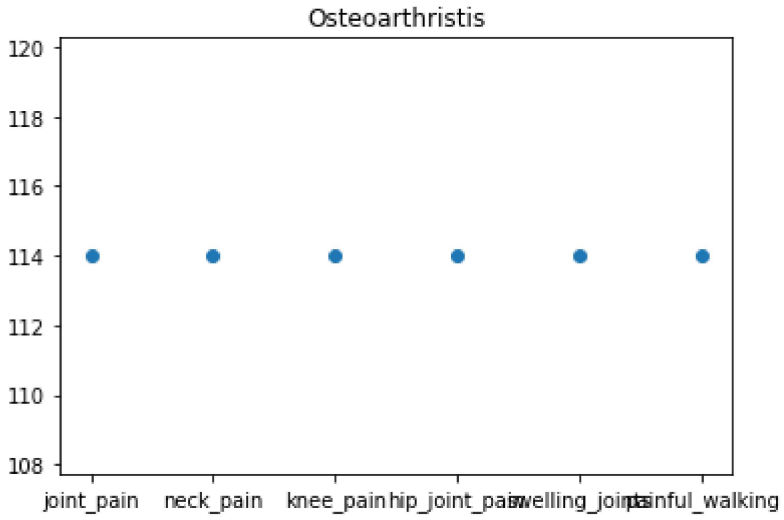

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



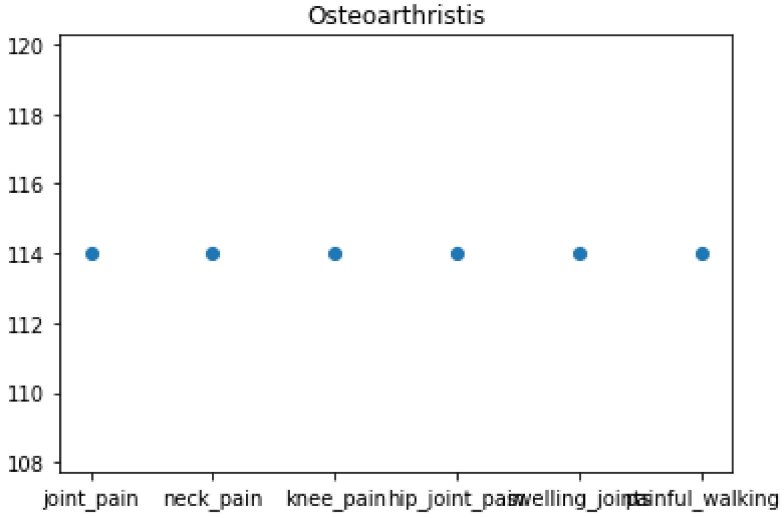

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



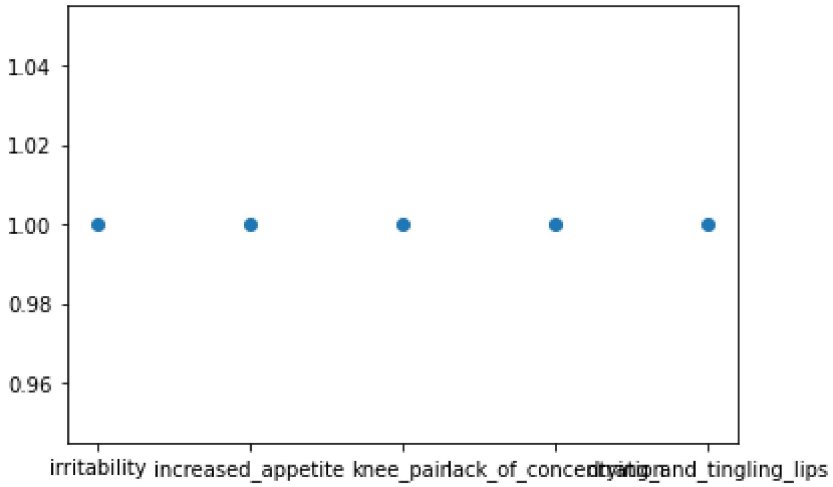
```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```

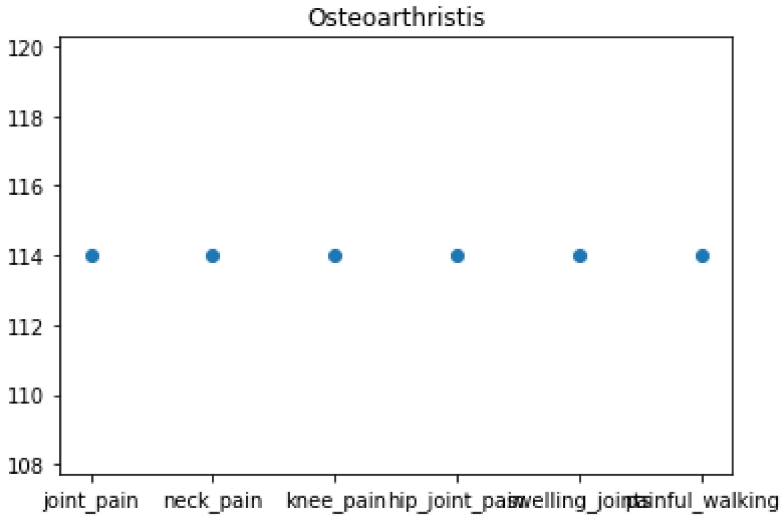



```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
```

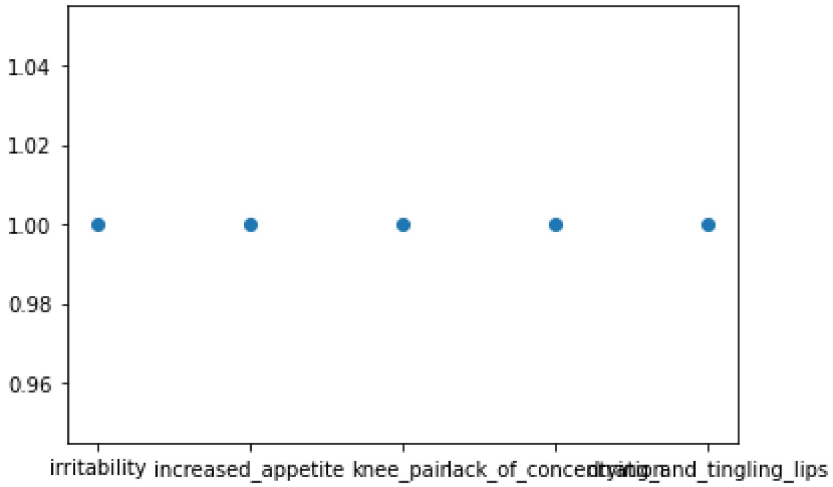
```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



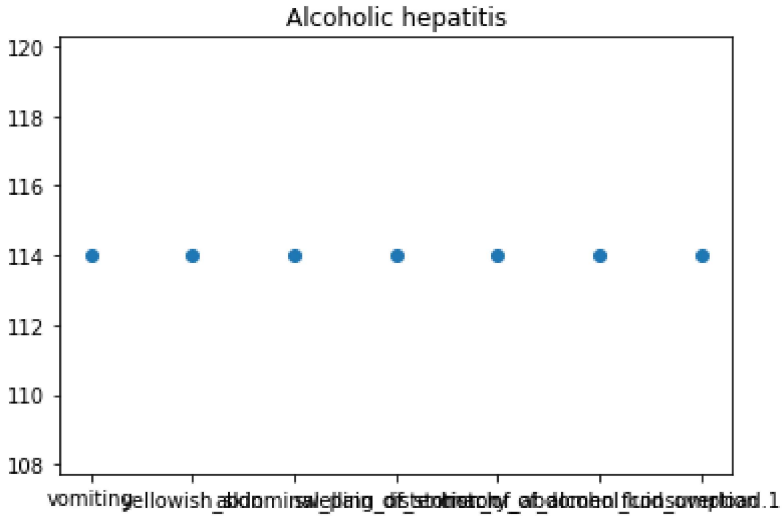

```
[114 114 114 114 114 114]
6
6
```



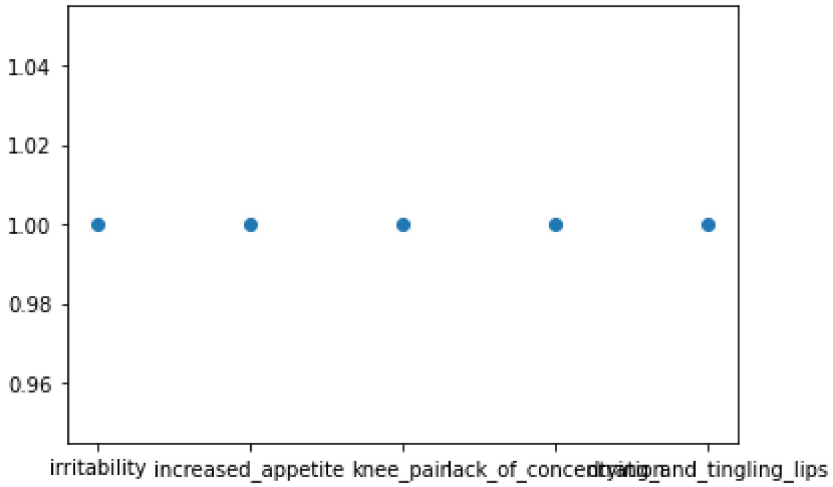

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



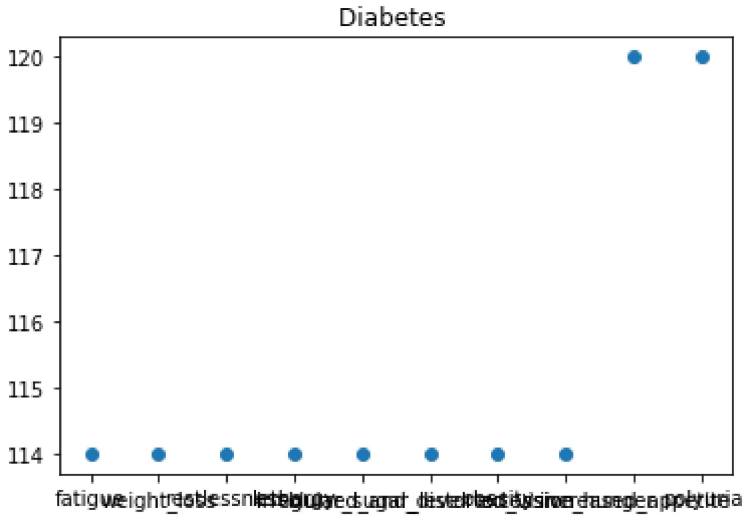

```
[114 114 114 114 114 114 114]
7
7
```



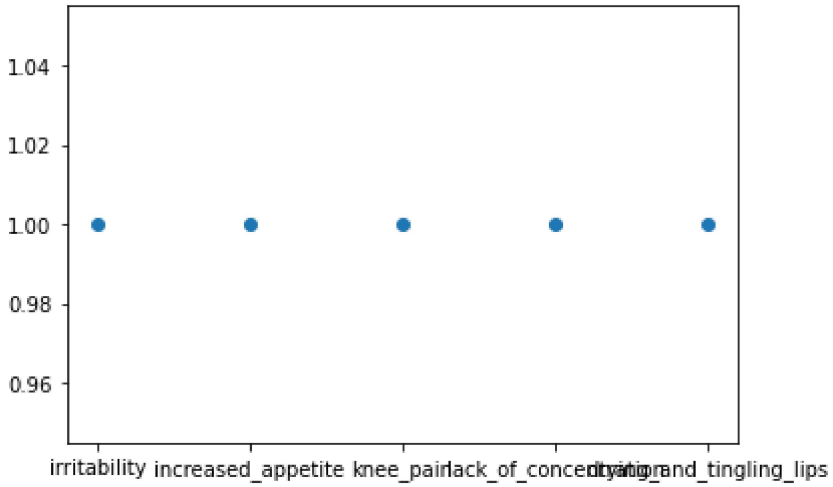

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



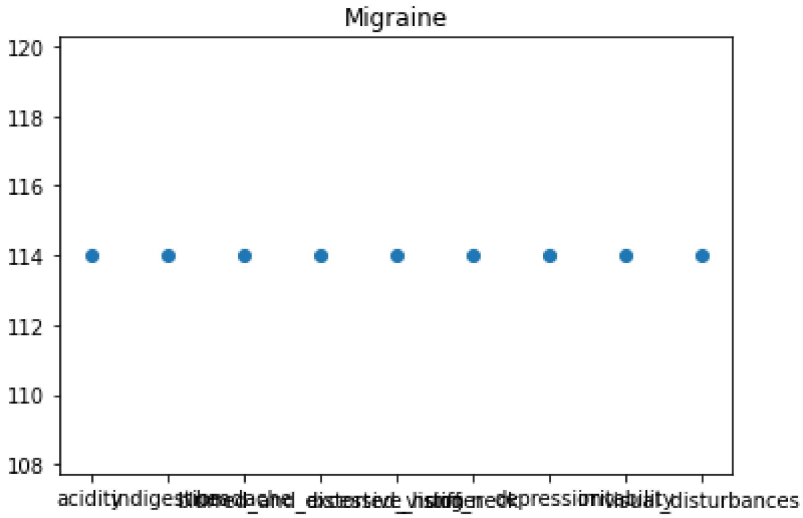

```
[114 114 114 114 114 114 114 114 120 120]
10
10
```



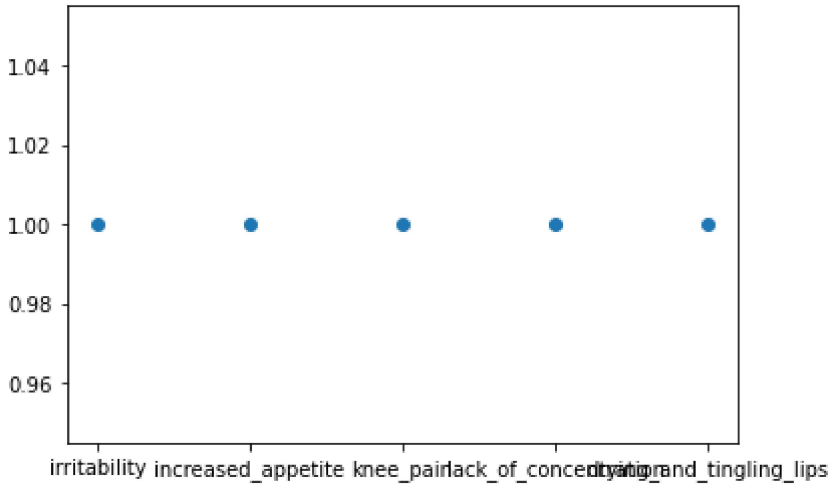

```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```



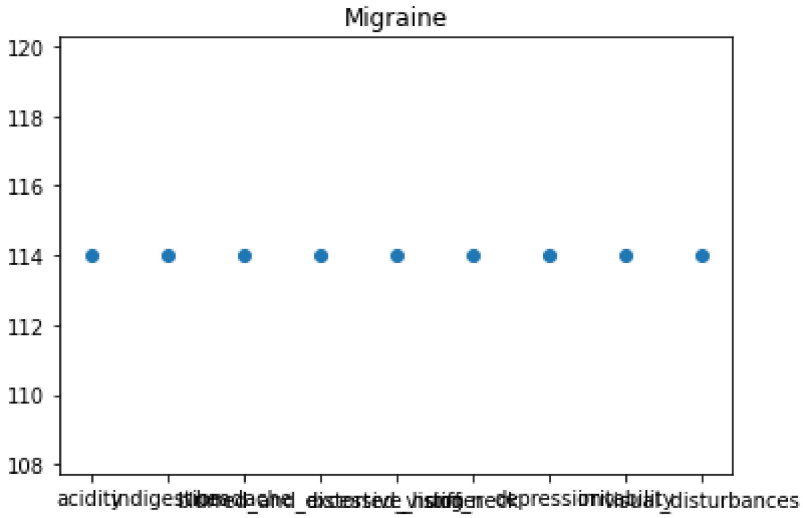

```
[114 114 114 114 114 114 114 114 114]
9
9
```




```
Decision Tree
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
['irritability', 'increased_appetite', 'knee_pain', 'lack_of_concentration', 'drying_
and_tingling_lips']
[1, 1, 1, 1, 1]
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
```

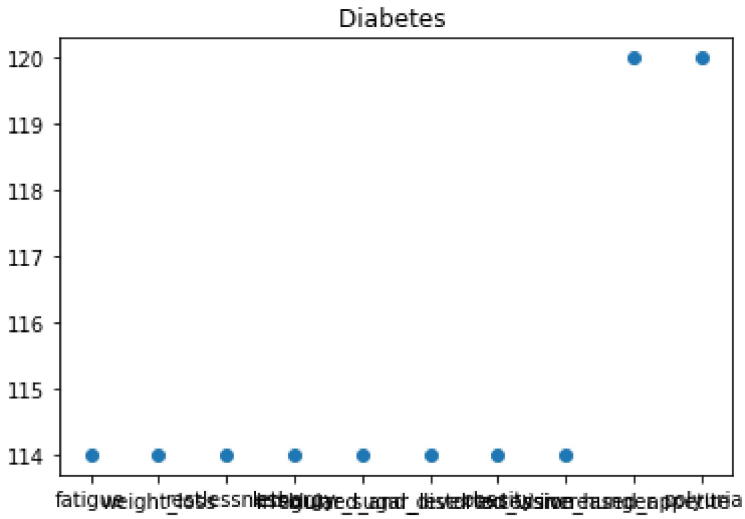



```
[114 114 114 114 114 114 114 114 114]
9
9
```



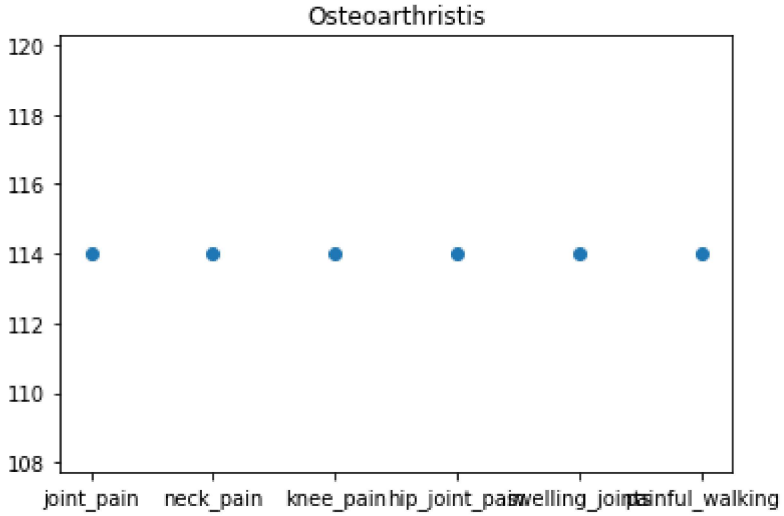

```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114 114 114 120 120]
10
10
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



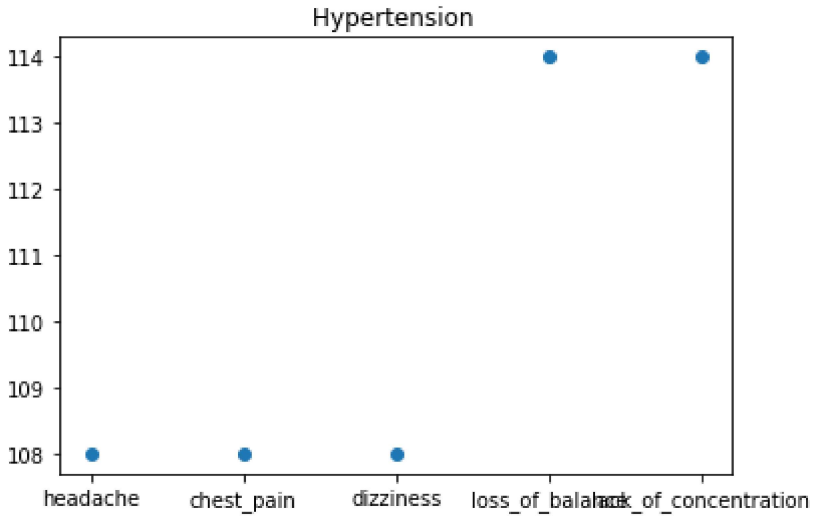

```
Naive Bayes
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[114 114 114 114 114 114]
6
6
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but GaussianNB was fitted with feature names
  warnings.warn(
```



```
Random Forest
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
[108 108 108 114 114]
5
5
```

```
C:\Users\Sazna\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does n
ot have valid feature names, but RandomForestClassifier was fitted with feature names
  warnings.warn(
```



In [ ]: