# DATA PREPROCESSING

## IMPORT THE NECESSARY LIBRARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
```

## IMPORT THE DATASET

In [13]:
```
df=pd.read_csv('Churn_Modelling.csv')
```

In [14]:
```
df.head()
```

Out[14]:

| RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 15634602 | Hargrave | 619 | France | Female | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| 1 | 15647311 | Hill | 608 | Spain | Female | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| 2 | 15619304 | Onio | 502 | France | Female | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |
| 3 | 15701354 | Boni | 699 | France | Female | 39 | 1 | 0.00 | 2 | 0 | 0 | 93826.63 | 0 |
| 4 | 15737888 | Mitchell | 850 | Spain | Female | 43 | 2 | 125510.82 | 1 | 1 | 1 | 79084.10 | 0 |

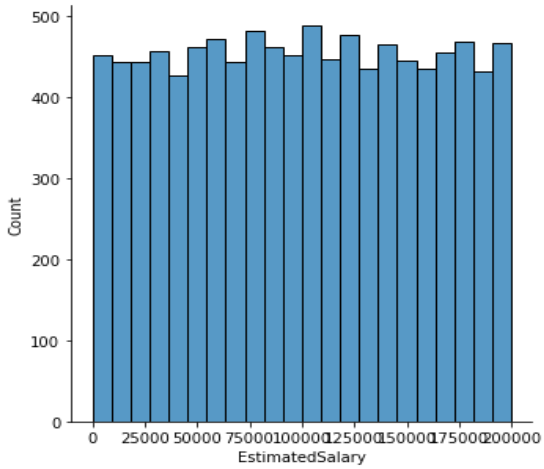
```
sns.displot(df['EstimatedSalary'])
```

Out[15]:

**VISUALIZATION:**

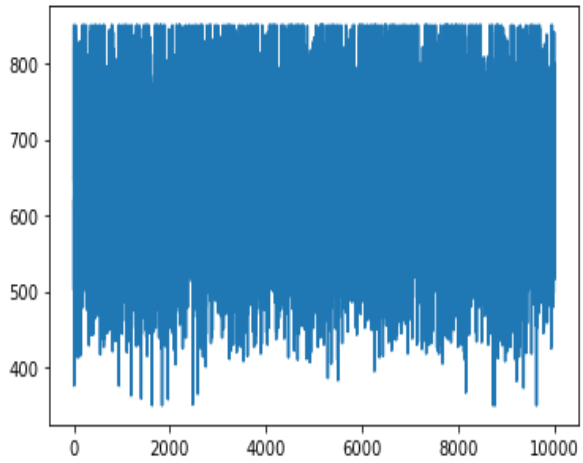
sns.displot(df['EstimatedSalary'])

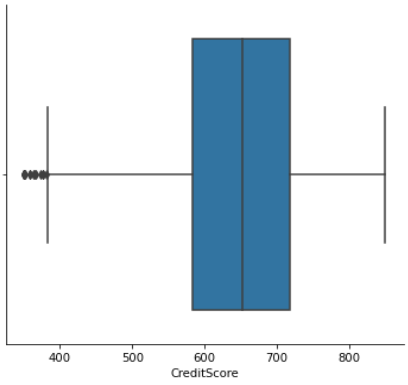
Out[15]:




**univariate analysis**

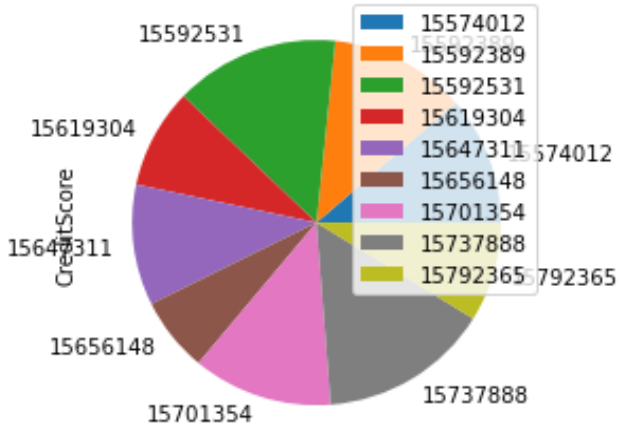
df.CreditScore.plot()



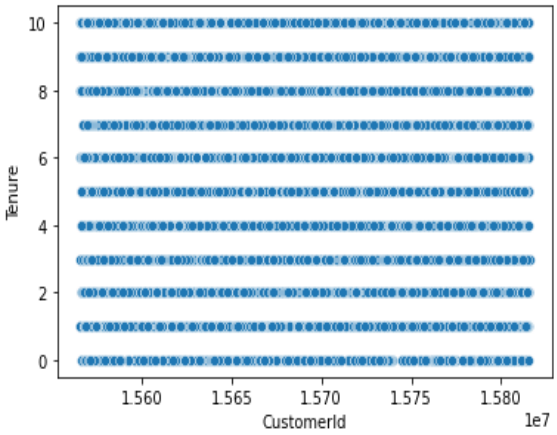



sns.catplot(x='CreditScore', kind='box',data=df)

df[1:10].groupby(['CustomerId']).sum().plot(kind='pie',y='CreditScore')

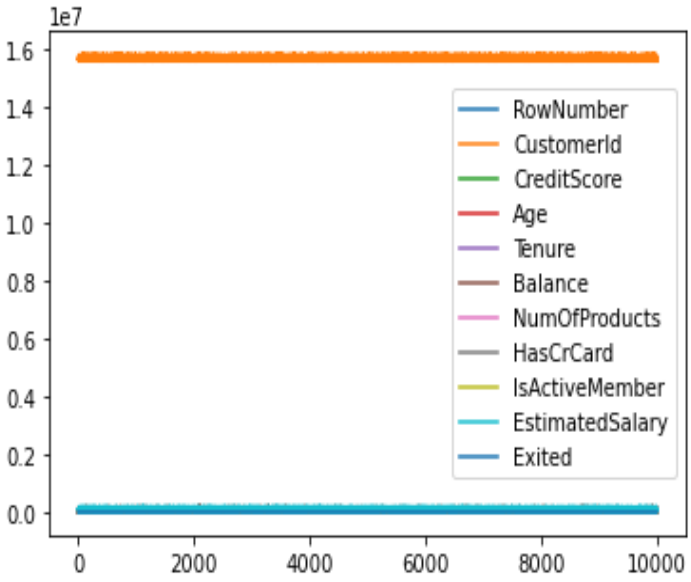



sns.scatterplot(df.CustomerId,df.Tenure)
plt.show()
C:\Users\DELL i5-3593\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning
: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional
argument will be `data`, and passing other arguments without an explicit keyword will result in a
n error or misinterpretation.
  warnings.warn(

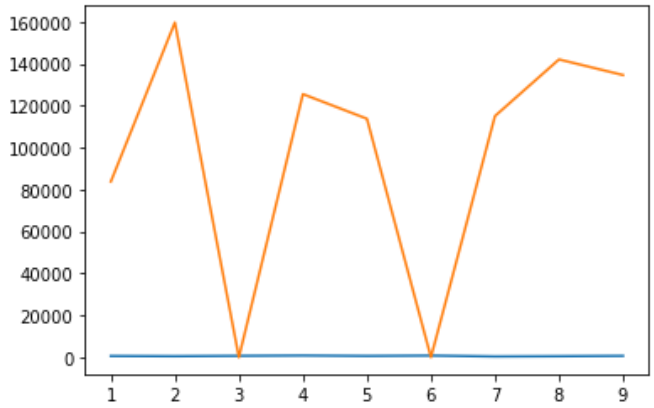



**Multivariate Analysis**

df.plot()




*#bivariate analysis*
df.CreditScore[1:10].plot()
df.Balance[1:10].plot()




**DESCRIPTIVE ANALYSIS**

df.describe()

| | RowNumber | CustomerId | CreditScore | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 10000.000000 | 1.000000e+04 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 |
| **mean** | 5000.500000 | 1.569094e+07 | 650.528800 | 38.921800 | 5.012800 | 76485.889288 | 1.530200 | 0.70550 | 0.515100 | 100090.239881 | 0.203700 |
| **std** | 2886.895680 | 7.193619e+04 | 96.653299 | 10.487806 | 2.892174 | 62397.405202 | 0.581654 | 0.45584 | 0.499797 | 57510.492818 | 0.402769 |
| **min** | 1.000000 | 1.556570e+07 | 350.000000 | 18.000000 | 0.000000 | 0.000000 | 1.000000 | 0.00000 | 0.000000 | 11.580000 | 0.000000 |

| | RowNumber | CustomerId | CreditScore | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **25%** | 2500.75000 | 1.562853e+07 | 584.000000 | 32.000000 | 3.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 51002.110000 | 0.000000 |
| **50%** | 5000.50000 | 1.569074e+07 | 652.000000 | 37.000000 | 5.000000 | 97198.540000 | 1.000000 | 1.000000 | 1.000000 | 100193.915000 | 0.000000 |
| **75%** | 7500.25000 | 1.575323e+07 | 718.000000 | 44.000000 | 7.000000 | 127644.240000 | 2.000000 | 1.000000 | 1.000000 | 149388.247500 | 0.000000 |
| **max** | 10000.00000 | 1.581569e+07 | 850.000000 | 92.000000 | 10.000000 | 250898.090000 | 4.000000 | 1.000000 | 1.000000 | 199992.480000 | 1.000000 |

df.isnull().any()

```
RowNumber          False
CustomerId         False
Surname            False
CreditScore        False
Geography          False
Gender             False
Age                False
Tenure             False
Balance            False
NumOfProducts      False
HasCrCard          False
IsActiveMember     False
EstimatedSalary    False
Exited             False
dtype: bool
```

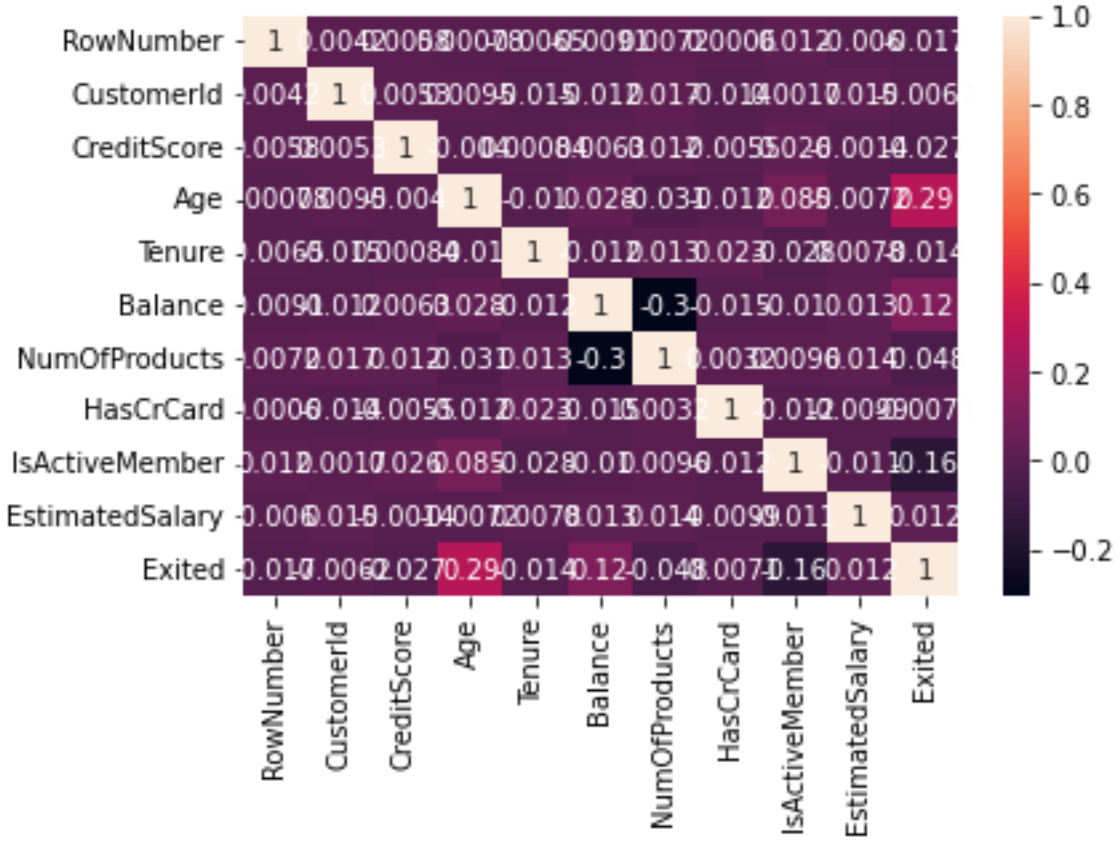
df.isnull().sum()

```
RowNumber          0
```

CustomerId 0
Surname 0
CreditScore 0
Geography 0
Gender 0
Age 0
Tenure 0
Balance 0
NumOfProducts 0
HasCrCard 0
IsActiveMember 0
EstimatedSalary 0
Exited 0
dtype: int64

**HANDLING VALUES**

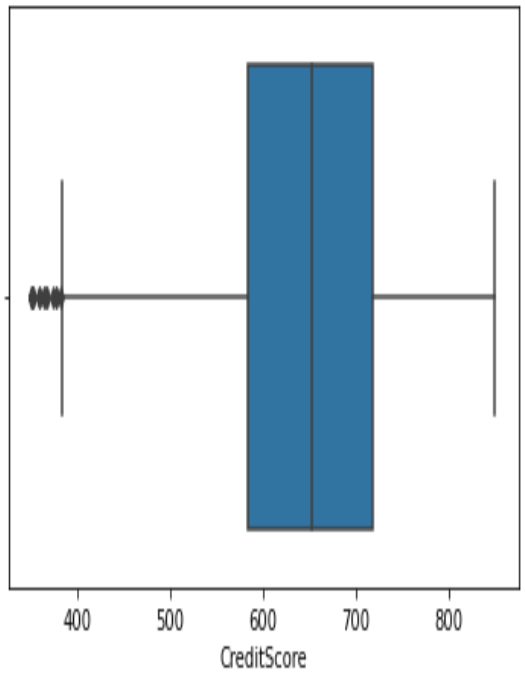
sns.heatmap(df.corr(),annot=**True**)



**OUTLIERS**

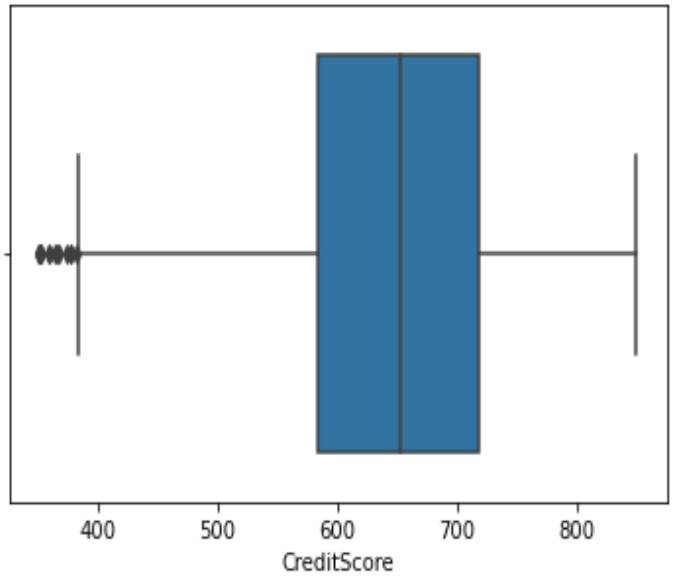
```
sns.boxplot(df.CreditScore)
```

```
C:\Users\DELL i5-3593\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning
: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(
```




```
Q1=df.Cre

ditScore.quantile(0.25)
Q3=df.CreditScore.quantile(0.75)

IQR=Q3-Q1

upper_limit=Q3+1.5*IQR
lower_limit=Q1-1.5*IQR

df['CreditScore']=np.where(df['CreditScore']>upper_limit,30,df['CreditScore'])

sns.boxplot(df.CreditScore)
```

C:\Users\DELL i5-3593\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning
: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit keyword will result in an
error or misinterpretation.
  warnings.warn(




Categorical Columns Encoding

```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()

df.Gender=le.fit_transform(df.Gender)

df.head(5)
```

| RowNumber | CustomerId | Surname | CreditScore | Geography | Gender | Age | Tenure | Balance | NumOfProducts | HasCrCard | IsActiveMember | EstimatedSalary | Exited |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 15634602 | Hargrave | 619 | France | 0 | 42 | 2 | 0.00 | 1 | 1 | 1 | 101348.88 | 1 |
| 1 | 15647311 | Hill | 608 | Spain | 0 | 41 | 1 | 83807.86 | 1 | 0 | 1 | 112542.58 | 0 |
| 2 | 15619304 | Onio | 502 | France | 0 | 42 | 8 | 159660.80 | 3 | 1 | 0 | 113931.57 | 1 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 | 15701354 | Boni | 699 | France | 0 | 39 | 1 | 0.00 | 2 | 0 |
| | 0 | 93826.63 | 0 | | | | | | | | |
| 4 | 5 | 15737888 | Mitchell | | 850 | Spain | 0 | 43 | 2 | 125510.82 | |
| | 1 | 1 | 1 | 79084.10 | 0 | | | | | | |

```
df_main=pd.get_dummies(df,columns=['Geography'])
```

```
df_main.head()
```

**SEPARATING INDEPENDENT AND DEPENDENT VARIABLES**

```
X=df_main.drop(columns=['EstimatedSalary'],axis=1)
X.head()
X_scaled=pd.DataFrame(scale(X),columns=X.columns)
X_scaled.head()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/464208997.py in
      1 X=df_main.drop(columns=['EstimatedSalary'],axis=1)
      2 X.head()
----> 3 X_scaled=pd.DataFrame(scale(X),columns=X.columns)
      4 X_scaled.head()

NameError: name 'scale' is not defined
```

```
y=Df_main.EstimatedSalary
y
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/912569982.py in
----> 1 y=Df_main.EstimatedSalary
      2 y

NameError: name 'Df_main' is not defined
```

**SCALING**

```
from sklearn.preprocessing import scale
X_scaled=pd.DataFrame(scale(X),columns=X.columns)
X_scaled.head()
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/883186635.py in
----> 1 X_scaled=pd.DataFrame(scale(X),columns=X.columns)
      2 X_scaled.head()

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
     64
     65             # extra_args > 0

~\anaconda3\lib\site-packages\sklearn\preprocessing\_data.py in scale(X, axis, with_mean,
with_std, copy)
    159
    160     """  # noqa
--> 161     X = check_array(X, accept_sparse='csc', copy=copy, ensure_2d=False,
    162                     estimator='the scale function', dtype=FLOAT_DTYPES,
    163                     force_all_finite='allow-nan')

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in inner_f(*args, **kwargs)
     61             extra_args = len(args) - len(all_args)
     62             if extra_args <= 0:
---> 63                 return f(*args, **kwargs)
64
     65             # extra_args > 0

~\anaconda3\lib\site-packages\sklearn\utils\validation.py in check_array(array, accept_spar
se, accept_large_sparse, dtype, order, copy, force_all_finite, ensure_2d, allow_nd, ensure_
min_samples, ensure_min_features, estimator)
    671                     array = array.astype(dtype, casting="unsafe", copy=False)
    672                 else:
--> 673                     array = np.asarray(array, order=order, dtype=dtype)
    674             except ComplexWarning as complex_warning:
    675                 raise ValueError("Complex data not supported\n"

~\anaconda3\lib\site-packages\numpy\core\_asarray.py in asarray(a, dtype, order, like)
    100         return _asarray_with_like(a, dtype=dtype, order=order, like=like)
    101
--> 102     return array(a, dtype, copy=False, order=order)
    103
    104

~\anaconda3\lib\site-packages\pandas\core\generic.py in __array__(self, dtype)
   1991
```

```
  1992     def __array__(self, dtype: NpDtype | None = None) -> np.ndarray:
-> 1993         return np.asarray(self._values, dtype=dtype)
  1994
  1995     def __array_wrap__(

~\anaconda3\lib\site-packages\numpy\core\_asarray.py in asarray(a, dtype, order, like)
   100         return _asarray_with_like(a, dtype=dtype, order=order, like=like)
   101
--> 102     return array(a, dtype, copy=False, order=order)
   103
104

ValueError: could not convert string to float: 'Hargrave'
```

**TRAIN AND TEST DATA**

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test =train_test_split(X_scaled,y, test_size=0.3,random_state=0)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/3510837689.py in
      1 from sklearn.model_selection import train_test_split
----> 2 X_train,X_test,y_train,y_test =train_test_split(X_scaled,y, test_size=0.3,random_state=0)

NameError: name 'X_scaled' is not defined
```

```
X_train.shape
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/4225672638.py in
----> 1 X_train.shape

NameError: name 'X_train' is not defined
```

```
X_test.shape
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/3624294392.py in
----> 1 X_test.shape

NameError: name 'X_test' is not defined
```

```
y_train.shape
```

```
---------------------------------------------------------------------------
```

```
NameError                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/3798806461.py in
----> 1 y_train.shape

NameError: name 'y_train' is not defined
```

```
y.train.shape
```

```
NameError                 Traceback (most recent call last)
C:\Users\DELLI5~1\AppData\Local\Temp/ipykernel_13384/830994784.py in
----> 1 y.train.shape

NameError: name 'y' is not defined
```