Model Predicting Cevical Cancer

Loading dataset

import pandas as pd

import matplotlib.pyplot as plt

In [120]: import numpy as np

warnings.filterwarnings("ignore")

from sklearn import tree import seaborn as sns

import warnings

Age

Age

Smokes

Num of pregnancies

Smokes (packs/year)

Smokes (years)

Number

sexual

partners

of

Out[120]:

RA1911028010069 - Final Assignment

0 18 15.0 0.0 1 15 1.0 14.0 1.0 34 0.0 1.0 3 52 5.0 16.0 4.0

First

sexual

intercourse

21.0 0.0

46 5 rows × 36 columns In [121]: | df.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 858 entries, 0 to 857 Data columns (total 36 columns): Number of sexual partners First sexual intercourse

df = pd.read csv('kag risk factors cervical cancer.csv')

pregnancies

Num of

Smokes

858 non-null int64 858 non-null object 858 non-null object

858 non-null object

858 non-null object

858 non-null object 858 non-null object

Smokes

(years)

0.0

0.0

37.0

0.0

Smokes

0.0

0.0

(packs/year) Contraceptives

37.0

Hormonal

0.0

0.0 1.0 1.0

Hormonal

Contraceptives IUD ...

0.0

(years)

0.0 3.0

0.0 0.0 ... 0.0 ... 0.0 ... 0.0 ...

0.0 ...

? ?

STDs:

since first

Time

diagnosis diagnosis

STDs:

since last

Time

?

?

Hormonal Contraceptives 858 non-null object 858 non-null object Hormonal Contraceptives (years) IUD 858 non-null object 858 non-null object IUD (years) STDs 858 non-null object STDs (number) 858 non-null object STDs:condylomatosis 858 non-null object STDs:cervical condylomatosis 858 non-null object STDs:vaginal condylomatosis 858 non-null object STDs:vulvo-perineal condylomatosis 858 non-null object 858 non-null object STDs:syphilis STDs:pelvic inflammatory disease 858 non-null object STDs:genital herpes 858 non-null object STDs:molluscum contagiosum 858 non-null object STDs:AIDS 858 non-null object 858 non-null object STDs:HIV STDs:Hepatitis B 858 non-null object STDs: HPV 858 non-null object STDs: Number of diagnosis 858 non-null int64 STDs: Time since first diagnosis 858 non-null object STDs: Time since last diagnosis 858 non-null object 858 non-null int64 Dx:Cancer Dx:CIN 858 non-null int64 858 non-null int64 Dx:HPV 858 non-null int64 Hinselmann 858 non-null int64 Schiller 858 non-null int64 Citology 858 non-null int64 Biopsy 858 non-null int64 dtypes: int64(10), object(26)

Replacing '?' and dropping columns with large number of null fields In [122]: | df = df.replace('?', np.nan) df.isnull().sum() df = df.drop(columns = ['STDs: Time since first diagnosis', 'STDs: Time since last diagnosis']) Removing all null fields In [123]: mEntries = df.isnull().sum(axis=1).tolist()

memory usage: 241.4+ KB

sorted(nMapping.items())

Number of sexual partners

First sexual intercourse

Num of pregnancies

Smokes (years)

STDs (number)

STDs:syphilis

STDs:AIDS

Schiller

Citology

dtype: int64

df.info()

IUD

STDs

IUD (years)

STDs (number)

STDs:Hepatitis B

In [125]: plt.figure(figsize=(20,5))

ax = plt.gca()

plt.show()

80

60

40

df.iloc[:,:].boxplot()

STDs: Number of diagnosis

STDs: HPV

Dx:Cancer Dx:CIN

Hinselmann

Schiller

Citology

Dx:HPV

Dx

Biopsy

STDs:condylomatosis

STDs:genital herpes

STDs:cervical condylomatosis STDs:vaginal condylomatosis

STDs:molluscum contagiosum

STDs:vulvo-perineal condylomatosis

STDs:pelvic inflammatory disease

Smokes

STDs

df = df.dropna() df.isnull().sum()

Out[123]: Age

Smokes (packs/year) Hormonal Contraceptives Hormonal Contraceptives (years) 0 IUD IUD (years) 0

nMapping = dict((x, mEntries.count(x)) for x in mEntries)

0

0

0

0

0

0

0

0 0

0

0

0 0

0

0

0 0

668 non-null int64

668 non-null float64

668 non-null float64

668 non-null float64 668 non-null float64

668 non-null float64

668 non-null float64

668 non-null float64

668 non-null float64 668 non-null float64

668 non-null float64

668 non-null float64

668 non-null float64

668 non-null float64

668 non-null float64

668 non-null float64 668 non-null float64

668 non-null float64

668 non-null float64 668 non-null float64

668 non-null float64 668 non-null int64

668 non-null int64

668 non-null int64

668 non-null int64 668 non-null int64

668 non-null int64

668 non-null int64

668 non-null int64 668 non-null int64

STDs:vaginal condylomatosis

STDs:vulvo-perineal condylomatosis

0.06 -0.01 0.04 0.05 -0.04 0.05

0.08 0.03 -0.11 -0.00 0.03 -0.02

0.00 -0.03 0.12 -0.06 -0.04 0.05

006 003 001 -001 -001 -0.01

0.07 -0.02 -0.02 0.03 0.03 -0.05

0.04 0.00 -0.00 -0.02 -0.02 -0.02

0.46 0.12 0.12 0.12

0.32 0.06 0.06 0.06

1.00 0.24 -0.01 -0.00 -0.00 -0.00

0.02 0.13 0.02 -0.01 -0.01 -0.01

0.02 0.08 -0.04 -0.01 -0.01 -0.01

0.02 0.13 -0.04 -0.01 0.15 -0.01

STDs

STDs:HIV STDs:Hepatitis B 0 STDs:HPV STDs: Number of diagnosis 0 Dx:Cancer 0 Dx:CIN 0 Dx:HPV 0 0 Dx Hinselmann 0

Converting object datatype into float64 In [124]: objectCols = [c for c in df.columns if df[c].dtype == "object"] nUniqueObj = list(map(lambda c: df[c].nunique(),objectCols)) d = dict(zip(objectCols, nUniqueObj)) sorted(d.items(), key = lambda x: x[1])newCols = {key: 'float64' for key in d}

> <class 'pandas.core.frame.DataFrame'> Int64Index: 668 entries, 0 to 857 Data columns (total 34 columns):

df = df.astype(newCols)

Age Number of sexual partners First sexual intercourse Num of pregnancies Smokes Smokes (years) Smokes (packs/year) Hormonal Contraceptives Hormonal Contraceptives (years)

STDs:condylomatosis 668 non-null float64 668 non-null float64 STDs:cervical condylomatosis STDs:vaginal condylomatosis STDs:vulvo-perineal condylomatosis 668 non-null float64 STDs:syphilis STDs:pelvic inflammatory disease 668 non-null float64 STDs:genital herpes STDs:molluscum contagiosum STDs:AIDS STDs:HIV

Biopsy dtypes: float64(24), int64(10) memory usage: 182.7 KB Plotting boxplot to detect outliers

plt.setp(ax.get_xticklabels(), rotation=90)

IUD (years)

df.drop(df.index[df['Number of sexual partners'] > 10], inplace = True)

We can see that 'Hinselmann', 'Schiller' and 'Citology' have some correlation with 'biopsy'

0.73 0.01 0.05 0.02 0.02 0.11 0.12 0.07

sns.heatmap(df[top_corr_features].corr(),fmt='.2f',annot = True)

Hormonal Contraceptives (years)

In [126]: df.drop(df.index[df['Age'] > 50], inplace = True)

First sexual intercourse of pregnancies

Removing outliers

Correlation Heatmap

In [127]: | corrmat = round(df.corr(), 2)

0.16 1.00 -0.12 0.16 0.21 0.16 0.13 0.05 0.05 0.07 0.02 0.04 0.01 -0.01 0.51 0.16 -0.09 1.00 0.06 0.14 0.10 0.20 0.25 0.20 0.14 0.05 0.01 -0.03

top_corr_features = corrmat.index

Out[127]: <matplotlib.axes._subplots.AxesSubplot at 0x248f0081f98>

Age - 1.00 0.16 0.39 0.51 0.01 0.14 0.08 0.16 0.36 0.26 0.20 0.02 -0.01 -0.02

Smokes - 0.01 0.21 -0.10 0.06 1.00 0.75 0.53 0.03 0.05 -0.06 -0.04 0.13 0.12 0.07

Smokes (packs/year) - 0.08 0.13 -0.01 0.10 0.53 0.73 1.00 -0.01 0.01 0.03 0.03 0.04 0.05 0.02

plt.figure(figsize=(25,25))

Hormonal Contraceptives - 0.16 0.05 0.03 0.20 0.03 0.01 -0.01 1.00 0.45 0.06 -0.02 -0.03 -0.04 -0.01 0.36 0.05 0.01 0.25 0.05 0.05 0.01 <mark>0.45 1.00</mark> 0.11 -0.00 -0.00 -0.01 0.00 Hormonal Contraceptives (years) 0.26 0.07 -0.05 0.20 -0.06 0.02 0.03 0.06 0.11 1.00 0.20 0.02 -0.05 0.14 -0.04 0.02 0.03 -0.02 -0.00 0.02 0.04 -0.01 0.05 0.13 0.11 0.04 -0.03 -0.00 0.05 0.01 1.00 0.92 0 STDs (number) - -0.01 0.01 0.01 0.01 0.12 0.12 0.05 -0.04 -0.01 0.05 0.01 0.92 1.00 0.90

STDs:condylomatosis - -0.02 -0.01 0.03 -0.03 0.07 0.07 0.02 -0.01 0.00 0.08 0.02 STDs:vaginal condylomatosis - 0.01 -0.06 0.08 -0.00 0.08 0.15 0.06 -0.07 -0.04 0.04 -0.01 0.24 0.36 0.32 STDs:vulvo-perineal condylomatosis - 0.02 -0.01 0.03 -0.03 0.08 0.07 0.03 -0.02 0.00 0.07 0.02 0.73 0.89 0.99 STDs:syphilis - 0.02 0.04 -0.11 0.12 0.12 0.03 0.01 -0.02 -0.00 -0.05 -0.04 0.46 0.32 0.05

STDs:cervical condylomatosis

STDs:pelvic inflammatory disease

STDs:AIDS

Schiller

In [128]:

In [129]:

0.03 0.05 0.00 0.06 0.02 0.01 0.01 0.03 0.02 0.01 0.01 0.12 0.06 0.01 STDs:HIV - 0.01 0.05 0.00 0.03 0.04 0.07 0.05 -0.06 -0.03 0.02 0.03 0.43 0.37 0.11 STDs:HPV - 0.05 0.02 0.04 -0.03 0.06 0.07 -0.01 0.04 0.05 -0.02 -0.01 0.17 0.08 -0.01

Selecting X and y based on correlation

X = df.iloc[:,29:31].valuesy = df.iloc[:,31].values

X test = sc.transform(X test)

rfc.fit(X_train, y_train)

Calculating accuracy of model

X_train = sc.fit_transform(X_train)

Training model using random forest classifier

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(criterion='entropy', random state=0)

min_samples_leaf=1, min_samples_split=2,

Out[129]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',

max_depth=None, max_features='auto', max_leaf_nodes=None,

min weight fraction leaf=0.0, n estimators=10, n jobs=None, oob score=False, random state=0, verbose=0, warm start=False)

min impurity decrease=0.0, min impurity split=None,

sc = StandardScaler()

from sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler

STDs: Number of diagnosis - 0.00 0.04 -0.02 0.05 0.11 0.09 0.04 -0.04 -0.04 0.03 0.01 0.91 0.90 0.71 Dx:Cancer - 0.10 0.02 0.06 0.03 -0.03 -0.02 -0.03 0.01 0.06 0.10 0.09 0.01 -0.01 -0.04 Dx:CIN - -0.02 0.06 -0.04 0.02 -0.03 -0.02 -0.01 0.00 0.01 -0.02 -0.02 -0.02 -0.02 -0.02 0.05 0.05 0.04 0.04 -0.06 -0.05 -0.03 -0.01 -0.02 0.10 0.09 -0.02 -0.03 -0.04 -0.05 -0.07 -0.00 0.03 0.00 -0.02 -0.03 0.05 0.02 0.04 -0.01 0.05 0.08 0.08 Citology - 0.03 0.03 -0.01 -0.04 -0.03 -0.02 -0.03 -0.02 0.02 0.02 0.01 0.07 0.07 0.08

0.04 0.01 0.03 0.03 0.02 0.00 -0.01 0.03 0.09 0.05 -0.00 0.12 0.12 0.12

cervical

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)

0.24 1.00 0.05 -0.01 -0.01 -0.01 0.05 -0.01 -0.01 0.44 -0.02 -0.01 -0.02 -0.02 0.02 0.02 -0.04 -0.04 0.00 -0.01 -0.01 1.00 -0.00 -0.00 0.00 -0.01 -0.01 -0.00 -0.00 1.00 0.23 0.70 0.44 0.11 0.11 0.11 -0.01 -0.04 -0.02 -0.01 -0.01 -0.01 -0.01 -0.04 -0.02 -0.01 -0.01 -0.01 0.02 0.08 0.02 -0.01 -0.01 -0.01

0.27 1.00 -0.00 0.11 -0.01 -0.00 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.01 -0.00 <mark>1.00</mark> 0.07 <mark>0.35</mark> -0.00 <mark>0.36</mark> 0.17 -0.01 -0.02 -0.01 -0.01 0.52 0.11 0.07 1.00 -0.02 -0.02 -0.01 -0.02 0.08 0.12 0.07 0.10 -0.02 -0.01 <mark>0.35</mark> -0.02 <u>1.00</u> -0.01 <u>0.90</u> <u>0.81</u> 0.16 0.16 0.13 0.20 0.01 -0.00 -0.00 -0.02 -0.01 <mark>1.00</mark> -0.01 <mark>0.43</mark> -0.01 -0.02 -0.02 0.07 0.08 -0.01 -0.01 0.08 0.16 -0.01 0.17 0.11 0.11 -0.01 -0.02 0.12 0.16 -0.02 0.16 0.09 0.10 -0.01 -0.01 0.07 0.13 -0.02 0.13 0.13 0.17 0.39 0.10 -0.01 -0.01 0.10 0.20 0.07 0.21 0.16 0.51 STDs

STDs:Hepatitis B

STDs:HIV

STDs:genital herpes

.01 -0.03 0.05 0.00 0.10 -0.02 0.08 0.05 -0.05 0.04 -0.03 0.04

0.05 -0.01 0.02 0.04 0.02 0.06 0.03 0.05 -0.07 -0.02 0.03 0.01

0.00 0.01 0.04 -0.02 0.06 -0.04 0.03 0.04 -0.00 -0.01 -0.01 0.03

0.03 -0.04 -0.03 0.05 0.03 0.02 0.06 0.04 0.03 0.06 -0.04 0.03

0.04 0.10 0.06 0.11 -0.03 -0.03 -0.00 -0.06 0.00 0.01 -0.03 0.02

0.07 0.13 0.07 0.09 -0.02 -0.02 -0.01 -0.05 -0.02 0.01 -0.02 0.00

0.05 0.14 -0.01 0.04 -0.03 -0.01 -0.03 -0.03 -0.03 -0.02 -0.03 -0.01

0.06 -0.05 0.04 -0.04 0.01 0.00 0.03 -0.01 0.05 0.02 -0.02 0.03

0.03 -0.02 0.05 -0.04 0.06 0.01 0.07 -0.02 0.02 0.09 0.02 0.09

0.03 -0.01 -0.01 0.01 0.09 -0.02 0.01 0.09 -0.01 0.04 0.01 -0.00

0.43 0.12 0.17 0.91 0.01 -0.02 0.02 -0.02 0.05 0.11 0.07 0.12

0.37 0.13 0.08 0.90 -0.01 -0.02 -0.01 -0.03 0.08 0.13 0.07 0.12

0.01 -0.00 -0.00 0.23 -0.01 -0.01 -0.01 -0.01 -0.02 -0.02 -0.02 -0.02

-0.04 -0.02 -0.04 -0.04 0.08 0.13 0.08 0.12

-0.04 -0.02 -0.04 -0.04 0.08 0.13 0.08 0.13

0.01 -0.00 -0.00 0.11 -0.01 -0.00 -0.01 -0.01 -0.01 -0.01 -0.01 1.00 0.27 -0.01 0.52 -0.02 -0.01 -0.02 -0.02 0.08 0.11 0.10 0.10

0.11 0.09 0.13 0.16

In [114]: from sklearn.model_selection import cross_val_score
 from sklearn.model_selection import KFold
 k = 5

kf = KFold(n_splits=k, random_state=None)
result = cross_val_score(rfc , X_train, y_train, cv = kf)
print("Accuracy: {:.2f}%".format(result.mean()*100))
Accuracy: 94.14%