```
O Rardon forest exemble learning
                               it val stall
  import pardas as pol
  from sklear, midel-selection import tean_test_split
  from ekleren ersemble import Rondon Frosett Classifies
  from shleads muchics import accuracy-score, classification report
   from google, colab impact files
   uplanded = files.upland()
   for filenane in uplcaded beys ():
        of = pd, lead_car (filename)
        print (f"Data loaded from: (filerane?")
        display (df. head ())
                                                    () le lodg 11.
                                                  (oner) bus of
  x = df. iloc [:,:-1]
                                                     (Twate the
  8 = df. iloz (:, -D)
  X train, X test, y train, y test = train_test_split (X, y) test_Rige = 0. 2, Rorder to
  rf_model = RandomForest classifier (n_estimatore = 100, random, state = 42)
  et-medel. fit (x train, y troin)
                                      17) Justic you = triop and
  y-pred = ef_model. predict (x-test)
  accuracy = accuracy _ more (y text, y - pred)
  print (f"newway of Rondom Forest Model: dawney *100: 247%")
  print ("Classification report: ")
                                  O water to higher or
  print (chamification_report (y-test, y-pred))
                precision
                                   F1-store
                          recall
                                                support
                0.79
                          0.78
                                   6.78
                       0.62
                 0.61
                                  6-61
                                              55
```

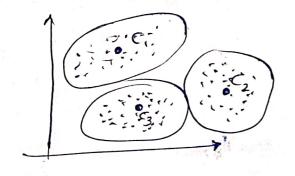
ob- 5 Holland is ofman

```
(2) Boosting algorithm
                       olati
 import nampy as up
 from shlear tree import Decilia Tree Classifier
 bean extern datasets import make classification a language
 from splean metrics import occuracy - scale ( , r ) 17
                             occupacy is adabast score (x x)
  class AdoBost:
       def - nit_ (self, n_estimators = 50):
             self. nestimators = n-estimators
                                     , property July & tintat
            seff.alphas = []
            self, models = []
        def fit (self, x, y):
           n-samples, n-features = x shape
           w = np.ones (n_ samples) /n_samples 9 20 games
           for estimator in range (self-in-estimators):
                model = Decision Tree Classifier (man-depth = 1)
                model. fit (X, y, sample_weighthe w) again
                y-pred = model. predict (x)
               ess = np, sun (w * (y-pred != y)) / np. sum (w)
                alpha = 0.5 * np. log ((1-ere)/ere) if ere <1 die 0
               self. alphas. append (alpha)
          sent will, models append (model) sies mas Folad. 19 store
               w= w*np. exp (-ulpha, *y, y-pred)
               w= W/np.sum (w)
         dif priduit (sell , x): time & more and a some
              final gred = np. yeros (x, shape [0])
              for model, alpha in gip (self models, self, alphas):
                  final-pred += alpha * model. predict (X)
             return np. sign (final - pred)
         def rose (self, x, y) : " dels dels
              between accuracy_ work (y, self, predict (x))
```

711...

```
X, y = nake_classification (n-samples = 500, n-features = 20, n. classes : 1
                                          the we from
     y = 2 y -1 margarent sections traged and rent
    adaboost = Ada Boost (n_extinators = 50)
    adaboost. fit (x, y) since trocesso. Logos esisten. ment.
   occulary = adapost. ecole (X, V)
    print (f" model accuracy: facuracy: , 4 f 3")
                       selfe residentices = n-situations
 output: Model Accuracy: 0.9580
                                  () = elebrar flee
                                : (R x for) to !!
3 K means squal = heartref n, selphon n
 import may as up somples on so yearn tragmi
 import pardas as (red much on the sound of solution sol
 import notplotlib, pyplot as plt
 from splean imports dotagets elans ( ) It Islan
 import seaborn at ens (>) today, Oslaw hay
 Grove ableven chutes import Kneam ((+ ! huge) * w) me ign 300
 iris = dotasets, load isis ()
 print ("Data set looded successfully") 4" 20 mgs
 Data = pd. Data Frame (isis. data, columns = ieis, feature_names)
 x = Data ila [:, 0:3] inalises mali
                                 mus for for - w
 us = []
benears = Knears (n-clusters=3, init = "K-means++1, mox-itex=100,
               n-nit=10, 200don- state =0)
 y- kneary = knears, bit-preduit (x) in right leter and
 knews, dutter_certers_ but a told =+ but build
 elt. reatter (x [y knear == 0, 0], x [y knear == 0, 1] s= 100, (= 12d)
              label: 'Inn. Setora') (, x 1/22)
plt. scottere (k_mans = striker - sedery [:, o), knuens, churter - certers - [:, ],
             S=100, c= block', label: (extractly)
```

plt. leneral()



(P) PCA

import pardas as pd import numpy as up from skleven, decomposition import PCA from spleass, peopleocessing import standar Scales import natplothis, pyplot as plt from gorgle color import biles

uploaded > feles. upload() for filename is uploaded keys (): df = pd. seed\_cer (filerane) print (+" Uploaded: ffilerane?") display (df. head ())

numeric - df = df. select\_dtypes (include = [np. number]) print ("Numerical features found: ", list (numeric. of . column))

relected = features = numeric\_df. columns

X = numeric\_df [selected\_beatures]. deopna() X-hooled = Standard Scales (). fit\_transferen (x)

pca = PCA (n-comparents = 2)

principal\_comparety = pea. fit\_konsform (x\_ realed)

pea\_ of = pd DataFrame (data - principal - components, columns = ['901', '102'])

Jel 18 125

plt, figure (figeige = (8.6) plt. grid (True)

pt. show 1)

print (" Explained Valiance Rotio: ", pea explained \_ radiance \_ ratio\_)

print (f'indel accuracy: {auriouy! . 4+ 4")

Variance Ration: [0. 52163044 0-48631263) 0/0:-Model Accuracy: 6.9580