problem Srtatement - PREDICT THE BURNED AREA OF FOREST FIRES WITH NEURAL NETWORKS

.1 Import neccessary Libraries

In [1]:
```
import pandas as pd
from matplotlib import pyplot as plt
```

2. Import data

In [2]:
```
Fire_forest = pd.read_csv('forestfires.csv')
Fire_forest
```

Out[2]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	m
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0	
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0	
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0	0	
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	...	0	0	
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
512	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	...	0	0	
513	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	...	0	0	
514	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	...	0	0	
515	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	...	0	0	
516	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0		0	0	

3. Data Understanding

In [3]:
```
Fire_forest.shape
```

Out[3]: (517, 31)

In [4]:
```
Fire_forest.dtypes
```

Out[4]:
```
month             object
day               object
FFMC             float64
DMC              float64
DC               float64
ISI              float64
temp             float64
RH                 int64
wind             float64
rain             float64
area             float64
dayfri             int64
daymon             int64
daysat             int64
daysun             int64
daythu             int64
daytue             int64
daywed             int64
monthapr           int64
monthaug           int64
monthdec           int64
monthfeb           int64
monthjan           int64
monthjul           int64
monthjun           int64
monthmar           int64
monthmay           int64
monthnov           int64
monthoct           int64
monthsep           int64
size_category     object
dtype: object
```

In [5]:
```
Fire_forest.isna().sum()
```

Out[5]:
```
month            0
day              0
FFMC             0
DMC              0
DC               0
ISI              0
temp             0
RH               0
wind             0
rain             0
area             0
dayfri           0
daymon           0
daysat           0
daysun           0
daythu           0
daytue           0
daywed           0
monthapr         0
monthaug         0
monthdec         0
monthfeb         0
monthjan         0
monthjul         0
monthjun         0
monthmar         0
monthmay         0
monthnov         0
monthoct         0
monthsep         0
size_category    0
dtype: int64
```

In [7]:
```
Fire_forest.describe()
```

Out[7]:

	FFMC	DMC	DC	ISI	temp	RH	wind	
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.00
mean	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.02
std	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.29
min	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.00
25%	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.00
50%	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.00
75%	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.00
max	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	6.40

8 rows × 28 columns

4. Data Preparation

In [11]:
```
x = Fire_forest.iloc[:,:11]
x
```

Out[11]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...
512	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 11 columns

In [22]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [33]:
```
x['month']=le.fit_transform(x['month'])
x['day']=le.fit_transform(x['day'])
```

In [26]:
```
x
```

Out[26]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	0	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00
1	10	5	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	10	2	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	7	0	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	7	3	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
...	...	...	...	...	...	...	...	...	...	...	...
512	1	3	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	1	3	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	1	3	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	2	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	9	5	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 11 columns

In [27]:
```
x.dtypes
```

Out[27]:
```
month      int32
day        int32
FFMC     float64
DMC      float64
DC       float64
ISI      float64
temp     float64
RH         int64
wind     float64
rain     float64
area     float64
dtype: object
```

In [29]:
```
x.shape
```

Out[29]:
```
(517, 11)
```

```
In [30]: y = Fire_forest['size_category']
         y
```

```
Out[30]: 0      small
         1      small
         2      small
         3      small
         4      small
                ...
         512    large
         513    large
         514    large
         515    small
         516    small
         Name: size_category, Length: 517, dtype: object
```

```
In [35]: y = Fire_forest['size_category']=le.fit_transform(Fire_forest['size_category'])
         y
```

```
Out[35]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 1, 1, 1, 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, 1, 1, 1,
                0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1,
                0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
                1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
                0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0,
                1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1,
                0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0,
                1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1,
                1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1,
                1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
                1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
                0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1,
                1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1], dtype=int64)
```

In [37]:
```
y = pd.DataFrame(y)
y
```

Out[37]:

	0
0	1
1	1
2	1
3	1
4	1
...	...
512	0
513	0
514	0
515	1
516	1

517 rows × 1 columns

In [38]:
```
y.dtypes
```

Out[38]:
```
0    int64
dtype: object
```

In [39]:
```
from sklearn.preprocessing import StandardScaler
std_scalar = StandardScaler()
```

In [41]:
```
scaled_x = std_scalar.fit_transform(x)
scaled_x
```

Out[41]:
```
array([[ 0.28422225, -1.42312073, -0.80595947, ...,  1.49861442,
        -0.07326831, -0.20201979],
       [ 0.97087134,  1.17671466, -0.00810203, ..., -1.74175564,
        -0.07326831, -0.20201979],
       [ 0.97087134, -0.38318657, -0.00810203, ..., -1.51828184,
        -0.07326831, -0.20201979],
       ...,
       [-1.08907592,  0.13678051, -1.64008316, ...,  1.49861442,
        -0.07326831, -0.02653216],
       [-1.08907592, -0.38318657,  0.68095666, ..., -0.00983371,
        -0.07326831, -0.20201979],
       [ 0.74198831,  1.17671466, -2.02087875, ...,  0.26950853,
        -0.07326831, -0.20201979]])
```

In [43]:
```
scaled_x = pd.DataFrame(scaled_x)
scaled_x
```

Out[43]:

	0	1	2	3	4	5	6	7	
0	0.284222	-1.423121	-0.805959	-1.323326	-1.830477	-0.860946	-1.842640	0.411724	1.4986
1	0.970871	1.176715	-0.008102	-1.179541	0.488891	-0.509688	-0.153278	-0.692456	-1.74175
2	0.970871	-0.383187	-0.008102	-1.049822	0.560715	-0.509688	-0.739383	-0.692456	-1.51828
3	0.284222	-1.423121	0.191362	-1.212361	-1.898266	-0.004756	-1.825402	3.233519	-0.00983
4	0.284222	0.136781	-0.243833	-0.931043	-1.798600	0.126966	-1.291012	3.356206	-1.23894
...	...	...	...	...	...	...	...	...	
512	-1.089076	0.136781	-1.640083	-0.846648	0.474768	-1.563460	1.536084	-0.753800	-0.73612
513	-1.089076	0.136781	-1.640083	-0.846648	0.474768	-1.563460	0.519019	1.638592	0.99579
514	-1.089076	0.136781	-1.640083	-0.846648	0.474768	-1.563460	0.398350	1.577248	1.4986
515	-1.089076	-0.383187	0.680957	0.549003	0.269382	0.500176	1.156839	-0.140366	-0.00983
516	0.741988	1.176715	-2.020879	-1.685913	-1.780442	-1.739089	-1.222058	-0.815143	0.2695

517 rows × 11 columns

In [44]:
```
scaled_x.dtypes
```

Out[44]:
```
0     float64
1     float64
2     float64
3     float64
4     float64
5     float64
6     float64
7     float64
8     float64
9     float64
10    float64
dtype: object
```

In [47]:
```
scaled_x['y']=y
Fire_forest = scaled_x
Fire_forest
```

Out[47]:

	0	1	2	3	4	5	6	7	
0	0.284222	-1.423121	-0.805959	-1.323326	-1.830477	-0.860946	-1.842640	0.411724	1.4986
1	0.970871	1.176715	-0.008102	-1.179541	0.488891	-0.509688	-0.153278	-0.692456	-1.7417
2	0.970871	-0.383187	-0.008102	-1.049822	0.560715	-0.509688	-0.739383	-0.692456	-1.5182
3	0.284222	-1.423121	0.191362	-1.212361	-1.898266	-0.004756	-1.825402	3.233519	-0.0098
4	0.284222	0.136781	-0.243833	-0.931043	-1.798600	0.126966	-1.291012	3.356206	-1.2389
...	...	...	...	...	...	...	...	...	
512	-1.089076	0.136781	-1.640083	-0.846648	0.474768	-1.563460	1.536084	-0.753800	-0.7361
513	-1.089076	0.136781	-1.640083	-0.846648	0.474768	-1.563460	0.519019	1.638592	0.9957
514	-1.089076	0.136781	-1.640083	-0.846648	0.474768	-1.563460	0.398350	1.577248	1.4986
515	-1.089076	-0.383187	0.680957	0.549003	0.269382	0.500176	1.156839	-0.140366	-0.0098
516	0.741988	1.176715	-2.020879	-1.685913	-1.780442	-1.739089	-1.222058	-0.815143	0.2695

517 rows × 12 columns

5. Model Building

In [48]:
```
pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\nandini\anaconda3\lib\sit
e-packages (2.7.0)Note: you may need to restart the kernel to use updated packa
ges.

Requirement already satisfied: libclang>=9.0.1 in c:\users\nandini\anaconda3\li
b\site-packages (from tensorflow) (12.0.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\nandini\anaconda
3\lib\site-packages (from tensorflow) (1.41.1)
Requirement already satisfied: h5py>=2.9.0 in c:\users\nandini\anaconda3\lib\si
te-packages (from tensorflow) (2.10.0)
Requirement already satisfied: absl-py>=0.4.0 in c:\users\nandini\anaconda3\lib
\site-packages (from tensorflow) (1.0.0)
Requirement already satisfied: wheel<1.0,>=0.32.0 in c:\users\nandini\anaconda3
\lib\site-packages (from tensorflow) (0.36.2)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\nandini\anaconda3
\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: protobuf>=3.9.2 in c:\users\nandini\anaconda3\li
b\site-packages (from tensorflow) (3.19.1)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0rc0 in c:\users
\nandini\anaconda3\lib\site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: numpy>=1.14.5 in c:\users\nandini\anaconda3\lib
\site-packages (from tensorflow) (1.20.1)
Requirement already satisfied: six>=1.12.0 in c:\users\nandini\anaconda3\lib\si
te-packages (from tensorflow) (1.15.0)
Requirement already satisfied: tensorboard~=2.6 in c:\users\nandini\anaconda3\l
ib\site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in c:\users
\nandini\anaconda3\lib\site-packages (from tensorflow) (0.22.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\nandini\ana
conda3\lib\site-packages (from tensorflow) (3.7.4.3)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in c:\users\nandini\anaco
nda3\lib\site-packages (from tensorflow) (2.0)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in c:\users\nandini\anaconda3
\lib\site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\nandini\anaconda
3\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\nandini\anaconda3\lib
\site-packages (from tensorflow) (1.12.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\nandini\anaconda3
\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\nandini\a
naconda3\lib\site-packages (from tensorflow) (1.1.2)
Requirement already satisfied: keras<2.8,>=2.7.0rc0 in c:\users\nandini\anacond
a3\lib\site-packages (from tensorflow) (2.7.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\nandini\anaconda3\l
ib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in c:\users\nandini\anacon
da3\lib\site-packages (from tensorboard~=2.6->tensorflow) (2.3.3)
Requirement already satisfied: markdown>=2.6.8 in c:\users\nandini\anaconda3\li
b\site-packages (from tensorboard~=2.6->tensorflow) (3.3.5)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in c:\users\nan
dini\anaconda3\lib\site-packages (from tensorboard~=2.6->tensorflow) (0.4.6)
Requirement already satisfied: setuptools>=41.0.0 in c:\users\nandini\anaconda3
\lib\site-packages (from tensorboard~=2.6->tensorflow) (52.0.0.post20210125)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in c:\user
```

```
s\nandini\anaconda3\lib\site-packages (from tensorboard~=2.6->tensorflow) (0.6.
1)
Requirement already satisfied: werkzeug>=0.11.15 in c:\users\nandini\anaconda3
\lib\site-packages (from tensorboard~=2.6->tensorflow) (1.0.1)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\nandini\anaconda
3\lib\site-packages (from tensorboard~=2.6->tensorflow) (2.25.1)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in c:\users\nandin
i\anaconda3\lib\site-packages (from tensorboard~=2.6->tensorflow) (1.8.0)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in c:\users\nandini\anaco
nda3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflo
w) (4.2.4)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\nandini\anaconda3\lib
\site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.7.
2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\nandini\anacon
da3\lib\site-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflo
w) (0.2.8)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\nandini\ana
conda3\lib\site-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=
2.6->tensorflow) (1.3.0)
Requirement already satisfied: importlib-metadata>='4.4' in c:\users\nandini\an
aconda3\lib\site-packages (from markdown>=2.6.8->tensorboard~=2.6->tensorflow)
(3.10.0)
Requirement already satisfied: zipp>=0.5 in c:\users\nandini\anaconda3\lib\site
-packages (from importlib-metadata>='4.4'->markdown>=2.6.8->tensorboard~=2.6->t
ensorflow) (3.4.1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\nandini\anacond
a3\lib\site-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensor
board~=2.6->tensorflow) (0.4.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\nandini\anacon
da3\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow)
(1.26.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\nandini\anaconda3\lib\s
ite-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\nandini\anaconda3
\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (4.
0.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nandini\anaconda3
\lib\site-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (20
20.12.5)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\nandini\anaconda3\li
b\site-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.
1->tensorboard~=2.6->tensorflow) (3.1.1)
```

In [49]:
```
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Layer, Lambda
```

In [54]:
```
from sklearn.model_selection import train_test_split
Train,Test = train_test_split(Fire_forest, train_size=0.20, random_state=1)
```

In [75]:
```
x_train = Fire_forest.iloc[0:361,:10]
y_train = Fire_forest.iloc[0:361,11:]
x_test = Fire_forest.iloc[361:,:10]
y_test = Fire_forest.iloc[361:,11:]
```

In [76]:
```
x_train.shape, y_train.shape
```

Out[76]:
```
((361, 10), (361, 1))
```

In [77]:
```
x_test.shape, y_test.shape
```

Out[77]:
```
((156, 10), (156, 1))
```

In [79]:
```
x_train.std()
```

Out[79]:
```
0    0.990589
1    0.999119
2    0.862874
3    0.677631
4    0.988629
5    1.011203
6    0.909068
7    0.963580
8    1.003026
9    0.184780
dtype: float64
```

In [80]:
```
x_train.mean()
```

Out[80]:
```
0    0.191021
1   -0.050465
2   -0.028797
3   -0.262395
4   -0.039506
5   -0.047021
6   -0.128877
7   -0.093976
8   -0.043571
9   -0.060152
dtype: float64
```

In [81]:
```
x_test.std()
```

Out[81]:
```
0    0.880209
1    0.998666
2    1.264832
3    1.318163
4    1.026341
5    0.971341
6    1.134854
7    1.053960
8    0.992053
9    1.796734
dtype: float64
```

6. Model Training

In [83]:
```
import tensorflow as tf
model = Sequential()
model.add(Dense(units = 20, activation = 'relu'))
model.add(Dense(units = 20, activation = 'tanh'))
model.add(Dense(units = 20, activation = 'softmax'))
```

In [85]:
```
model.compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics=['a
```

In [86]:
```
model.fit(x=x_train,y=y_train, epochs=517)
```

```
12/12 [==============================] - 0s 4ms/step - loss: 0.3189 - accurac
y: 0.8670
Epoch 325/517
12/12 [==============================] - 0s 3ms/step - loss: 0.3175 - accurac
y: 0.8670
Epoch 326/517
12/12 [==============================] - 0s 3ms/step - loss: 0.3173 - accurac
y: 0.8670
Epoch 327/517
12/12 [==============================] - 0s 4ms/step - loss: 0.3182 - accurac
y: 0.8726
Epoch 328/517
12/12 [==============================] - 0s 3ms/step - loss: 0.3173 - accurac
y: 0.8643
Epoch 329/517
12/12 [==============================] - 0s 4ms/step - loss: 0.3167 - accurac
y: 0.8698
Epoch 330/517
12/12 [==============================] - 0s 3ms/step - loss: 0.3156 - accurac
y: 0.8698
```

7. Model Testing

In [88]:
```
y_pred = model.predict(x_test)
y_pred
```

Out[88]:
```
array([[1.91293657e-01, 8.08636844e-01, 9.08430684e-06, ...,
        9.14217162e-06, 2.78853963e-06, 1.33816536e-06],
       [2.00225087e-03, 9.97994542e-01, 2.09989096e-07, ...,
        2.24218212e-07, 1.01446915e-07, 6.93924989e-08],
       [1.51753753e-01, 8.48169565e-01, 5.18468414e-06, ...,
        4.37929839e-06, 3.15870625e-06, 1.80616610e-06],
       ...,
       [4.85631637e-04, 9.99489903e-01, 7.91676257e-07, ...,
        1.64577398e-06, 1.64680853e-06, 8.02588033e-07],
       [7.31402636e-01, 2.68293649e-01, 3.33586795e-05, ...,
        1.77252987e-05, 1.59517258e-05, 5.89966885e-06],
       [6.49711676e-03, 9.93489742e-01, 5.46926287e-07, ...,
        4.66624726e-07, 7.56469660e-07, 4.33037371e-07]], dtype=float32)
```

In [90]:
```
y_pred = pd.DataFrame(y_pred)
y_pred
```

Out[90]:

	0	1	2	3	4	5	6	7	
0	0.191294	0.808637	9.084307e-06	2.015402e-06	2.100445e-06	2.892260e-06	1.834352e-06	1.179939e-06	3.7
1	0.002002	0.997995	2.099891e-07	9.556631e-08	8.330071e-08	9.331300e-08	9.825842e-08	8.043025e-08	3.0
2	0.151754	0.848170	5.184684e-06	5.364324e-06	2.700172e-06	2.002973e-06	3.428590e-06	2.877409e-06	5.0
3	0.306113	0.693803	6.655445e-06	4.348084e-06	4.054340e-06	2.924922e-06	4.293060e-06	2.427266e-06	5.9
4	0.198337	0.801599	4.965500e-06	3.269256e-06	2.931649e-06	2.102216e-06	2.621883e-06	2.181415e-06	5.1
...	...	...	...	...	...	...	...	...	
151	0.000482	0.999504	4.741599e-07	1.248864e-06	3.026239e-07	3.365855e-07	3.716432e-07	3.009342e-07	9.3
152	0.000157	0.999834	2.679266e-07	4.198523e-07	2.223715e-07	2.056934e-07	1.705131e-07	2.607273e-07	1.1
153	0.000486	0.999490	7.916763e-07	1.528645e-06	6.410603e-07	5.638700e-07	4.203350e-07	9.007645e-07	3.2
154	0.731403	0.268294	3.335868e-05	8.958926e-06	8.900718e-06	1.060696e-05	1.602828e-05	6.045027e-06	1.2
155	0.006497	0.993490	5.469263e-07	9.135197e-07	3.469348e-07	2.988962e-07	7.575721e-07	4.082693e-07	8.1

156 rows × 20 columns

8. Model Evaluation

For test data

In [91]:
```
Fire_forest_Result = model.evaluate(x_test,y_test)
Fire_forest_Result
```

```
5/5 [==============================] - 0s 4ms/step - loss: 1.1805 - accuracy:
0.5897
```

Out[91]:
```
[1.1804752349853516, 0.5897436141967773]
```

For train data

In [92]:
```
Fire_forest_Result = model.evaluate(x_train,y_train)
Fire_forest_Result
```

```
12/12 [==============================] - 0s 8ms/step - loss: 0.2411 - accuracy:
0.8947
```

Out[92]:
```
[0.24114875495433807, 0.8947368264198303]
```

In [98]:
```
print('Loss    :', round(Fire_forest_Result[0],2))
print('Accuracy:',round(Fire_forest_Result[1],2))
```

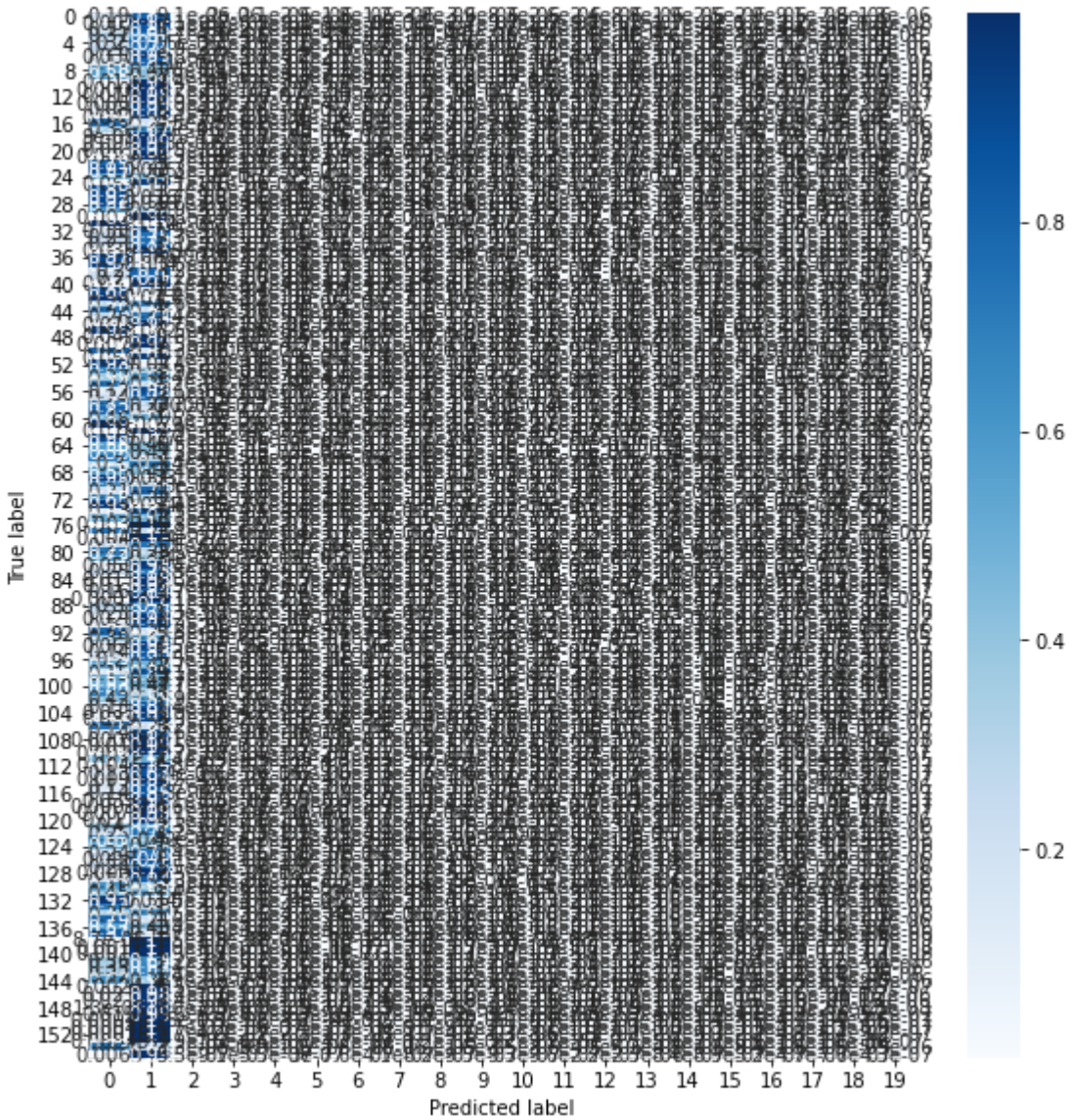
```
Loss    : 0.24
Accuracy: 0.89
```

9 . Model Deployment

In [100]:
```
model.save('Fire_forest.h5')
```

In [101]:
```
import seaborn as sns
```

In [112]:
```
figure = plt.figure(figsize=(8, 8))
sns.heatmap(y_pred, annot=True,cmap=plt.cm.Blues)
plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
```



In [ ]: