```
Q Search...
                                                                                                 Products
                                                                                                            Customers
                                                                                                                        Use cases
281
     # In[109]:
282
283
                                                                                                         df = pd.read csv("DataWithoutHeader162.csv")
                                                                                                         df.columns = ['Temperature', 'Humidity', 'Windspeed', 'Traffic', 'PM 2.5']
                                                                           Home
     train, test = train_test_split(my_df,
                                                                                                         #print(df.head())
285
                                      test size = 0.3,
                                                                           PUBLIC
286
                                      random_state = 0
                                                                                                         forecast col = 'PM 2.5'
                                                                                                         df['label'] = df[forecast_col].shift(1)
287
                                                                           Stack Overflow
                                      stratify = my_df['Dataset'])
                                                                                                         df.fillna(value=-99999, inplace=True)
     train_X = train[train.columns[:len(train.columns)-1]]
                                                                              Tags
     test X = test[test.columns[:len(test.columns)-1]]
                                                                                                         X = np.array(df.drop(['label','PM 2.5'] , 1))
                                                                                                         X = preprocessing.scale(X)
     train Y = train['Dataset']
                                                                              Users
                                                                                                         df.dropna(inplace = True)
     test Y = test['Dataset']
                                                                              Jobs
292
                                                                                                         y = np.array(df['label'])
293
                                                                                                         df.dropna(inplace - True)
                                                                            TEAMS
                                                                                       What's this?
     # In[113]:
                                                                                                         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.05)
295
                                                                            Free 30 Day Trial
296
     types=['rbf','linear', 'sigmoid']
                                                                                                         #kernel definition
                                                                                                         def radial basis(gamma=10):
298
     for i in types:
                                                                                                             return lambda x, y: np.exp(-gamma*la.norm(np.subtract(x, y)))
299
          model = svm.SVC(kernel=i, random_state=0)
300
          model.fit(train X,train Y)
                                                                                                          #SupportVectorMachine with radial basis Kernel
                                                                                                         clf_SVM_radial_basis = SVC(kernel = radial_basis())
                                                                                                         clf_SVM_radial_basis.fit(X_train,y train)
                                                                                                         confidence3 = cti_Svii_radial_basis.score(x_test,y_test)
                                                                                                         print("Confidence of SVM with radial basis Kernel = ",(confidence3*100),"%")
```