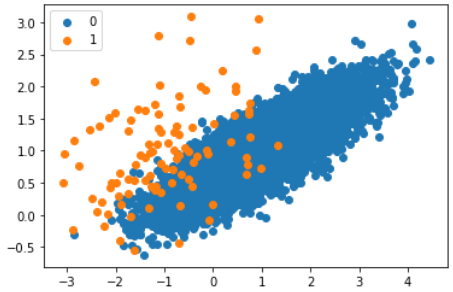
# Using smote to balance datasets

```
from sklearn.datasets import make_classification
from matplotlib import pyplot as plt
from numpy import where
from collections import Counter


# Define the dataset
X, y = make_classification(n_samples = 10000, n_features = 2 , n_redundant=0,  n_clusters_per_class=1, weights=[0.99], flip_y = 0 , random_state=1)



#Summarize class distribution
counter = Counter(y)
print(counter)
```

```
Counter({0: 9900, 1: 100})
```

```
#scatter plot of the samples
for label,_ in counter.items():
  row_ix = where(y==label)
  plt.scatter(X[row_ix,0], X[row_ix,1], label = str(label))
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f256b337a90>
```




```
#Oversample and plot imbalanced dataset with SMOTE
#Transform the dataset using SMOTE
from imblearn.over_sampling import SMOTE
oversample = SMOTE()
X1,y1 = oversample.fit_resample(X,y)
```

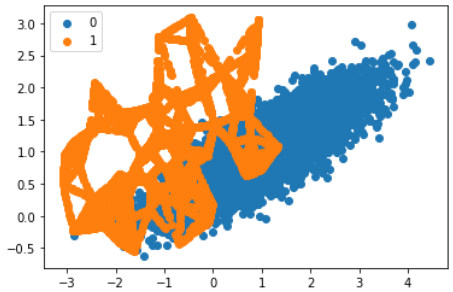
```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in version
  warnings.warn(msg, category=FutureWarning)
```

```
counter1 = Counter(y1)
print(counter1)
```

```
Counter({0: 9900, 1: 9900})
```

```
#scatter plot of the samples
for label,_ in counter1.items():
  row_ix1 = where(y1==label)
  plt.scatter(X1[row_ix1,0], X1[row_ix1,1], label = str(label))
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f25683a7ed0>
```




```
#Oversample with SMOTE and random
from imblearn.under_sampling import RandomUnderSampler
from imblearn.pipeline import Pipeline
over = SMOTE(sampling_strategy=0.1)
under = RandomUnderSampler(sampling_strategy=0.5)
# a=0.5, majority samples after resampling m= (minority/0.5)
step = [('o',over),('u',under)]
pipeline = Pipeline(steps=step)



X2,y2 = pipeline.fit_resample(X,y)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in version
  warnings.warn(msg, category=FutureWarning)
```

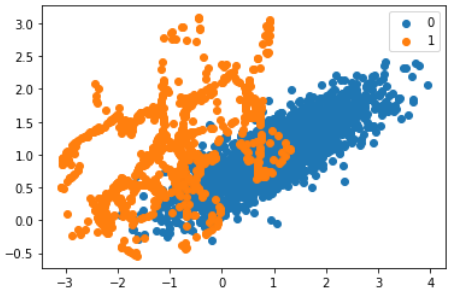
```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in version
  warnings.warn(msg, category=FutureWarning)
```

```
counter2 = Counter(y2)
print(counter2)
```

```
Counter({0: 1980, 1: 990})
```

```
#scatter plot of the samples
for label,_ in counter2.items():
  row_ix2 = where(y2==label)
  plt.scatter(X2[row_ix2,0], X2[row_ix2,1], label = str(label))
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f2565f00d90>
```




```
# decision tree classifier on imbalanced dataset
from numpy import mean
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.tree import DecisionTreeClassifier

#define the model
model = DecisionTreeClassifier()
cv = RepeatedStratifiedKFold(n_splits = 10, n_repeats=3, random_state=1)
scores=cross_val_score(model,X,y,scoring = 'roc_auc', cv=cv)
print('Mean ROC AUC: %.3f' %mean(scores))
```

```
Mean ROC AUC: 0.769
```

```
# decision tree classifier on oversampled dataset
#define pipeline
step = [('over', SMOTE()), ('model', DecisionTreeClassifier())]
pipeline = Pipeline(steps = step)
cv = RepeatedStratifiedKFold(n_splits = 10, n_repeats=3, random_state=1)
scores=cross_val_score(model,X1,y1,scoring = 'roc_auc', cv=cv)
print('Mean ROC AUC: %.3f' %mean(scores))
```

```
Mean ROC AUC: 0.935
```

```
# decision tree classifier on oversampled and undersampled dataset
#define pipeline
step = [('over', over), ('under', under), ('model', DecisionTreeClassifier())]
pipeline = Pipeline(steps = step)
cv = RepeatedStratifiedKFold(n_splits = 10, n_repeats=3, random_state=1)
scores=cross_val_score(model,X2,y2,scoring = 'roc_auc', cv=cv)
print('Mean ROC AUC: %.3f' %mean(scores))
```

```
Mean ROC AUC: 0.914
```

## With new values

```
#Oversample and plot imbalanced dataset with SMOTE
#Transform the dataset using SMOTE
from imblearn.over_sampling import SMOTE
oversample = SMOTE()
X3,y3 = oversample.fit_resample(X,y)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in version
  warnings.warn(msg, category=FutureWarning)
```

```
counter3 = Counter(y3)
print(counter3)
```

```
Counter({0: 9900, 1: 9900})
```

```
#scatter plot of the samples
for label,_ in counter3.items():
  row_ix3 = where(y3==label)
  plt.scatter(X3[row_ix3,0], X3[row_ix3,1], label = str(label))
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f2565e3e590>
```


## Taking sampling startegy as 0.2 for over and as 0.7 for under

```
#Oversample with SMOTE and random
from imblearn.under_sampling import RandomUnderSampler
from imblearn.pipeline import Pipeline
over = SMOTE(sampling_strategy=0.2)
under = RandomUnderSampler(sampling_strategy=0.7)
# a=0.5, majority samples after resampling m= (minority/0.5)
step = [('o',over),('u',under)]
pipeline = Pipeline(steps=step)


X4,y4 = pipeline.fit_resample(X,y)
```

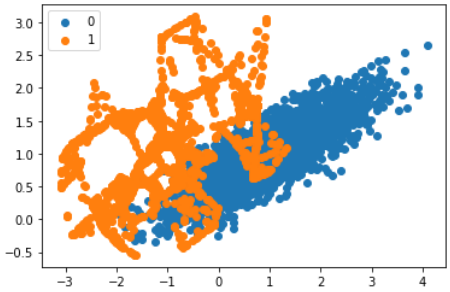
```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in version
  warnings.warn(msg, category=FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function safe_indexing is deprecated; safe_indexing is deprecated in version
  warnings.warn(msg, category=FutureWarning)
```

```
counter4 = Counter(y4)
print(counter4)
```

```
Counter({0: 2828, 1: 1980})
```

```
#scatter plot of the samples
for label,_ in counter4.items():
  row_ix4 = where(y4==label)
  plt.scatter(X4[row_ix4,0], X4[row_ix4,1], label = str(label))
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f2565e7d7d0>
```




```
# decision tree classifier on oversampled and undersampled dataset
#define pipeline
step = [('over', over), ('under', under), ('model', DecisionTreeClassifier())]
pipeline = Pipeline(steps = step)
cv = RepeatedStratifiedKFold(n_splits = 10, n_repeats=3, random_state=1)
scores=cross_val_score(model,X4,y4,scoring = 'roc_auc', cv=cv)
print('Mean ROC AUC: %.3f' %mean(scores))
```

```
Mean ROC AUC: 0.926
```

```
# decision tree classifier on oversampled dataset
#define pipeline
step = [('over', SMOTE()), ('model', DecisionTreeClassifier())]
pipeline = Pipeline(steps = step)
cv = RepeatedStratifiedKFold(n_splits = 10, n_repeats=3, random_state=1)
scores=cross_val_score(model,X4,y4,scoring = 'roc_auc', cv=cv)
print('Mean ROC AUC: %.3f' %mean(scores))
```

```
Mean ROC AUC: 0.926
```

✓ 0s completed at 18:07