Packages:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

sns.set\_style('darkgrid')

import warnings

warnings.simplefilter(action='ignore', category=FutureWarning)

-----------------------------------------------------------------------

SNS

sns.set\_style('darkgrid')

sns.scatterplot(data=college, x='Outstate', y='F.Undergrad', hue='Private')

sns.heatmap(titanic\_train.isnull(), yticklabels=False,cbar=False,cmap='viridis')

sns.boxplot(x='Pclass', y='Age', data=titanic\_train)

sns.displot(titanic\_train['Age'].dropna(),kde=False, bins=30)

sns.pairplot(customers)

sns.countplot(x='purpose',hue='not.fully.paid',data=loans,palette='Set1')

sns.jointplot(x='fico', y='int.rate', data=loans,color='purple')

sns.lmplot(data=loans, x ='fico', y='int.rate', hue='credit.policy', col='not.fully.paid',palette='Set1')

setosa = iris[iris['species']=='setosa']

sns.kdeplot(data=setosa, x=setosa['sepal\_width'], y=setosa['sepal\_length'], cmap='plasma', shade=True, thresh=0.1)

------------------------------------------------------------------------

Pandas

loans[loans['not.fully.paid']== 1]['fico'].hist(alpha=0.5, color='blue', label='not.fully.paid =1', bins = 30)

loans[loans['not.fully.paid']== 0]['fico'].hist(alpha=0.5, color='red', label='not.fully.paid =0', bins = 30)

------------------------------------------------------------------------

Sklearn:

Pre-processing

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.model\_selection import GridSearchCV

from sklearn.decomposition import PCA

-----------------------------------------------------------------------

Regression:

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

print('MAE:', metrics.mean\_absolute\_error(y\_test, predictions))

print('MSE:', metrics.mean\_squared\_error(y\_test, predictions))

print('RMSE:', np.sqrt(metrics.mean\_squared\_error(y\_test, predictions)))

-----------------------------------------------------------------------

Classification:

from sklearn.metrics import classification\_report,confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.cluster import KMeans

from sklearn.naive\_bayes import GaussianNB

-----------------------------------------------------------------------

Datasets:

from sklearn.datasets import make\_blobs

data = make\_blobs(n\_samples = 200, n\_features = 2, centers = 4, cluster\_std=1.8, random\_state = 101)

data[0], data[0][:,0], data[0][:,1], data[1]

from sklearn.datasets import load\_breast\_cancer

from sklearn.datasets import load\_iris