

```
In [1]: #Isolation forest help us to split the data and create random forest more than 100  
#in this 100 decision tree you find different outliers  
#in this 100 tree one point is common that called isolation point.
```

## 1.Import libraries

```
In [2]: import pandas as pd  
import numpy as np  
from sklearn import datasets  
from sklearn.ensemble import IsolationForest
```

## 2.Import Datasets

```
In [3]: df=datasets.load_iris()
data=pd.DataFrame(df.data,columns=df.feature_names)
target=pd.DataFrame(df.target,columns=["species"])
target
data=pd.concat([data,target],axis=1)
data
```

Out[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

3.Data Preparation

```
In [4]: data_encoder=pd.get_dummies(data)
data_encoder
```

Out[4]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

4.Training the model

```
In [5]: #Training the model
model=IsolationForest(random_state=12,contamination=0.01)
#if we increase contamination it will give much more outlayers
#its depend on company and programmer to find out which is outlayers.
#we take example of sale of sanitizer in2020 it must be higer than any year so if we consider as output or nor decide by company only.
```

```
In [6]: model.fit(data)
```

Out[6]: IsolationForest(contamination=0.01, random\_state=12)

```
In [7]: y_pred=model.predict(data)
y_pred
#-1 indicated outlayers 1 indicate inputlayers
```

```
Out[7]: 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])
```

```
In [8]: data.head()
```

Out[8]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

5.Data Prepration

```
In [9]: #Add new data at 150 position
data.loc[150]=[40,50,20,10,1]
```

6.Model Testing

```
In [10]: #Training the model
clf=IsolationForest(contamination=0.1,random_state=12)
clf.fit(data)
```

```
Out[10]: IsolationForest(contamination=0.1, random_state=12)
```

```
In [11]: cl=clf.predict(data)
```

```
In [12]: np.where(cl==-1)
```

```
Out[12]: (array([ 13,  14,  15,  18,  22,  32,  41,  60, 106, 109, 117, 118, 119,  
                131, 150], dtype=int64),)
```