In [1]:
```
import tensorflow
print(tensorflow.__version__)
```

```
2.9.2
```

In [1]:
```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:
```
dataset=pd.read_csv("/content/aps_data.csv")
```

```
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (81) have m
ixed types.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

In [7]:
```
dataset
```

Out[7]:

	class	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	...	ee_002	ee_003	ee_004	ee_005	ee_006	ee
0	neg	6	0	0	0	0	0	0	0	0	...	26	8	26	52	0	
1	neg	90	0	0	66	0	0	0	0	0	...	1268	526	554	300	118	
2	neg	30	na	16	14	0	0	0	0	0	...	480	84	74	50	46	
3	neg	444	na	14	12	0	0	0	0	0	...	1614	1144	3598	2460	1258	
4	neg	82	na	12	10	0	0	0	0	1464	...	1010	132	310	56	92	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4995	neg	31394	na	0	na	0	0	0	0	0	...	393008	207182	367634	267778	106778	4
4996	neg	1598	na	0	na	0	0	0	0	0	...	5298	3164	9710	47042	2296	
4997	neg	42	2	2130706454	56	0	0	0	0	0	...	660	272	334	76	14	
4998	neg	32752	na	0	na	0	0	0	0	0	...	533772	37122	56252	23188	1138	
4999	neg	616	6	58	54	0	0	0	0	0	...	4630	2268	4594	2760	2108	

5000 rows × 171 columns

In [3]:
```
dataset.isnull().sum()
```

Out[3]:
```
class     0
aa_000    0
ab_000    0
ac_000    0
ad_000    0
         ..
ee_007    0
ee_008    0
ee_009    0
ef_000    0
eg_000    0
Length: 171, dtype: int64
```

In [8]:
```
np.nan
```

Out[8]:
```
nan
```

In [4]:
```
dataset=dataset.replace("na",np.nan)
```

In [5]:
```
pd.options.display.max_rows=1000
```

In [13]:
```
dataset.isnull().sum()
```

Out[13]:
```
class        0
aa_000       0
ab_000    3849
ac_000     308
ad_000    1236
ae_000     224
af_000     224
ag_000      58
ag_001      58
ag_002      58
ag_003      58
ag_004      58
ag_005      58
ag_006      58
ag_007      58
ag_008      58
ag_009      58
ah_000      60
```

ai_000 56
aj_000 56
ak_000 384
al_000 65
am_0 56
an_000 65
ao_000 53
ap_000 65
aq_000 53
ar_000 242
as_000 56
at_000 56
au_000 56
av_000 224
ax_000 224
ay_000 58
ay_001 58
ay_002 58
ay_003 58
ay_004 58
ay_005 58
ay_006 58
ay_007 58
ay_008 58
ay_009 58
az_000 58
az_001 58
az_002 58
az_003 58
az_004 58
az_005 58
az_006 58
az_007 58
az_008 58
az_009 58
ba_000 61
ba_001 61
ba_002 61
ba_003 61
ba_004 61
ba_005 61
ba_006 61
ba_007 61
ba_008 61
ba_009 61
bb_000 60
bc_000 243
bd_000 243
be_000 225
bf_000 224
bg_000 65
bh_000 65
bi_000 53
bj_000 53
bk_000 1880
bl_000 2234
bm_000 3273
bn_000 3631
bo_000 3832
bp_000 3960
bq_000 4043
br_000 4105
bs_000 66
bt_000 12
bu_000 64
bv_000 64
bx_000 288
by_000 43
bz_000 242
ca_000 383
cb_000 66
cc_000 288
cd_000 61
ce_000 224
cf_000 1236
cg_000 1236
ch_000 1236
ci_000 38
cj_000 38
ck_000 38
cl_000 778
cm_000 796
cn_000 61
cn_001 61
cn_002 61
cn_003 61
cn_004 61
cn_005 61
cn_006 61

```
cn_007      61
cn_008      61
cn_009      61
co_000    1236
cp_000     242
cq_000      64
cr_000    3849
cs_000      58
cs_001      58
cs_002      58
cs_003      58
cs_004      58
cs_005      58
cs_006      58
cs_007      58
cs_008      58
cs_009      58
ct_000    1150
cu_000    1150
cv_000    1150
cx_000    1150
cy_000    1150
cz_000    1150
da_000    1150
db_000    1150
dc_000    1150
dd_000     224
de_000     242
df_000     361
dg_000     361
dh_000     361
di_000     361
dj_000     361
dk_000     361
dl_000     361
dm_000     361
dn_000      64
do_000     243
dp_000     243
dq_000     243
dr_000     243
ds_000     243
dt_000     243
du_000     243
dv_000     243
dx_000     242
dy_000     242
dz_000     242
ea_000     242
eb_000     361
ec_00      837
ed_000     778
ee_000      58
ee_001      58
ee_002      58
ee_003      58
ee_004      58
ee_005      58
ee_006      58
ee_007      58
ee_008      58
ee_009      58
ef_000     242
eg_000     242
dtype: int64
```

In [11]:
```
dataset.isnull().sum().sum()
```

Out[11]:
```
71649
```

In [18]:
```
pd.DataFrame(dataset.isnull().sum().sort_values(ascending=False)).rename(columns={0:"Null Value Count"})
```

Out[18]:

	Null Value Count
br_000	4105
bq_000	4043
bp_000	3960
ab_000	3849
cr_000	3849
bo_000	3832
bn_000	3631
bm_000	3273
bl_000	2234

bk_000	1880
ch_000	1236
co_000	1236
cf_000	1236
cg_000	1236
ad_000	1236
db_000	1150
ct_000	1150
cu_000	1150
cv_000	1150
cx_000	1150
cy_000	1150
cz_000	1150
da_000	1150
dc_000	1150
ec_00	837
cm_000	796
cl_000	778
ed_000	778
ak_000	384
ca_000	383
di_000	361
df_000	361
dg_000	361
dh_000	361
eb_000	361
dk_000	361
dl_000	361
dm_000	361
dj_000	361
ac_000	308
cc_000	288
bx_000	288
do_000	243
bc_000	243
bd_000	243
dt_000	243
du_000	243
dv_000	243
ds_000	243
dp_000	243
dq_000	243
dr_000	243
ar_000	242
ea_000	242
ef_000	242
de_000	242
bz_000	242
eg_000	242
dz_000	242
dy_000	242
dx_000	242
cp_000	242
be_000	225
ce_000	224

ae_000	224
af_000	224
av_000	224
ax_000	224
dd_000	224
bf_000	224
bs_000	66
cb_000	66
bg_000	65
ap_000	65
al_000	65
an_000	65
bh_000	65
bv_000	64
cq_000	64
dn_000	64
bu_000	64
cn_005	61
ba_002	61
cn_009	61
cn_008	61
cn_007	61
cn_006	61
cn_004	61
cn_003	61
cn_002	61
cn_001	61
cn_000	61
ba_000	61
ba_001	61
ba_003	61
ba_005	61
ba_009	61
ba_007	61
ba_006	61
cd_000	61
ba_008	61
ba_004	61
bb_000	60
ah_000	60
ag_007	58
ay_003	58
ay_005	58
ay_002	58
ay_004	58
ay_006	58
ay_007	58
ay_001	58
ay_000	58
ag_009	58
ag_008	58
cs_008	58
ag_006	58
ag_005	58
ag_004	58

ag_003	58
ag_002	58
ag_001	58
ay_009	58
ag_000	58
ee_000	58
ee_001	58
ee_002	58
ee_003	58
ay_008	58
ee_004	58
az_000	58
az_006	58
cs_001	58
cs_002	58
cs_003	58
cs_004	58
cs_005	58
az_001	58
az_007	58
az_008	58
az_009	58
ee_006	58
ee_007	58
ee_008	58
ee_009	58
cs_006	58
cs_007	58
ee_005	58
cs_009	58
az_005	58
az_004	58
az_003	58
az_002	58
cs_000	58
ai_000	56
aj_000	56
am_0	56
as_000	56
at_000	56
au_000	56
ao_000	53
aq_000	53
bi_000	53
bj_000	53
by_000	43
ck_000	38
cj_000	38
ci_000	38
bt_000	12
aa_000	0
class	0

In [19]:
```
[feature for feature in dataset.columns if feature not in ['class']]
```

Out[19]:
```
['aa_000',
 'ab_000',
 'ac_000',
 'ad_000',
 'ae_000',
 'af_000',
 'ag_000',
 'ag_001',
 'ag_002',
 'ag_003',
 'ag_004',
 'ag_005',
 'ag_006',
 'ag_007',
 'ag_008',
 'ag_009',
 'ah_000',
 'ai_000',
 'aj_000',
 'ak_000',
 'al_000',
 'am_0',
 'an_000',
 'ao_000',
 'ap_000',
 'aq_000',
 'ar_000',
 'as_000',
 'at_000',
 'au_000',
 'av_000',
 'ax_000',
 'ay_000',
 'ay_001',
 'ay_002',
 'ay_003',
 'ay_004',
 'ay_005',
 'ay_006',
 'ay_007',
 'ay_008',
 'ay_009',
 'az_000',
 'az_001',
 'az_002',
 'az_003',
 'az_004',
 'az_005',
 'az_006',
 'az_007',
 'az_008',
 'az_009',
 'ba_000',
 'ba_001',
 'ba_002',
 'ba_003',
 'ba_004',
 'ba_005',
 'ba_006',
 'ba_007',
 'ba_008',
 'ba_009',
 'bb_000',
 'bc_000',
 'bd_000',
 'be_000',
 'bf_000',
 'bg_000',
 'bh_000',
 'bi_000',
 'bj_000',
 'bk_000',
 'bl_000',
 'bm_000',
 'bn_000',
 'bo_000',
 'bp_000',
 'bq_000',
 'br_000',
 'bs_000',
 'bt_000',
 'bu_000',
 'bv_000',
 'bx_000',
 'by_000',
 'bz_000',
 'ca_000',
 'cb_000',
 'cc_000',
```

```
'cd_000',
'ce_000',
'cf_000',
'cg_000',
'ch_000',
'ci_000',
'cj_000',
'ck_000',
'cl_000',
'cm_000',
'cn_000',
'cn_001',
'cn_002',
'cn_003',
'cn_004',
'cn_005',
'cn_006',
'cn_007',
'cn_008',
'cn_009',
'co_000',
'cp_000',
'cq_000',
'cr_000',
'cs_000',
'cs_001',
'cs_002',
'cs_003',
'cs_004',
'cs_005',
'cs_006',
'cs_007',
'cs_008',
'cs_009',
'ct_000',
'cu_000',
'cv_000',
'cx_000',
'cy_000',
'cz_000',
'da_000',
'db_000',
'dc_000',
'dd_000',
'de_000',
'df_000',
'dg_000',
'dh_000',
'di_000',
'dj_000',
'dk_000',
'dl_000',
'dm_000',
'dn_000',
'do_000',
'dp_000',
'dq_000',
'dr_000',
'ds_000',
'dt_000',
'du_000',
'dv_000',
'dx_000',
'dy_000',
'dz_000',
'ea_000',
'eb_000',
'ec_00',
'ed_000',
'ee_000',
'ee_001',
'ee_002',
'ee_003',
'ee_004',
'ee_005',
'ee_006',
'ee_007',
'ee_008',
'ee_009',
'ef_000',
'eg_000']
```

In [6]:
```
### Filling null values in feature with median of that feature
for feature in [feature for feature in dataset.columns if feature not in ['class']]:
    dataset[feature]=dataset[feature].fillna(dataset[feature].median())
```

In [21]:
```
dataset.isnull().sum()
```

Out[21]:

class	0
aa_000	0
ab_000	0
ac_000	0
ad_000	0
ae_000	0
af_000	0
ag_000	0
ag_001	0
ag_002	0
ag_003	0
ag_004	0
ag_005	0
ag_006	0
ag_007	0
ag_008	0
ag_009	0
ah_000	0
ai_000	0
aj_000	0
ak_000	0
al_000	0
am_0	0
an_000	0
ao_000	0
ap_000	0
aq_000	0
ar_000	0
as_000	0
at_000	0
au_000	0
av_000	0
ax_000	0
ay_000	0
ay_001	0
ay_002	0
ay_003	0
ay_004	0
ay_005	0
ay_006	0
ay_007	0
ay_008	0
ay_009	0
az_000	0
az_001	0
az_002	0
az_003	0
az_004	0
az_005	0
az_006	0
az_007	0
az_008	0
az_009	0
ba_000	0
ba_001	0
ba_002	0
ba_003	0
ba_004	0
ba_005	0
ba_006	0
ba_007	0
ba_008	0
ba_009	0
bb_000	0
bc_000	0
bd_000	0
be_000	0
bf_000	0
bg_000	0
bh_000	0
bi_000	0
bj_000	0
bk_000	0
bl_000	0
bm_000	0
bn_000	0
bo_000	0
bp_000	0
bq_000	0
br_000	0
bs_000	0
bt_000	0
bu_000	0
bv_000	0
bx_000	0
by_000	0
bz_000	0
ca_000	0
cb_000	0

```
cc_000    0
cd_000    0
ce_000    0
cf_000    0
cg_000    0
ch_000    0
ci_000    0
cj_000    0
ck_000    0
cl_000    0
cm_000    0
cn_000    0
cn_001    0
cn_002    0
cn_003    0
cn_004    0
cn_005    0
cn_006    0
cn_007    0
cn_008    0
cn_009    0
co_000    0
cp_000    0
cq_000    0
cr_000    0
cs_000    0
cs_001    0
cs_002    0
cs_003    0
cs_004    0
cs_005    0
cs_006    0
cs_007    0
cs_008    0
cs_009    0
ct_000    0
cu_000    0
cv_000    0
cx_000    0
cy_000    0
cz_000    0
da_000    0
db_000    0
dc_000    0
dd_000    0
de_000    0
df_000    0
dg_000    0
dh_000    0
di_000    0
dj_000    0
dk_000    0
dl_000    0
dm_000    0
dn_000    0
do_000    0
dp_000    0
dq_000    0
dr_000    0
ds_000    0
dt_000    0
du_000    0
dv_000    0
dx_000    0
dy_000    0
dz_000    0
ea_000    0
eb_000    0
ec_00     0
ed_000    0
ee_000    0
ee_001    0
ee_002    0
ee_003    0
ee_004    0
ee_005    0
ee_006    0
ee_007    0
ee_008    0
ee_009    0
ef_000    0
eg_000    0
dtype: int64
```

In [22]:
```
dataset.isnull().sum().sum()
```

Out[22]:
```
0
```

In [7]:
```
dataset.isnull().sum()
```

In [7]:
```
dataset.isnull().sum()
```

Out[7]:

```
class     0
aa_000    0
ab_000    0
ac_000    0
ad_000    0
ae_000    0
af_000    0
ag_000    0
ag_001    0
ag_002    0
ag_003    0
ag_004    0
ag_005    0
ag_006    0
ag_007    0
ag_008    0
ag_009    0
ah_000    0
ai_000    0
aj_000    0
ak_000    0
al_000    0
am_0      0
an_000    0
ao_000    0
ap_000    0
aq_000    0
ar_000    0
as_000    0
at_000    0
au_000    0
av_000    0
ax_000    0
ay_000    0
ay_001    0
ay_002    0
ay_003    0
ay_004    0
ay_005    0
ay_006    0
ay_007    0
ay_008    0
ay_009    0
az_000    0
az_001    0
az_002    0
az_003    0
az_004    0
az_005    0
az_006    0
az_007    0
az_008    0
az_009    0
ba_000    0
ba_001    0
ba_002    0
ba_003    0
ba_004    0
ba_005    0
ba_006    0
ba_007    0
ba_008    0
ba_009    0
bb_000    0
bc_000    0
bd_000    0
be_000    0
bf_000    0
bg_000    0
bh_000    0
bi_000    0
bj_000    0
bk_000    0
bl_000    0
bm_000    0
bn_000    0
bo_000    0
bp_000    0
bq_000    0
br_000    0
bs_000    0
bt_000    0
bu_000    0
bv_000    0
bx_000    0
by_000    0
bz_000    0
```

```
ca_000    0
cb_000    0
cc_000    0
cd_000    0
ce_000    0
cf_000    0
cg_000    0
ch_000    0
ci_000    0
cj_000    0
ck_000    0
cl_000    0
cm_000    0
cn_000    0
cn_001    0
cn_002    0
cn_003    0
cn_004    0
cn_005    0
cn_006    0
cn_007    0
cn_008    0
cn_009    0
co_000    0
cp_000    0
cq_000    0
cr_000    0
cs_000    0
cs_001    0
cs_002    0
cs_003    0
cs_004    0
cs_005    0
cs_006    0
cs_007    0
cs_008    0
cs_009    0
ct_000    0
cu_000    0
cv_000    0
cx_000    0
cy_000    0
cz_000    0
da_000    0
db_000    0
dc_000    0
dd_000    0
de_000    0
df_000    0
dg_000    0
dh_000    0
di_000    0
dj_000    0
dk_000    0
dl_000    0
dm_000    0
dn_000    0
do_000    0
dp_000    0
dq_000    0
dr_000    0
ds_000    0
dt_000    0
du_000    0
dv_000    0
dx_000    0
dy_000    0
dz_000    0
ea_000    0
eb_000    0
ec_00     0
ed_000    0
ee_000    0
ee_001    0
ee_002    0
ee_003    0
ee_004    0
ee_005    0
ee_006    0
ee_007    0
ee_008    0
ee_009    0
ef_000    0
eg_000    0
dtype: int64
```

In [24]:
```
pd.DataFrame(dataset.isnull().sum().sort_values(ascending=False)).rename(columns={0:"Null Value Count"})
```

	Null Value Count

Out[24]:

class	0
cs_003	0
cn_009	0
co_000	0
cp_000	0
cq_000	0
cr_000	0
cs_000	0
cs_001	0
cs_002	0
cs_004	0
cy_000	0
cs_005	0
cs_006	0
cs_007	0
cs_008	0
cs_009	0
ct_000	0
cu_000	0
cv_000	0
cn_008	0
cn_007	0
cn_006	0
cn_005	0
cb_000	0
cc_000	0
cd_000	0
ce_000	0
cf_000	0
cg_000	0
ch_000	0
ci_000	0
cj_000	0
ck_000	0
cl_000	0
cm_000	0
cn_000	0
cn_001	0
cn_002	0
cn_003	0
cn_004	0
cx_000	0
cz_000	0
bz_000	0
ee_001	0
dx_000	0
dy_000	0
dz_000	0
ea_000	0
eb_000	0
ec_00	0
ed_000	0
ee_000	0
ee_002	0
da_000	0

ee_003	0
ee_004	0
ee_005	0
ee_006	0
ee_007	0
ee_008	0
ee_009	0
ef_000	0
dv_000	0
du_000	0
dt_000	0
ds_000	0
db_000	0
dc_000	0
dd_000	0
de_000	0
df_000	0
dg_000	0
dh_000	0
di_000	0
dj_000	0
dk_000	0
dl_000	0
dm_000	0
dn_000	0
do_000	0
dp_000	0
dq_000	0
dr_000	0
ca_000	0
by_000	0
aa_000	0
av_000	0
an_000	0
ao_000	0
ap_000	0
aq_000	0
ar_000	0
as_000	0
at_000	0
au_000	0
ax_000	0
ay_009	0
ay_000	0
ay_001	0
ay_002	0
ay_003	0
ay_004	0
ay_005	0
ay_006	0
ay_007	0
am_0	0
al_000	0
ak_000	0

aj_000	0
ab_000	0
ac_000	0
ad_000	0
ae_000	0
af_000	0
ag_000	0
ag_001	0
ag_002	0
ag_003	0
ag_004	0
ag_005	0
ag_006	0
ag_007	0
ag_008	0
ag_009	0
ah_000	0
ai_000	0
ay_008	0
az_000	0
bx_000	0
bm_000	0
be_000	0
bf_000	0
bg_000	0
bh_000	0
bi_000	0
bj_000	0
bk_000	0
bl_000	0
bn_000	0
az_001	0
bo_000	0
bp_000	0
bq_000	0
br_000	0
bs_000	0
bt_000	0
bu_000	0
bv_000	0
bd_000	0
bc_000	0
bb_000	0
ba_009	0
az_002	0
az_003	0
az_004	0
az_005	0
az_006	0
az_007	0
az_008	0
az_009	0
ba_000	0
ba_001	0
ba_002	0

ba_003	0
ba_004	0
ba_005	0
ba_006	0
ba_007	0
ba_008	0
eg_000	0

In [25]:
```
dataset.head()
```

Out[25]:

	class	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	...	ee_002	ee_003	ee_004	ee_005	ee_006	ee_007	e
0	neg	6	0	0	0	0	0	0	0	0	...	26	8	26	52	0	0	
1	neg	90	0	0	66	0	0	0	0	0	...	1268	526	554	300	118	260	
2	neg	30	0.0	16	14	0	0	0	0	0	...	480	84	74	50	46	0	
3	neg	444	0.0	14	12	0	0	0	0	0	...	1614	1144	3598	2460	1258	8524	
4	neg	82	0.0	12	10	0	0	0	0	1464	...	1010	132	310	56	92	1292	

5 rows × 171 columns

In [8]:
```
X=dataset.drop("class",axis=1)
```

In [9]:
```
y=dataset["class"]
```

In [29]:
```
X.info(verbose=True,show_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 170 columns):
 #    Column  Non-Null Count  Dtype
---   ------  --------------  -----
 0    aa_000  5000 non-null   int64
 1    ab_000  5000 non-null   object
 2    ac_000  5000 non-null   object
 3    ad_000  5000 non-null   object
 4    ae_000  5000 non-null   object
 5    af_000  5000 non-null   object
 6    ag_000  5000 non-null   object
 7    ag_001  5000 non-null   object
 8    ag_002  5000 non-null   object
 9    ag_003  5000 non-null   object
 10   ag_004  5000 non-null   object
 11   ag_005  5000 non-null   object
 12   ag_006  5000 non-null   object
 13   ag_007  5000 non-null   object
 14   ag_008  5000 non-null   object
 15   ag_009  5000 non-null   object
 16   ah_000  5000 non-null   object
 17   ai_000  5000 non-null   object
 18   aj_000  5000 non-null   object
 19   ak_000  5000 non-null   object
 20   al_000  5000 non-null   object
 21   am_0    5000 non-null   object
 22   an_000  5000 non-null   object
 23   ao_000  5000 non-null   object
 24   ap_000  5000 non-null   object
 25   aq_000  5000 non-null   object
 26   ar_000  5000 non-null   object
 27   as_000  5000 non-null   object
 28   at_000  5000 non-null   object
 29   au_000  5000 non-null   object
 30   av_000  5000 non-null   object
 31   ax_000  5000 non-null   object
 32   ay_000  5000 non-null   object
 33   ay_001  5000 non-null   object
 34   ay_002  5000 non-null   object
 35   ay_003  5000 non-null   object
 36   ay_004  5000 non-null   object
 37   ay_005  5000 non-null   object
 38   ay_006  5000 non-null   object
 39   ay_007  5000 non-null   object
 40   ay_008  5000 non-null   object
 41   ay_009  5000 non-null   object
 42   az_000  5000 non-null   object
 43   az_001  5000 non-null   object
 44   az_002  5000 non-null   object
 45   az_003  5000 non-null   object
```

```
46   az_004  5000 non-null   object
47   az_005  5000 non-null   object
48   az_006  5000 non-null   object
49   az_007  5000 non-null   object
50   az_008  5000 non-null   object
51   az_009  5000 non-null   object
52   ba_000  5000 non-null   object
53   ba_001  5000 non-null   object
54   ba_002  5000 non-null   object
55   ba_003  5000 non-null   object
56   ba_004  5000 non-null   object
57   ba_005  5000 non-null   object
58   ba_006  5000 non-null   object
59   ba_007  5000 non-null   object
60   ba_008  5000 non-null   object
61   ba_009  5000 non-null   object
62   bb_000  5000 non-null   object
63   bc_000  5000 non-null   object
64   bd_000  5000 non-null   object
65   be_000  5000 non-null   object
66   bf_000  5000 non-null   object
67   bg_000  5000 non-null   object
68   bh_000  5000 non-null   object
69   bi_000  5000 non-null   object
70   bj_000  5000 non-null   object
71   bk_000  5000 non-null   object
72   bl_000  5000 non-null   object
73   bm_000  5000 non-null   object
74   bn_000  5000 non-null   object
75   bo_000  5000 non-null   object
76   bp_000  5000 non-null   object
77   bq_000  5000 non-null   object
78   br_000  5000 non-null   object
79   bs_000  5000 non-null   object
80   bt_000  5000 non-null   object
81   bu_000  5000 non-null   object
82   bv_000  5000 non-null   object
83   bx_000  5000 non-null   object
84   by_000  5000 non-null   object
85   bz_000  5000 non-null   object
86   ca_000  5000 non-null   object
87   cb_000  5000 non-null   object
88   cc_000  5000 non-null   object
89   cd_000  5000 non-null   object
90   ce_000  5000 non-null   object
91   cf_000  5000 non-null   object
92   cg_000  5000 non-null   object
93   ch_000  5000 non-null   object
94   ci_000  5000 non-null   object
95   cj_000  5000 non-null   object
96   ck_000  5000 non-null   object
97   cl_000  5000 non-null   object
98   cm_000  5000 non-null   object
99   cn_000  5000 non-null   object
100  cn_001  5000 non-null   object
101  cn_002  5000 non-null   object
102  cn_003  5000 non-null   object
103  cn_004  5000 non-null   object
104  cn_005  5000 non-null   object
105  cn_006  5000 non-null   object
106  cn_007  5000 non-null   object
107  cn_008  5000 non-null   object
108  cn_009  5000 non-null   object
109  co_000  5000 non-null   object
110  cp_000  5000 non-null   object
111  cq_000  5000 non-null   object
112  cr_000  5000 non-null   object
113  cs_000  5000 non-null   object
114  cs_001  5000 non-null   object
115  cs_002  5000 non-null   object
116  cs_003  5000 non-null   object
117  cs_004  5000 non-null   object
118  cs_005  5000 non-null   object
119  cs_006  5000 non-null   object
120  cs_007  5000 non-null   object
121  cs_008  5000 non-null   object
122  cs_009  5000 non-null   object
123  ct_000  5000 non-null   object
124  cu_000  5000 non-null   object
125  cv_000  5000 non-null   object
126  cx_000  5000 non-null   object
127  cy_000  5000 non-null   object
128  cz_000  5000 non-null   object
129  da_000  5000 non-null   object
130  db_000  5000 non-null   object
131  dc_000  5000 non-null   object
132  dd_000  5000 non-null   object
133  de_000  5000 non-null   object
134  df_000  5000 non-null   object
```

```
 135  dg_000  5000 non-null   object
 136  dh_000  5000 non-null   object
 137  di_000  5000 non-null   object
 138  dj_000  5000 non-null   object
 139  dk_000  5000 non-null   object
 140  dl_000  5000 non-null   object
 141  dm_000  5000 non-null   object
 142  dn_000  5000 non-null   object
 143  do_000  5000 non-null   object
 144  dp_000  5000 non-null   object
 145  dq_000  5000 non-null   object
 146  dr_000  5000 non-null   object
 147  ds_000  5000 non-null   object
 148  dt_000  5000 non-null   object
 149  du_000  5000 non-null   object
 150  dv_000  5000 non-null   object
 151  dx_000  5000 non-null   object
 152  dy_000  5000 non-null   object
 153  dz_000  5000 non-null   object
 154  ea_000  5000 non-null   object
 155  eb_000  5000 non-null   object
 156  ec_00   5000 non-null   object
 157  ed_000  5000 non-null   object
 158  ee_000  5000 non-null   object
 159  ee_001  5000 non-null   object
 160  ee_002  5000 non-null   object
 161  ee_003  5000 non-null   object
 162  ee_004  5000 non-null   object
 163  ee_005  5000 non-null   object
 164  ee_006  5000 non-null   object
 165  ee_007  5000 non-null   object
 166  ee_008  5000 non-null   object
 167  ee_009  5000 non-null   object
 168  ef_000  5000 non-null   object
 169  eg_000  5000 non-null   object
dtypes: int64(1), object(169)
memory usage: 6.5+ MB
```

In [10]:
```
y=pd.get_dummies(y,drop_first=True)
```

In [11]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

In [12]:
```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(4000, 170)
(1000, 170)
(4000, 1)
(1000, 1)
```

In [13]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
```

In [14]:
```
# Importing the Keras libraries and packages
import tensorflow.keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Dropout
```

In [15]:
```
classifier=Sequential()

classifier.add(Dense(units=10,kernel_initializer='he_uniform',activation='relu',input_dim =170))

classifier.add(Dense(units = 10, kernel_initializer = 'he_uniform',activation='relu'))

classifier.add(Dense(units = 1, kernel_initializer = 'glorot_uniform', activation = 'sigmoid'))
```

In [16]:
```
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy',metrics=['accuracy'])
```

In [17]:
```
classifier.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 10)                1710

 dense_1 (Dense)             (None, 10)                110

 dense_2 (Dense)             (None, 1)                 11

=================================================================
Total params: 1,831
Trainable params: 1,831
Non-trainable params: 0
_________________________________________________________________
```

In [44]:
```
X_train
```

Out[44]:

	aa_000	ab_000	ac_000	ad_000	ae_000	af_000	ag_000	ag_001	ag_002	ag_003	...	ee_002	ee_003	ee_004	ee_005	ee_006
2913	366	0.0	40	38	0	0	0	0	0	0	...	1468	288	1090	13010	70
3275	31098	0.0	740	636	0	0	0	0	0	0	...	268066	130198	305668	389780	244054
775	30472	0.0	1226	1124	0	0	0	0	0	0	...	611900	268978	510700	455730	295016
217	39728	0.0	2130706432	0	0	0	0	0	0	0	...	476254	220284	527760	462754	103530
1245	31578	0.0	0	136.0	0	0	0	0	0	0	...	190842	95130	228770	286128	288934
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4931	40798	0.0	2130706432	364	0	0	0	0	0	0	...	266738	133304	277078	248244	182874
3264	40692	0.0	4500	3402	0	0	0	0	0	0	...	354658	111084	232294	232178	288616
1653	88764	0.0	156.0	136.0	0.0	0.0	0	0	0	0	...	834538	421958	924668	770782	567650
2607	61756	0.0	0	136.0	0	0	0	0	0	0	...	533474	337336	878310	601734	295580
2732	242	0	10	6	0	0	0	0	0	0	...	4152	1080	5192	38	64

4000 rows × 170 columns

In [18]:
```
model_history=classifier.fit(X_train,y_train,batch_size=10,epochs=30,validation_split=0.25)
```

```
Epoch 1/30
300/300 [==============================] - 4s 5ms/step - loss: 0.2309 - accuracy: 0.9487 - val_loss: 0.0884 - v
al_accuracy: 0.9850
Epoch 2/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0659 - accuracy: 0.9840 - val_loss: 0.0606 - v
al_accuracy: 0.9840
Epoch 3/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0412 - accuracy: 0.9850 - val_loss: 0.0489 - v
al_accuracy: 0.9860
Epoch 4/30
300/300 [==============================] - 1s 5ms/step - loss: 0.0322 - accuracy: 0.9880 - val_loss: 0.0473 - v
al_accuracy: 0.9860
Epoch 5/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0233 - accuracy: 0.9910 - val_loss: 0.0559 - v
al_accuracy: 0.9830
Epoch 6/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0192 - accuracy: 0.9927 - val_loss: 0.0529 - v
al_accuracy: 0.9850
Epoch 7/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0153 - accuracy: 0.9937 - val_loss: 0.0522 - v
al_accuracy: 0.9870
Epoch 8/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0161 - accuracy: 0.9940 - val_loss: 0.0671 - v
al_accuracy: 0.9890
Epoch 9/30
300/300 [==============================] - 1s 5ms/step - loss: 0.0152 - accuracy: 0.9937 - val_loss: 0.0669 - v
al_accuracy: 0.9850
Epoch 10/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0115 - accuracy: 0.9960 - val_loss: 0.0668 - v
al_accuracy: 0.9880
Epoch 11/30
300/300 [==============================] - 2s 6ms/step - loss: 0.0104 - accuracy: 0.9963 - val_loss: 0.0692 - v
al_accuracy: 0.9880
Epoch 12/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0093 - accuracy: 0.9967 - val_loss: 0.0845 - v
al_accuracy: 0.9880
Epoch 13/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0093 - accuracy: 0.9967 - val_loss: 0.0737 - v
al_accuracy: 0.9880
Epoch 14/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0090 - accuracy: 0.9970 - val_loss: 0.0833 - v
al_accuracy: 0.9880
Epoch 15/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0083 - accuracy: 0.9973 - val_loss: 0.0879 - v
```

```
al_accuracy: 0.9880
Epoch 16/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0061 - accuracy: 0.9977 - val_loss: 0.0866 - v
al_accuracy: 0.9900
Epoch 17/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0053 - accuracy: 0.9983 - val_loss: 0.1012 - v
al_accuracy: 0.9870
Epoch 18/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0079 - accuracy: 0.9967 - val_loss: 0.0996 - v
al_accuracy: 0.9870
Epoch 19/30
300/300 [==============================] - 1s 5ms/step - loss: 0.0065 - accuracy: 0.9963 - val_loss: 0.0973 - v
al_accuracy: 0.9880
Epoch 20/30
300/300 [==============================] - 1s 5ms/step - loss: 0.0052 - accuracy: 0.9983 - val_loss: 0.1048 - v
al_accuracy: 0.9870
Epoch 21/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0032 - accuracy: 0.9987 - val_loss: 0.1291 - v
al_accuracy: 0.9860
Epoch 22/30
300/300 [==============================] - 2s 7ms/step - loss: 0.0028 - accuracy: 0.9987 - val_loss: 0.1355 - v
al_accuracy: 0.9860
Epoch 23/30
300/300 [==============================] - 3s 9ms/step - loss: 0.0163 - accuracy: 0.9963 - val_loss: 0.1230 - v
al_accuracy: 0.9860
Epoch 24/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0100 - accuracy: 0.9970 - val_loss: 0.1267 - v
al_accuracy: 0.9890
Epoch 25/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0048 - accuracy: 0.9973 - val_loss: 0.1246 - v
al_accuracy: 0.9870
Epoch 26/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0025 - accuracy: 0.9987 - val_loss: 0.1262 - v
al_accuracy: 0.9880
Epoch 27/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0022 - accuracy: 0.9987 - val_loss: 0.1333 - v
al_accuracy: 0.9880
Epoch 28/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0017 - accuracy: 0.9993 - val_loss: 0.1384 - v
al_accuracy: 0.9850
Epoch 29/30
300/300 [==============================] - 1s 5ms/step - loss: 0.0017 - accuracy: 0.9993 - val_loss: 0.1451 - v
al_accuracy: 0.9840
Epoch 30/30
300/300 [==============================] - 1s 4ms/step - loss: 0.0021 - accuracy: 0.9997 - val_loss: 0.1389 - v
al_accuracy: 0.9870
```

In [19]:
```
model_history.history.keys()
```

Out[19]:
```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```

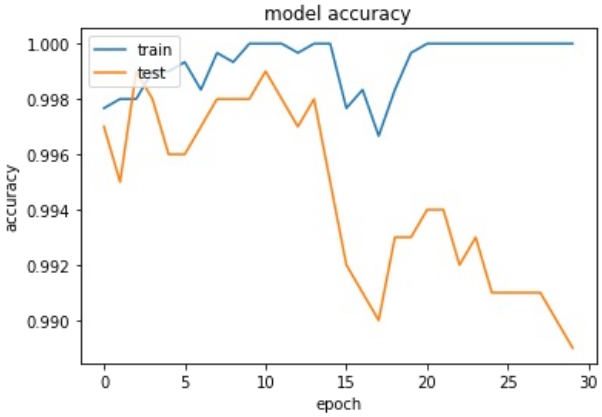
In [51]:
```
model_history.history['accuracy']
```

Out[51]:
```
[0.9976666569709778,
 0.9980000257492065,
 0.9980000257492065,
 0.9990000128746033,
 0.9990000128746033,
 0.9993333220481873,
 0.9983333349227905,
 0.999666690826416,
 0.9993333220481873,
 1.0,
 1.0,
 1.0,
 0.999666690826416,
 1.0,
 1.0,
 0.9976666569709778,
 0.9983333349227905,
 0.996666669845581,
 0.9983333349227905,
 0.999666690826416,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0,
 1.0]
```

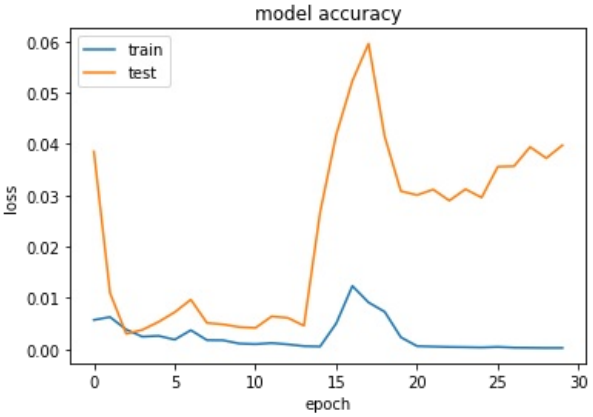
In [53]:
```
# list all data in history

print(model_history.history.keys())
# summarize history for accuracy
```

```
plt.plot(model_history.history['accuracy'])
plt.plot(model_history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

```
dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])
```



In [56]:
```
# summarize history for accuracy
plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('model accuracy')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



In [71]:
```
X_test.shape
```

Out[71]:
```
(1000, 170)
```

In [72]:
```
y_pred=classifier.predict(X_test)
```

```
32/32 [==============================] - 0s 2ms/step
```

In [ ]:
```
#for the sigmoid my threshold is 0.
```

In [73]:
```
y_pred=y_pred>0.5
```

In [62]:
```
from sklearn.metrics import confusion_matrix
```

In [64]:
```
y_test.shape
```

Out[64]:
```
(1000, 1)
```

In [74]:
```
y_pred.shape
```

Out[74]:
```
(1000, 1)
```

In [75]:
```
confusion_matrix(y_test,y_pred)
```

Out[75]:
```
array([[972,  14],
       [  4,  10]])
```

In [20]:
```
import time
import os

def saveModel_path(model_dir="SAVED_MODELS"):
```

```
    os.makedirs(model_dir, exist_ok=True)
    fileName = time.strftime("Model_%Y_%m_%d_%H_%M_%S_.h5")
    model_path = os.path.join(model_dir, fileName)
    print(f"your model will be saved at the following location\n{model_path}")
    return model_path
```

In [22]:
```
classifier.save(saveModel_path())
```

```
your model will be saved at the following location
SAVED_MODELS/Model_2023_01_21_12_32_30_.h5
```

In [76]:
```
#hyperparameter tunig with keras tuner

!pip install keras-tuner --upgrade
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting keras-tuner
  Downloading keras_tuner-1.1.3-py3-none-any.whl (135 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.7/135.7 KB 9.6 MB/s eta 0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.8/dist-packages (from keras-tuner) (21.3)
Requirement already satisfied: tensorboard in /usr/local/lib/python3.8/dist-packages (from keras-tuner) (2.9.1)
Collecting kt-legacy
  Downloading kt_legacy-1.0.4-py3-none-any.whl (9.6 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.8/dist-packages (from keras-tuner) (2.25.1)
Requirement already satisfied: ipython in /usr/local/lib/python3.8/dist-packages (from keras-tuner) (7.9.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from keras-tuner) (1.21.6)
Requirement already satisfied: decorator in /usr/local/lib/python3.8/dist-packages (from ipython->keras-tuner)
(4.4.2)
Collecting jedi>=0.10
  Downloading jedi-0.18.2-py2.py3-none-any.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 60.2 MB/s eta 0:00:00
Requirement already satisfied: pygments in /usr/local/lib/python3.8/dist-packages (from ipython->keras-tuner) (
2.6.1)
Requirement already satisfied: backcall in /usr/local/lib/python3.8/dist-packages (from ipython->keras-tuner) (
0.2.0)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.8/dist-packages (from ipython->keras-tu
ner) (5.7.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.8/dist-packages (from ipython->keras-
tuner) (57.4.0)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from ipy
thon->keras-tuner) (2.0.10)
Requirement already satisfied: pexpect in /usr/local/lib/python3.8/dist-packages (from ipython->keras-tuner) (4
.8.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.8/dist-packages (from ipython->keras-tuner
) (0.7.5)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from packagi
ng->keras-tuner) (3.0.9)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from requests->ker
as-tuner) (2022.12.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from requests->
keras-tuner) (1.24.3)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from requests->kera
s-tuner) (4.0.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from requests->keras-tun
er) (2.10)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from tensorboard->ker
as-tuner) (1.0.1)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.8/dist-packages (from
tensorboard->keras-tuner) (0.4.6)
Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.8/dist-packages (from tensorboard->keras-
tuner) (1.3.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.8/dist-packages (from tensorboard->ker
as-tuner) (3.4.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.8/dist-packages (from te
nsorboard->keras-tuner) (1.8.1)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.8/dist-packages
(from tensorboard->keras-tuner) (0.6.1)
Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.8/dist-packages (from tensorboard->keras-t
uner) (0.38.4)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.8/dist-packages (from tensorboar
d->keras-tuner) (2.16.0)
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.8/dist-packages (from tensorboard->kera
s-tuner) (1.51.1)
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.8/dist-packages (from tensorboar
d->keras-tuner) (3.19.6)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from google-aut
h<3,>=1.6.3->tensorboard->keras-tuner) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.
6.3->tensorboard->keras-tuner) (4.9)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.8/dist-packages (from google-auth<3,>=1.6.3
->tensorboard->keras-tuner) (1.15.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from google-au
th<3,>=1.6.3->tensorboard->keras-tuner) (5.2.1)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from google-
auth-oauthlib<0.5,>=0.4.1->tensorboard->keras-tuner) (1.3.1)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.8/dist-packages (from jedi>=0.10->
ipython->keras-tuner) (0.8.3)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.8/dist-packages (from markdown
>=2.6.8->tensorboard->keras-tuner) (6.0.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.8/dist-packages (from prompt-toolkit<2.1.0,>=2
.0.0->ipython->keras-tuner) (0.2.5)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.8/dist-packages (from pexpect->ipython
->keras-tuner) (0.7.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from importlib-metadata>=4.
4->markdown>=2.6.8->tensorboard->keras-tuner) (3.11.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.8/dist-packages (from pyasn1-modu
les>=0.2.1->google-auth<3,>=1.6.3->tensorboard->keras-tuner) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from requests-oauthli
b>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard->keras-tuner) (3.2.2)
Installing collected packages: kt-legacy, jedi, keras-tuner
Successfully installed jedi-0.18.2 keras-tuner-1.1.3 kt-legacy-1.0.4
```

In [77]:

```
from kerastuner.tuners import RandomSearch
```

In [78]:
```
def build_model(hp):
  model=Sequential()
  model.add(Dense(units=32,kernel_initializer = 'he_uniform',activation='relu',input_dim=170))
  model.add(Dense(units = 32, kernel_initializer = 'he_uniform',activation='relu'))
  model.add(Dense(1,activation='sigmoid'))
  optimizer=hp.Choice("optimizer",values=["adam","sgd","rmsprop","adadelta"])
  model.compile(optimizer=optimizer,loss='binary_crossentropy',metrics=['accuracy'])
  return model
```

In [87]:
```
tuner=RandomSearch(build_model,objective='val_accuracy',max_trials=5,directory='project1',project_name='aps-fau
```

In [ ]:

In [88]:
```
tuner.search(X_train,y_train,validation_data=(X_test,y_test),epochs=5)
```

```
Trial 4 Complete [00h 00m 06s]
val_accuracy: 0.9909999966621399

Best val_accuracy So Far: 0.9909999966621399
Total elapsed time: 00h 00m 23s
```

In [83]:
```
tuner.get_best_hyperparameters()[0].values
```

Out[83]:
```
{'optimizer': 'adam'}
```

In [85]:
```
mymodel=tuner.get_best_models(num_models=1)[0]
```

In [86]:
```
mymodel.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32)                5472

 dense_1 (Dense)             (None, 32)                1056

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 6,561
Trainable params: 6,561
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

In [90]:
```
for i in range( 2, 20):
  print(i)
```

```
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

In [ ]:
```
32
33
34
35
36
....512
```

In [ ]:
```
32,64,128,256,512
```

In [105…
```
from tensorflow.keras import optimizers
def build_model(hp):
    model = Sequential()
    for i in range(hp.Int('num_layers', 2, 20)):
        model.add(Dense(units=hp.Int('units_' + str(i),
```

```
                                        min_value=32,
                                        max_value=512,
                                        step=32),
                                activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(
        optimizer=optimizers.Adam(
            hp.Choice('learning_rate', [1e-2, 1e-3, 1e-4])),
        loss='binary_crossentropy',
        metrics=['accuracy'])
    return model
```

In [106…
```
tuner_second=RandomSearch(build_model,objective='val_accuracy',max_trials=5,directory='project3',project_name='
```

In [107…
```
tuner_second.search(X_train,y_train,epochs=5,validation_data=(X_test,y_test))
```

```
Trial 5 Complete [00h 00m 06s]
val_accuracy: 0.9860000014305115

Best val_accuracy So Far: 0.9940000176429749
Total elapsed time: 00h 00m 33s
```

In [108…
```
tuner_second.get_best_hyperparameters()[0].values
```

Out[108]:
```
{'num_layers': 20,
 'units_0': 352,
 'units_1': 512,
 'learning_rate': 0.0001,
 'units_2': 416,
 'units_3': 224,
 'units_4': 128,
 'units_5': 256,
 'units_6': 64,
 'units_7': 352,
 'units_8': 64,
 'units_9': 512,
 'units_10': 128,
 'units_11': 288,
 'units_12': 256,
 'units_13': 352,
 'units_14': 128,
 'units_15': 32,
 'units_16': 32,
 'units_17': 32,
 'units_18': 32,
 'units_19': 32}
```

In [109…
```
classifier2=tuner_second.get_best_models(num_models=1)[0]
```

```
WARNING:tensorflow:Detecting that an object or model or tf.train.Checkpoint is being deleted with unrestored va
lues. See the following logs for the specific values in question. To silence these warnings, use `status.expect
_partial()`. See https://www.tensorflow.org/api_docs/python/tf/train/Checkpoint#restorefor details about the st
atus object returned by the restore function.
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-0.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-0.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-1.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-1.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-2.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-2.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-3.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-3.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-4.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-4.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-5.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-5.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-6.k
ernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).layer_with_weights-6.b
ias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.iter
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.beta_1
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.beta_2
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.decay
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer.learning_rat
e
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-0.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-0.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-1.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-1.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-2.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-2.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-3.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-3.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-4.kernel
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-4.bias
WARNING:tensorflow:Value in checkpoint could not be found in the restored object: (root).optimizer's state 'm'
for (root).layer_with_weights-5.kernel
```

In [110…

```
classifier2.summary()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-110-6be4506ce783> in <module>
----> 1 classifier2.summary()

/usr/local/lib/python3.8/dist-packages/keras/engine/training.py in summary(self, line_length, positions, print_
fn, expand_nested, show_trainable)
   2867     """
   2868     if not self.built:
-> 2869       raise ValueError(
   2870           'This model has not yet been built. '
   2871           'Build the model first by calling `build()` or by calling '

ValueError: This model has not yet been built. Build the model first by calling `build()` or by calling the mod
el on a batch of data.
```

In [ ]: