Importing the warning module to supressing the warning which is getting generated while running the code

In [86]:

```
import warnings
warnings.filterwarnings("ignore")
```

importing the all the package required for performing array operation,dataframe,plotting.

In [71]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Loading the data-set

In [72]:

```
data=pd.read_csv("mnist_train.csv")
```

considerng the 40000 datapoints out of 42000 datapoints.

In [ ]:

```
dataa=data[0:40000]
```

storing all the output/class/label in output_label dataframe

In [74]:

```
output_label=dataa["label"]
```

displaying the output/class dataframe

In [95]:

```
output_label
```

Out[95]:

```
0        1
1        0
2        1
3        4
4        0
5        0
6        7
7        3
8        5
9        3
10       8
11       9
12       1
13       3
```

```
14       3
15       1
16       2
17       0
18       7
19       5
20       8
21       6
22       2
23       0
24       2
25       3
26       6
27       9
28       9
29       7
        ..
39970    7
39971    1
39972    1
39973    4
39974    7
39975    7
39976    4
39977    2
39978    2
39979    8
39980    9
39981    7
39982    1
39983    2
39984    0
39985    1
39986    3
39987    2
39988    2
39989    6
39990    8
39991    5
39992    2
39993    3
39994    1
39995    0
39996    3
39997    0
39998    0
39999    3
Name: label, Length: 40000, dtype: int64
```

removing the output/class label from the whole dataframe.

In [76]:

```
dataa=dataa.drop("label",axis=1)
```

displaying the whole dataframe after removing output/class label from the whole dataframe.

In [77]:

```
dataa
```

Out[77]:

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel774	pixel775	pixel776	pixel777	pixel778	p
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel774	pixel775	pixel776	pixel777	pixel778	p
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
14	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
16	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
17	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
18	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
20	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
21	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
22	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
23	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
25	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
26	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
27	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
28	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
29	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
39970	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39971	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39972	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39973	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39974	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39975	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39976	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39977	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39978	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39979	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39980	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39981	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39982	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39983	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39984	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39985	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39986	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39987	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39988	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39989	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39990	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39991	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

39992	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel774	pixel775	pixel776	pixel777	pixel778	p
39993	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39994	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
39999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	

40000 rows × 784 columns

Sklearn is package available to write the machine learning algorithm. Hence importing sklearn package and then importing TSNE module from sklearn.

In [78]:

```
from sklearn.manifold import TSNE
```

Setting parameter for TSNE model like dimensions,neighbourhood points,iteration size etc.

In [89]:

```
model_TSNE=TSNE(n_components=2,perplexity=50,learning_rate=1000,n_iter=10000,random_state=0)
```

Transforming the 784 dimentions data to 2 dimensions data using TSNE model.

In [90]:

```
tsne_transformed_data=model_TSNE.fit_transform(dataa)
```

arranging 2 dimensions and output_label row wise.

In [91]:

```
tsne_data_array_format=np.vstack((tsne_transformed_data.T,output_label)).T
```

Loading the data in 3 column of the dataframe tse_transformed_dataframe.

In [92]:

```
tse_transformed_dataframe=pd.DataFrame(data=tsne_data_array_format,columns=("Dim_1","Dim_2","outpu
t"))
```

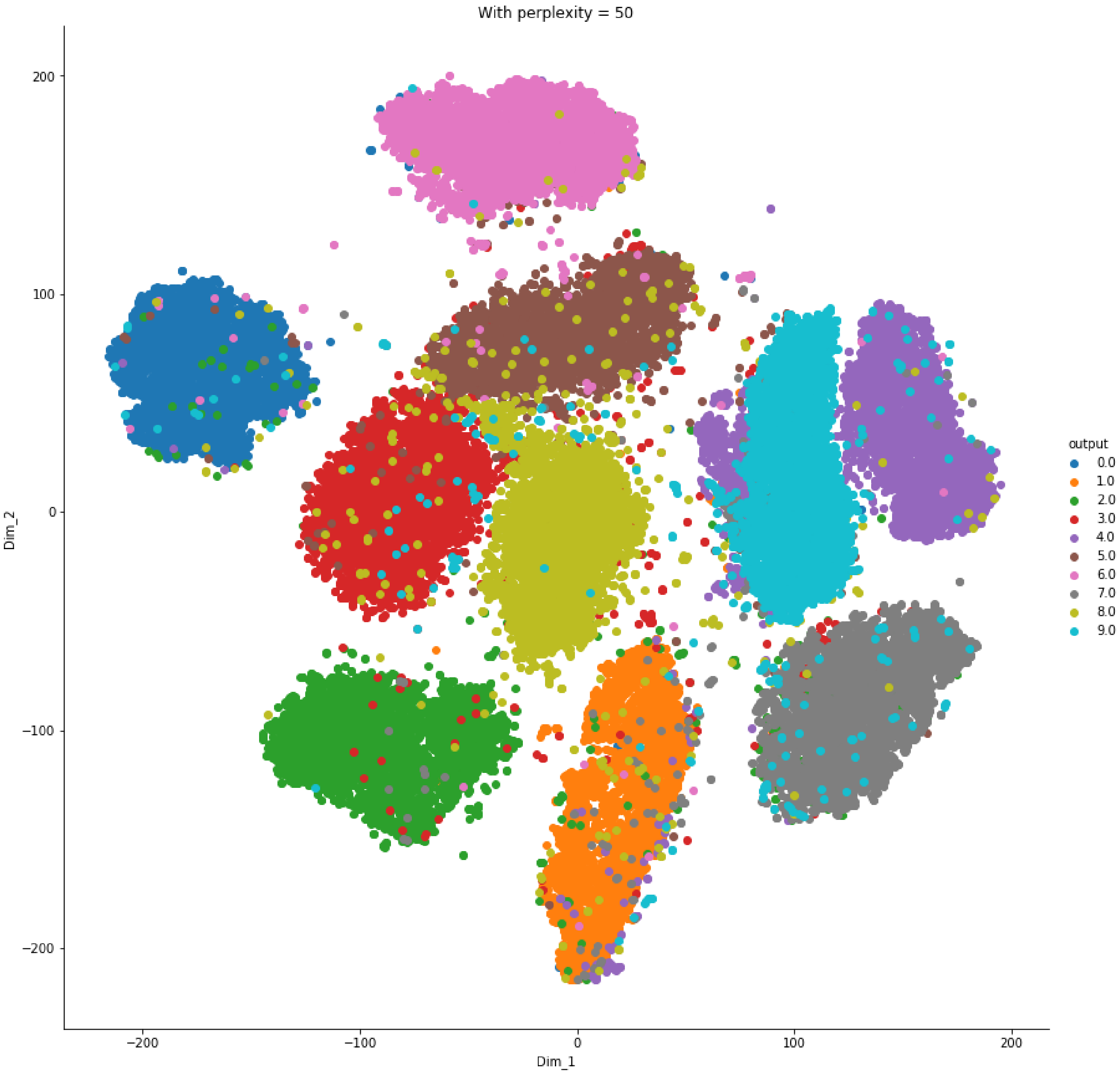
plotting the data using Seaborn

In [93]:

```
import seaborn as sn
```

In [96]:

```
g=sn.FacetGrid(tse_transformed_dataframe,hue="output",size=12)
g.map(plt.scatter, "Dim_1", "Dim_2").add_legend()
plt.title("With perplexity = 50")
plt.show()
```



Conclusion- Clearly by using t-SNE we can able to visualize handwritten image from 0 to 9 of dimensions 784 reduced to 2 dimensons .

Observations: To get this output we as machine learning developer or aspiring data scientist need to have lot of patience as the code(tsne_transformed_data=model_TSNE.fit_transform(dataa)) i have executed for 2 days.

In [ ]: