```
#################################################################

# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the input directory

from pandas import DataFrame
from sklearn.model_selection import train_test_split


import os
#print(os.listdir("../input"))

# Any results you write to the current directory are saved as output.

train_d = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
#train.describe(include="all")
count = 1

combine = [train_d,test]
#smoke = train_d['smoker_status']
for train in combine:
    train['smoker_status_1'] = train['smoker_status'].replace(['__', '11', ',,', '.',
       '_', '□□□', ',', '>??', '?', '>', 'quit?', 'non>', 'N?A', '..',
       '??', '?', 'N?a', '>'], 'ND')


    sex_age = train["sex and age"].str.split(',').tolist()
    #print(np.array(sex_age))
    age = []
    sex = []
    #df2 = pd.DataFrame(np.array(sex_age), columns = ["sex","age"])
    #print(sex_age[0:7])
    #sex_age = sex_age [~np.isnan(sex_age)]
    #sex_age.fillna([0,0])
    for ages in sex_age:
        try:
            if pd.isnull(ages):
                #print('NAN')
                age.append(0)
                sex.append(0)
                continue
            #print(ages[1])
            if float(ages[1]):
                #print("HIIII")
                age.append(ages[1])
                sex.append(ages[0])
        except ValueError:
            #print(f'Error {ages}')
            try:
                if float(ages[1]):
                    #print("HIIII")
                    age.append(ages[1])
                    sex.append(ages[0])
            except ValueError:
                try:
                    if float(ages[0]):
                        age.append(ages[0])
                        sex.append(ages[1])
                except:
```

```
                    age.append(0)
                    sex.append(0)
    #print(age)
    #print(sex)
    #DataFrame.from_records(sex_age)
    #df = pd.DataFrame({'age':age,'sex':sex})
    #df2 = pd.DataFrame(np.array(age), columns = ["age"])

    age_1 = [float(i) for i in age]
    #print(age_1)
    train['age'] = age_1
    train['sex'] = sex

    js_la = train["job_status and living_area"].str.split('?',n=1,expand = True)
    train['temp_js'] = js_la[0].str.lower()
    #print(train['temp_js'].head(5))
    train['temp_la'] = js_la[1].str.lower()
    #print(train['temp_la'].head(5))
    #train['temp_js'].head(2572)
    #train['temp_la'].unique()


    train['temp_js'] = train['temp_js'].replace(['remote','remotee','r'], 'r')
    train['temp_js'] = train['temp_js'].replace(['city','c'], 'c')
    train['temp_js'] = train['temp_js'].replace(['government','govt.'], 'g')
    train['temp_js'] = train['temp_js'].replace(['private_sector', 'privattte',
    'private', 'private sector'], 'p')
    train['temp_js'] = train['temp_js'].replace(['biz', 'business_owner', 'business
    owner'], 'b')
    train['temp_js'] = train['temp_js'].replace(['unemployed', '', 'n.a'], 'u')
    train['temp_js'] = train['temp_js'].replace(['leave', 'parental_leave', 'parental
    leave'], 'l')

    #print(train['temp_js'].head(5))

    train['temp_la'] = train['temp_la'].replace(['remote','remotee','r'], 'r')
    train['temp_la'] = train['temp_la'].replace(['city','c'], 'c')
    train['temp_la'] = train['temp_la'].replace(['government','govt.'], 'g')
    train['temp_la'] = train['temp_la'].replace(['private_sector', 'privattte',
    'private', 'private sector'], 'p')
    train['temp_la'] = train['temp_la'].replace(['biz', 'business_owner', 'business
    owner'], 'b')
    train['temp_la'] = train['temp_la'].replace(['unemployed', '', 'n.a'], 'u')
    train['temp_la'] = train['temp_la'].replace(['leave', 'parental_leave', 'parental
    leave'], 'l')


    train['temp_la'] = train['temp_la'].replace(['null'], '')
    train['temp_js'] = train['temp_js'].replace(['null'], '')
    #print(train['temp_la'].head(5))

    js = ['' for _ in range(len(train["job_status and living_area"]))]
    la = ['' for _ in range(len(train["job_status and living_area"]))]

    loc = ['r','c']
    job = ['g','p','u','b','l']

    #print(train['temp_la'][0])
    for i in range(len(train["job_status and living_area"])):
        if train['temp_la'][i] in loc:
            la[i] = train['temp_la'][i]
            #js[i] = train['temp_js'][i]
        if train['temp_la'][i] in job:
        #else:
            #js[i] = train['temp_la'][i]
            la[i] = train['temp_js'][i]
        if train['temp_js'][i] in loc:
            js[i] = train['temp_la'][i]
            #la[i] = train['temp_js'][i]
```

```
        if train['temp_js'][i] in job:
            #la[i] = train['temp_la'][i]
            js[i] = train['temp_js'][i]
        '''if la[i] == 'p':
            print(train['temp_la'][i])
            print(train['temp_js'][i])
            pass'''
    '''print(set(js))
    print(set(la))
    print(la.count('p'))
    print(la.count('u'))'''
    train['job'] = js
    train['location'] = la


    train['female'] = train['sex']
    train['female'] = train['female'].replace(['F',' F', 'female', 'Female' ,'f'
    ,'Other',' Other','femalle','FEMALE' ],1)
    train['female'] = train['female'].replace(['M',' M' ,'male' ,'MALE','m', 'mmale',
    'MM','Male'],0)

    train['male'] = train['sex']
    train['male'] = train['male'].replace(['M',' M' ,'male' ,'MALE','m', 'mmale',
    'MM','Male'],1)
    train['male'] = train['male'].replace(['F',' F', 'female', 'Female' ,'f' ,'Other','
    Other','femalle','FEMALE' ],0)

    train['remote'] = train['location']
    train['remote'] = train['remote'].replace('r',1)
    train['remote'] = train['remote'].replace(['c','','p'],0)

    train['city'] = train['location']
    train['city'] = train['city'].replace('c',1)
    train['city'] = train['city'].replace(['r','','p'],0)

    train['job'] = train['job'].replace('','u')

    train['govt'] = train['job']
    train['govt'] = train['govt'].replace('g',1)
    train['govt'] = train['govt'].replace(['g','p','u','b','l'],0)

    train['priv'] = train['job']
    train['priv'] = train['priv'].replace('p',1)
    train['priv'] = train['priv'].replace(['g','u','b','l'],0)

    train['unemp'] = train['job']
    train['unemp'] = train['unemp'].replace('u',1)
    train['unemp'] = train['unemp'].replace(['g','p','b','l'],0)

    train['business'] = train['job']
    train['business'] = train['business'].replace('b',1)
    train['business'] = train['business'].replace(['g','p','u','l'],0)

    train['leave'] = train['job']
    train['leave'] = train['leave'].replace('l',1)
    train['leave'] = train['leave'].replace(['g','p','u','b'],0)


    train['high_BP'] = train['high_BP'].fillna(0)
    train['high_BP'] = train['high_BP'].replace('.,',0)

    train['average_blood_sugar'] = train['average_blood_sugar'].fillna(0)
    train['BMI'] = train['BMI'].fillna(0)

    train['heart_condition_detected_2017'] =
    train['heart_condition_detected_2017'].fillna(0)
    train['heart_condition_detected_2017'] =
    train['heart_condition_detected_2017'].replace(['.','N?A','n.a','.',',','n.a','N?A','
    '],0)
```

```
193
194         train['smoker_status_1'] = train['smoker_status_1'].fillna('ND')
195         train['smoker_status_1'] =
            train['smoker_status_1'].replace(['.','.',',','n.a','N?A',''],'ND')
196
197
198         #print(train["average_blood_sugar"].unique())
199         train['BMI'] = train['BMI'].replace(['?','.',',','n.a','N?A',''],0)
200         train['BMI'] = train['BMI'].astype(float)
201         train['average_blood_sugar'] = train['average_blood_sugar'].astype(float)
202
203
204
205         #print(train['smoker_status_1'].unique())
206         # ['non-smoker' 'quit' 'active_smoker' nan 'ND']
207
208         train['active_smoker'] = train['smoker_status_1']
209         train['active_smoker'] = train['active_smoker'].replace('active_smoker',1)
210         train['active_smoker'] =
            train['active_smoker'].replace(['non-smoker','quit','active_smoker','ND'],0)
211
212         train['non_smoker'] = train['smoker_status_1']
213         train['non_smoker'] = train['non_smoker'].replace('non-smoker',1)
214         train['non_smoker'] =
            train['non_smoker'].replace(['non-smoker','quit','active_smoker','ND'],0)
215
216         train['quit'] = train['smoker_status_1']
217         train['quit'] = train['quit'].replace('quit',1)
218         train['quit'] = train['quit'].replace(['non-smoker','quit','active_smoker','ND'],0)
219
220         train['nd_smoker'] = train['smoker_status_1']
221         train['nd_smoker'] = train['nd_smoker'].replace('ND',1)
222         train['nd_smoker'] =
            train['nd_smoker'].replace(['non-smoker','quit','active_smoker','ND'],0)
223
224
225         train['married'] = train['married'].replace('.',0)
226         #print(train['age'].loc[pd.isnull(train['married'])])
227         #train['married'].loc[pd.isnull(train['married'])] = (lambda x: 1 if
            train['age'].loc[pd.isnull(train['married'])] > 30 else 0)
228         train['married'].loc[pd.isnull(train['married'])] =
            np.where(train['age'].loc[pd.isnull(train['married'])]>30,1,0)
229
230         from copy import deepcopy
231         import statistics
232
233         age_wo0 = deepcopy(age)
234         age_wo0[:] = (float(value) for value in age_wo0 if value != 0)
235
236         #print(set(sex))
237         #print(float(statistics.mean(age_wo0)))
238
239         train['age'] = train['age'].replace(0,float(statistics.mean(age_wo0)))
240
241         #print(float(statistics.mean(train['BMI'])))
242
243         train['BMI'] = train['BMI'].replace(0,float(statistics.mean(train['BMI']))+5.0)
244         train['average_blood_sugar'] =
            train['average_blood_sugar'].replace(0,float(statistics.mean(train['average_blood_sug
            ar'])))
245
246
247         train['TreatmentA'] = train['TreatmentA'].fillna(0)
248         train['TreatmentB'] = train['TreatmentB'].fillna(0)
249         train['TreatmentC'] = train['TreatmentC'].fillna(0)
250         train['TreatmentD'] = train['TreatmentD'].fillna(0)
251         train['TreatmentD'] = train['TreatmentD'].replace('0+E1860:E1868',0)
252
253
```

```
254       train['high_BP'] = train['high_BP'].astype(float)
255       train['heart_condition_detected_2017'] =
          train['heart_condition_detected_2017'].astype(float)
256       train['married'] = train['married'].astype(float)
257       train['average_blood_sugar'] = train['average_blood_sugar'].astype(float)
258       train['TreatmentA'] = train['TreatmentA'].astype(float)
259       train['TreatmentB'] = train['TreatmentB'].astype(float)
260       train['BMI'] = train['BMI'].astype(float)
261       train['TreatmentC'] = train['TreatmentC'].astype(float)
262       train['TreatmentD'] = train['TreatmentD'].astype(float)
263
264       train['age'] = train['age'].astype(float)
265       train['BMI'] = train['BMI'].astype(float)
266       train['female'] = train['female'].astype(float)
267       train['male'] = train['male'].astype(float)
268       train['remote'] = train['remote'].astype(float)
269       train['city'] = train['city'].astype(float)
270       train['govt'] = train['govt'].astype(float)
271       train['priv'] = train['priv'].astype(float)
272       train['unemp'] = train['unemp'].astype(float)
273       train['business'] = train['business'].astype(float)
274       train['leave'] = train['leave'].astype(float)
275       train['active_smoker'] = train['active_smoker'].astype(float)
276       train['non_smoker'] = train['non_smoker'].astype(float)
277       train['quit'] = train['quit'].astype(float)
278       train['nd_smoker'] = train['nd_smoker'].astype(float)
279
280
281       #for i in range(len(train['BMI'])):
282       #    BMI_SUM = (train['BMI'])
283       #print(train.columns)
284       #print(train['male'].unique())
285
286       if count == 1:
287           train.to_csv(r'D:\MS-UNSW\DataSoc_Datathon/pandas_train.csv')
288           count+=1
289
290       Drop = ['sex and age', 'job_status and
          living_area','smoker_status','temp_js','temp_la','sex','job','location','smoker_statu
          s_1']
291       train.drop(train[Drop], axis = 1, inplace = True)
292       #print(train.columns)
293
294       #train.to_csv(r'D:\MS-UNSW\DataSoc_Datathon/pandas.csv')
295       train.head()
296
297
298   train_d['stroke_in_2018'] = train_d['stroke_in_2018'].fillna(0)
299   train_d['stroke_in_2018'] =
      train_d['stroke_in_2018'].replace(['.',',','?','n.a','nuLL','N?A',''],0)
300   train_d['stroke_in_2018'] = train_d['stroke_in_2018'].astype(float)
301
302
303
304   predictors = train_d.drop(['stroke_in_2018', 'id'], axis=1)
305   target = train_d["stroke_in_2018"]
306   x_train, x_val, y_train, y_val = train_test_split(predictors, target, test_size = 0.52,
      random_state = 0)
307
308   # Perceptron
309   from sklearn.linear_model import Perceptron
310
311   perceptron = Perceptron()
312   perceptron.fit(x_train, y_train)
313   y_pred = perceptron.predict(x_val)
314   acc_perceptron = round(accuracy_score(y_pred, y_val) * 100, 2)
315   #print(acc_perceptron)
316
317   ids = test['id']
```

```
predictions = perceptron.predict(test.drop('id', axis=1))

#set the output as a dataframe and convert to csv file named submission.csv
output = pd.DataFrame({ 'id' : ids, 'stroke_in_2018': predictions })
output.to_csv('submission_per.csv', index=False)
```