smoking-prediction

October 7, 2024

Importing Important Libraries

[2]:
```
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import r2_score, classification_report as cr,␣
 ↪confusion_matrix as cm
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import matplotlib.pyplot as plt
```

Reading Dataset

[3]:
```
z = pd.read_csv(r"C:\Users\skj_h\OneDrive\Desktop\smoking.csv")
z
```

[3]:

	ID	gender	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	\
0	0	F	40	155	60	81.3	1.2	
1	1	F	40	160	60	81.0	0.8	
2	2	M	55	170	60	80.0	0.8	
3	3	M	40	165	70	88.0	1.5	
4	4	F	40	155	60	86.0	1.0	
…	…	…	…	…	…	…	…	
55687	55676	F	40	170	65	75.0	0.9	
55688	55681	F	45	160	50	70.0	1.2	
55689	55683	F	55	160	50	68.5	1.0	
55690	55684	M	60	165	60	78.0	0.8	
55691	55691	M	55	160	65	85.0	0.9	

	eyesight(right)	hearing(left)	hearing(right)	…	hemoglobin	\
0	1.0	1.0	1.0	…	12.9	
1	0.6	1.0	1.0	…	12.7	
2	0.8	1.0	1.0	…	15.8	
3	1.5	1.0	1.0	…	14.7	
4	1.0	1.0	1.0	…	12.5	

```
…                …            …              …  …         …
55687            0.9          1.0            1.0  …      12.3
55688            1.2          1.0            1.0  …      14.0
55689            1.2          1.0            1.0  …      12.4
55690            1.0          1.0            1.0  …      14.4
55691            0.7          1.0            1.0  …      15.0

       Urine protein  serum creatinine   AST   ALT   Gtp  oral  dental caries  \
0                1.0               0.7  18.0  19.0  27.0     Y              0
1                1.0               0.6  22.0  19.0  18.0     Y              0
2                1.0               1.0  21.0  16.0  22.0     Y              0
3                1.0               1.0  19.0  26.0  18.0     Y              0
4                1.0               0.6  16.0  14.0  22.0     Y              0
…                …                 …    …     …     …     …              …
55687            1.0               0.6  14.0   7.0  10.0     Y              1
55688            1.0               0.9  20.0  12.0  14.0     Y              0
55689            1.0               0.5  17.0  11.0  12.0     Y              0
55690            1.0               0.7  20.0  19.0  18.0     Y              0
55691            1.0               0.8  26.0  29.0  41.0     Y              0

       tartar  smoking
0           Y        0
1           Y        0
2           N        1
3           Y        0
4           N        0
…       …       …
55687       Y        0
55688       Y        0
55689       N        0
55690       N        0
55691       Y        1

[55692 rows x 27 columns]
```

Counting of null value

[4]:
```
z.isnull().sum()
```

[4]:
```
ID                     0
gender                 0
age                    0
height(cm)             0
weight(kg)             0
waist(cm)              0
eyesight(left)         0
eyesight(right)        0
```

```
hearing(left)          0
hearing(right)         0
systolic               0
relaxation             0
fasting blood sugar    0
Cholesterol            0
triglyceride           0
HDL                    0
LDL                    0
hemoglobin             0
Urine protein          0
serum creatinine       0
AST                    0
ALT                    0
Gtp                    0
oral                   0
dental caries          0
tartar                 0
smoking                0
dtype: int64
```

Shape of Dataset

[5]:
```
z.shape
```

[5]:
```
(55692, 27)
```

Size of Dataset

[6]:
```
z.size
```

[6]:
```
1503684
```

Number of Dimension of dataset

[7]:
```
z.ndim
```

[7]:
```
2
```

Counting of non Null Values and Datatype

[8]:
```
z.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 55692 entries, 0 to 55691
Data columns (total 27 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ID                   55692 non-null  int64
```

```
 1   gender               55692 non-null  object
 2   age                  55692 non-null  int64
 3   height(cm)           55692 non-null  int64
 4   weight(kg)           55692 non-null  int64
 5   waist(cm)            55692 non-null  float64
 6   eyesight(left)       55692 non-null  float64
 7   eyesight(right)      55692 non-null  float64
 8   hearing(left)        55692 non-null  float64
 9   hearing(right)       55692 non-null  float64
 10  systolic             55692 non-null  float64
 11  relaxation           55692 non-null  float64
 12  fasting blood sugar  55692 non-null  float64
 13  Cholesterol          55692 non-null  float64
 14  triglyceride         55692 non-null  float64
 15  HDL                  55692 non-null  float64
 16  LDL                  55692 non-null  float64
 17  hemoglobin           55692 non-null  float64
 18  Urine protein        55692 non-null  float64
 19  serum creatinine     55692 non-null  float64
 20  AST                  55692 non-null  float64
 21  ALT                  55692 non-null  float64
 22  Gtp                  55692 non-null  float64
 23  oral                 55692 non-null  object
 24  dental caries        55692 non-null  int64
 25  tartar               55692 non-null  object
 26  smoking              55692 non-null  int64
dtypes: float64(18), int64(6), object(3)
memory usage: 11.5+ MB
```

Datatype of respective columns

[9]:
```
z.dtypes
```

[9]:
```
ID                     int64
gender                object
age                    int64
height(cm)             int64
weight(kg)             int64
waist(cm)            float64
eyesight(left)       float64
eyesight(right)      float64
hearing(left)        float64
hearing(right)       float64
systolic             float64
relaxation           float64
fasting blood sugar  float64
Cholesterol          float64
triglyceride         float64
```

```
HDL                    float64
LDL                    float64
hemoglobin             float64
Urine protein          float64
serum creatinine       float64
AST                    float64
ALT                    float64
Gtp                    float64
oral                    object
dental caries            int64
tartar                  object
smoking                  int64
dtype: object
```

Removing of ID column from Dataset

[10]:
```
z.drop(["ID"], axis = 1, inplace = True)
```

[11]:
```
b = z.copy()
for i in b:
    if(b[i].dtype == "object"):
        b.drop([i], axis = 1, inplace = True)
b
```

[11]:

	age	height(cm)	weight(kg)	waist(cm)	eyesight(left)	\
0	40	155	60	81.3	1.2	
1	40	160	60	81.0	0.8	
2	55	170	60	80.0	0.8	
3	40	165	70	88.0	1.5	
4	40	155	60	86.0	1.0	
…	…	…	…	…	…	
55687	40	170	65	75.0	0.9	
55688	45	160	50	70.0	1.2	
55689	55	160	50	68.5	1.0	
55690	60	165	60	78.0	0.8	
55691	55	160	65	85.0	0.9	

	eyesight(right)	hearing(left)	hearing(right)	systolic	relaxation	\
0	1.0	1.0	1.0	114.0	73.0	
1	0.6	1.0	1.0	119.0	70.0	
2	0.8	1.0	1.0	138.0	86.0	
3	1.5	1.0	1.0	100.0	60.0	
4	1.0	1.0	1.0	120.0	74.0	
…	…	…	…	…	…	
55687	0.9	1.0	1.0	110.0	68.0	
55688	1.2	1.0	1.0	101.0	62.0	
55689	1.2	1.0	1.0	117.0	72.0	

```
55690              1.0            1.0             1.0     133.0        76.0
55691              0.7            1.0             1.0     124.0        75.0

       …   HDL    LDL  hemoglobin  Urine protein  serum creatinine   AST  \
0      …  73.0  126.0        12.9            1.0               0.7  18.0
1      …  42.0  127.0        12.7            1.0               0.6  22.0
2      …  55.0  151.0        15.8            1.0               1.0  21.0
3      …  45.0  226.0        14.7            1.0               1.0  19.0
4      …  62.0  107.0        12.5            1.0               0.6  16.0
…    …   …    …          …             …                 …    …
55687  …  75.0  118.0        12.3            1.0               0.6  14.0
55688  …  73.0   79.0        14.0            1.0               0.9  20.0
55689  …  79.0   63.0        12.4            1.0               0.5  17.0
55690  …  48.0  146.0        14.4            1.0               0.7  20.0
55691  …  34.0  150.0        15.0            1.0               0.8  26.0

        ALT   Gtp  dental caries  smoking
0      19.0  27.0              0        0
1      19.0  18.0              0        0
2      16.0  22.0              0        1
3      26.0  18.0              0        0
4      14.0  22.0              0        0
…     …   …             …       …
55687   7.0  10.0              1        0
55688  12.0  14.0              0        0
55689  11.0  12.0              0        0
55690  19.0  18.0              0        0
55691  29.0  41.0              0        1

[55692 rows x 23 columns]
```

Correlation cofficient

[12]:
```
b.corr()["smoking"].sort_values(ascending = False)
```

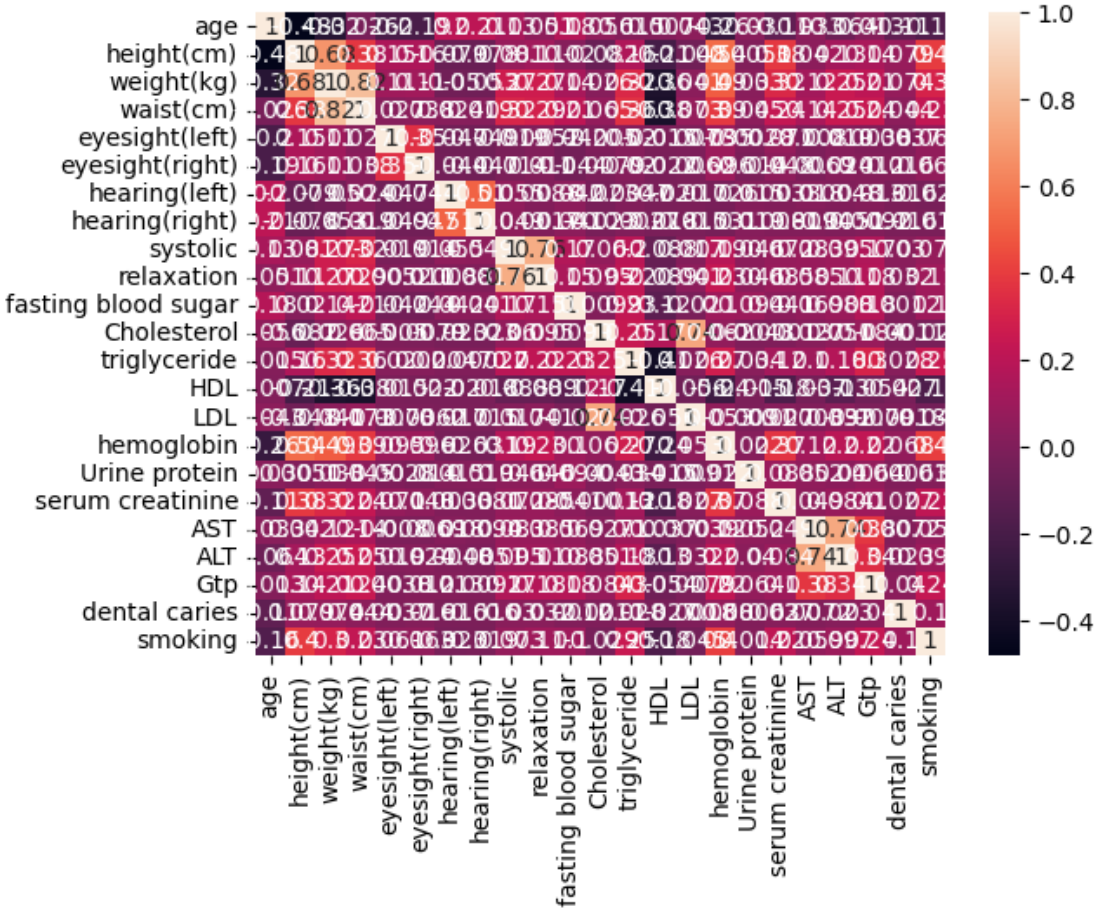
[12]:
```
smoking               1.000000
hemoglobin            0.400678
height(cm)            0.396675
weight(kg)            0.302780
triglyceride          0.251799
Gtp                   0.236619
waist(cm)             0.226259
serum creatinine      0.216812
relaxation            0.108309
dental caries         0.103857
fasting blood sugar   0.100279
ALT                   0.097338
```

```
systolic               0.073109
eyesight(right)        0.063017
eyesight(left)         0.061204
AST                    0.059253
Urine protein          0.014267
hearing(right)        -0.018855
hearing(left)         -0.023209
Cholesterol           -0.028548
LDL                   -0.045220
age                   -0.162557
HDL                   -0.178470
Name: smoking, dtype: float64
```

Heatmap for showing coorelation cofficient of respective dataset

[13]:
```
sns.heatmap(b.corr(), annot = True, alpha = 1)
```

[13]:
```
<Axes: >
```



Knowing column names present in dataset

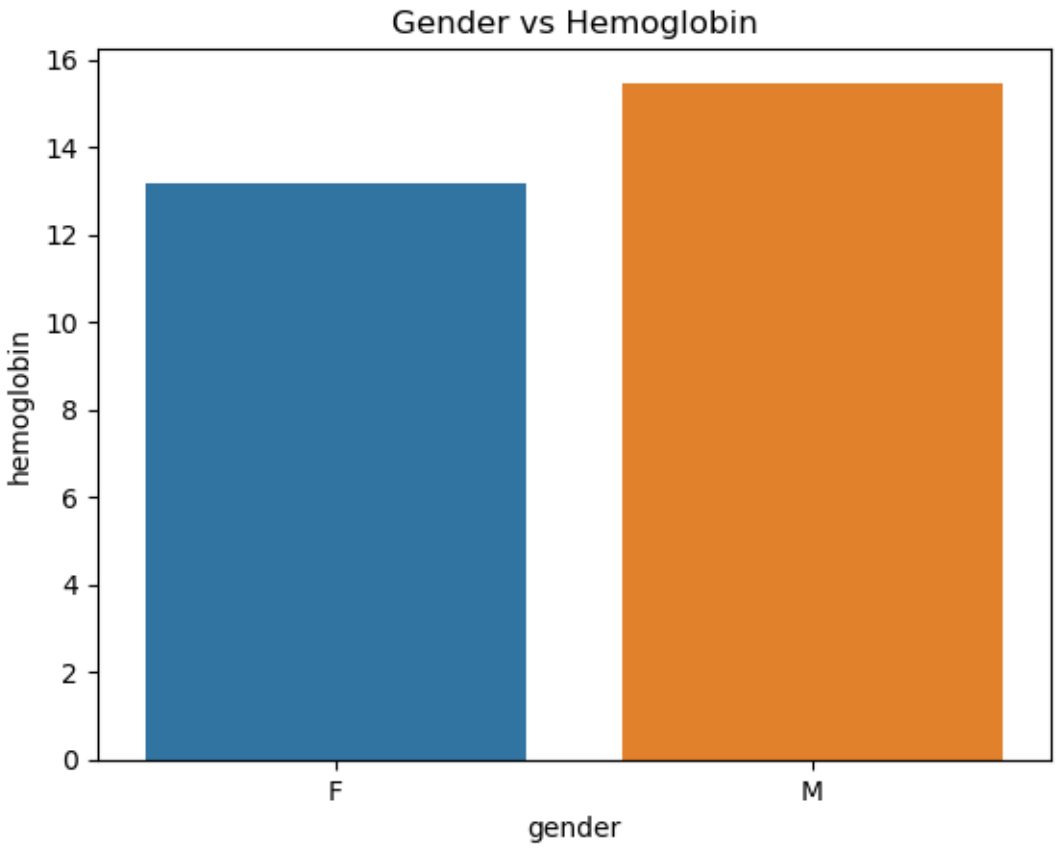
[14]:
```
z.columns
```

[14]:
```
Index(['gender', 'age', 'height(cm)', 'weight(kg)', 'waist(cm)',
       'eyesight(left)', 'eyesight(right)', 'hearing(left)', 'hearing(right)',
       'systolic', 'relaxation', 'fasting blood sugar', 'Cholesterol',
       'triglyceride', 'HDL', 'LDL', 'hemoglobin', 'Urine protein',
       'serum creatinine', 'AST', 'ALT', 'Gtp', 'oral', 'dental caries',
       'tartar', 'smoking'],
      dtype='object')
```

Bivariate analysis

Ploting Barplot between Gender and Hemoglobin

[16]:
```
plt.title("Gender vs Hemoglobin");
sns.barplot(x = z["gender"], y = z["hemoglobin"], data = z, hue = z["gender"]);
```

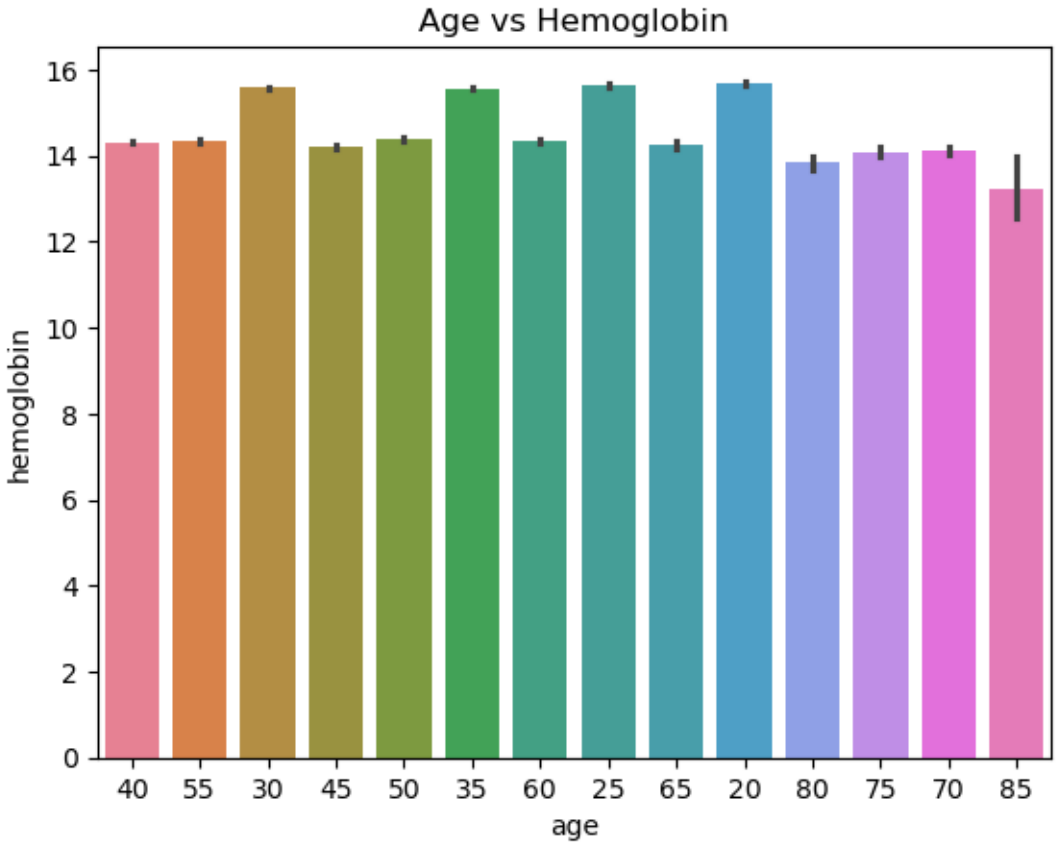


[17]:
```
z["age"] = z["age"].astype(str)
```

Ploting barplot between Age and Hemoglobin

[18]:
```
plt.title("Age vs Hemoglobin")
sns.barplot(x = z["age"], y = z["hemoglobin"], data = z, hue = z["age"])
```

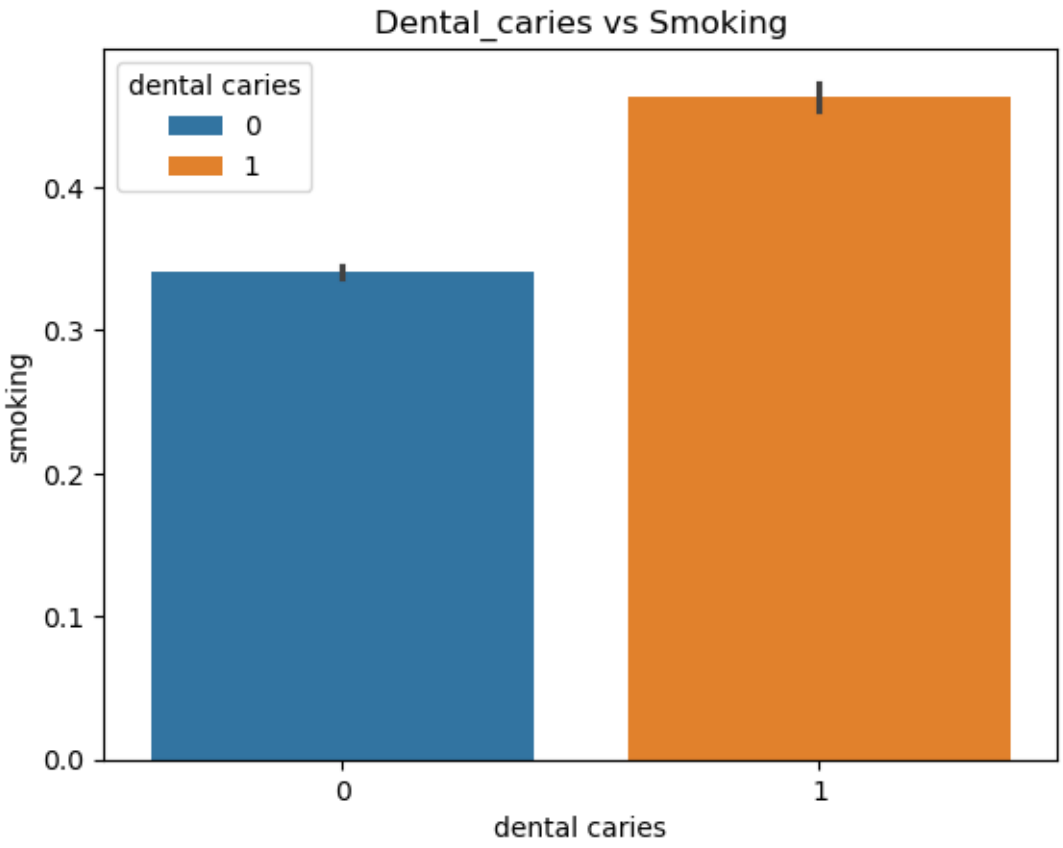
[18]:
```
<Axes: title={'center': 'Age vs Hemoglobin'}, xlabel='age', ylabel='hemoglobin'>
```



Ploting barplot between Dental caries and Smoking

[19]:
```
plt.title("Dental_caries vs Smoking")
sns.barplot(x = z["dental caries"], y = z["smoking"], data = z, hue = z["dental␣
↪caries"])
```

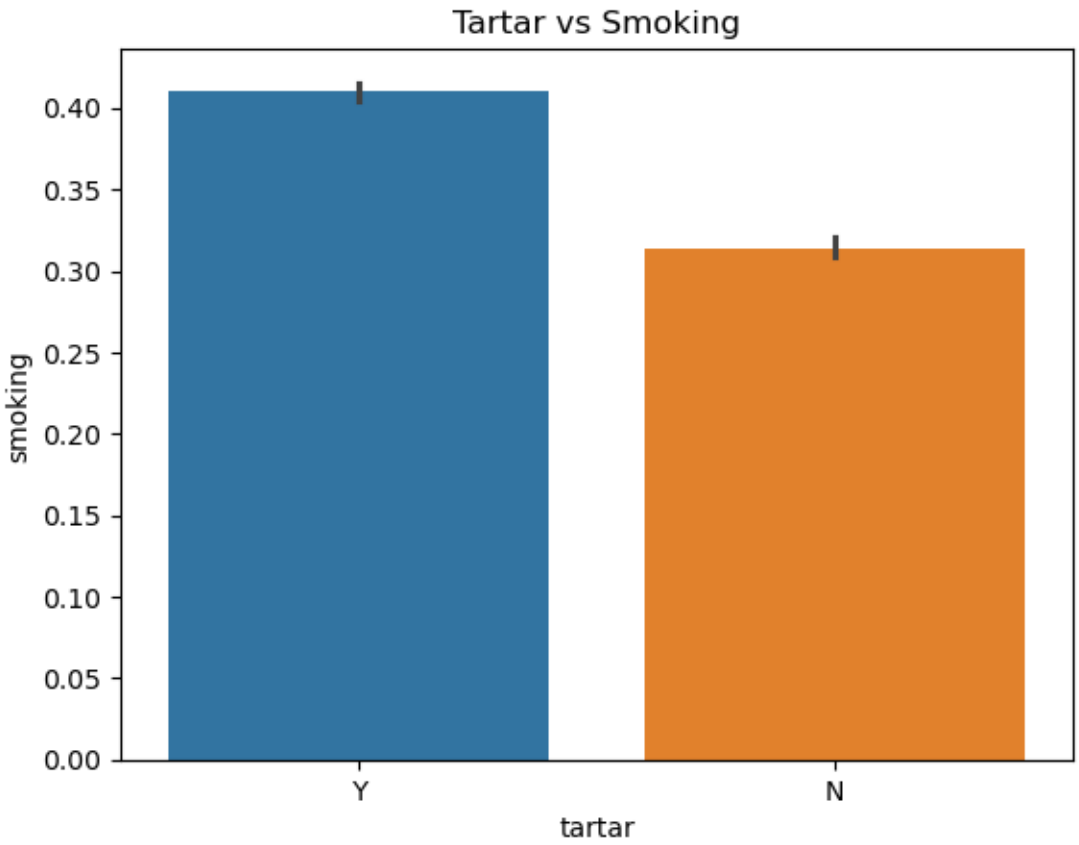
[19]:
```
<Axes: title={'center': 'Dental_caries vs Smoking'}, xlabel='dental caries',
ylabel='smoking'>
```



Ploting barplot between Tartar and Smoking

[20]:
```
plt.title("Tartar vs Smoking")
sns.barplot(x = z["tartar"], y = z["smoking"], data = z, hue = z["tartar"])
```

[20]:
```
<Axes: title={'center': 'Tartar vs Smoking'}, xlabel='tartar', ylabel='smoking'>
```



[21]:
```
z["smoking"].value_counts()
```

[21]:
```
smoking
0    35237
1    20455
Name: count, dtype: int64
```

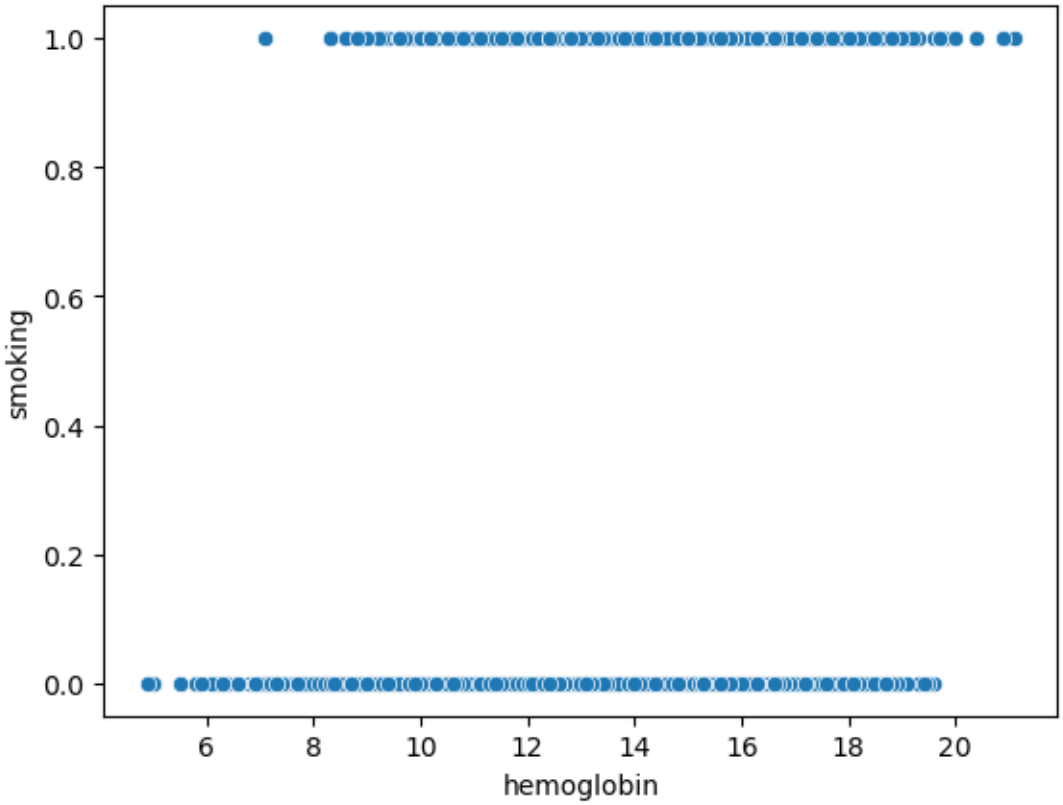
[22]:
```
z.columns
```

[22]:
```
Index(['gender', 'age', 'height(cm)', 'weight(kg)', 'waist(cm)',
       'eyesight(left)', 'eyesight(right)', 'hearing(left)', 'hearing(right)',
       'systolic', 'relaxation', 'fasting blood sugar', 'Cholesterol',
       'triglyceride', 'HDL', 'LDL', 'hemoglobin', 'Urine protein',
       'serum creatinine', 'AST', 'ALT', 'Gtp', 'oral', 'dental caries',
       'tartar', 'smoking'],
      dtype='object')
```

Ploting scatterplot between Hemoglobin and Smoking

[23]:
```
sns.scatterplot(x = z["hemoglobin"], y = z["smoking"], data = z)
```

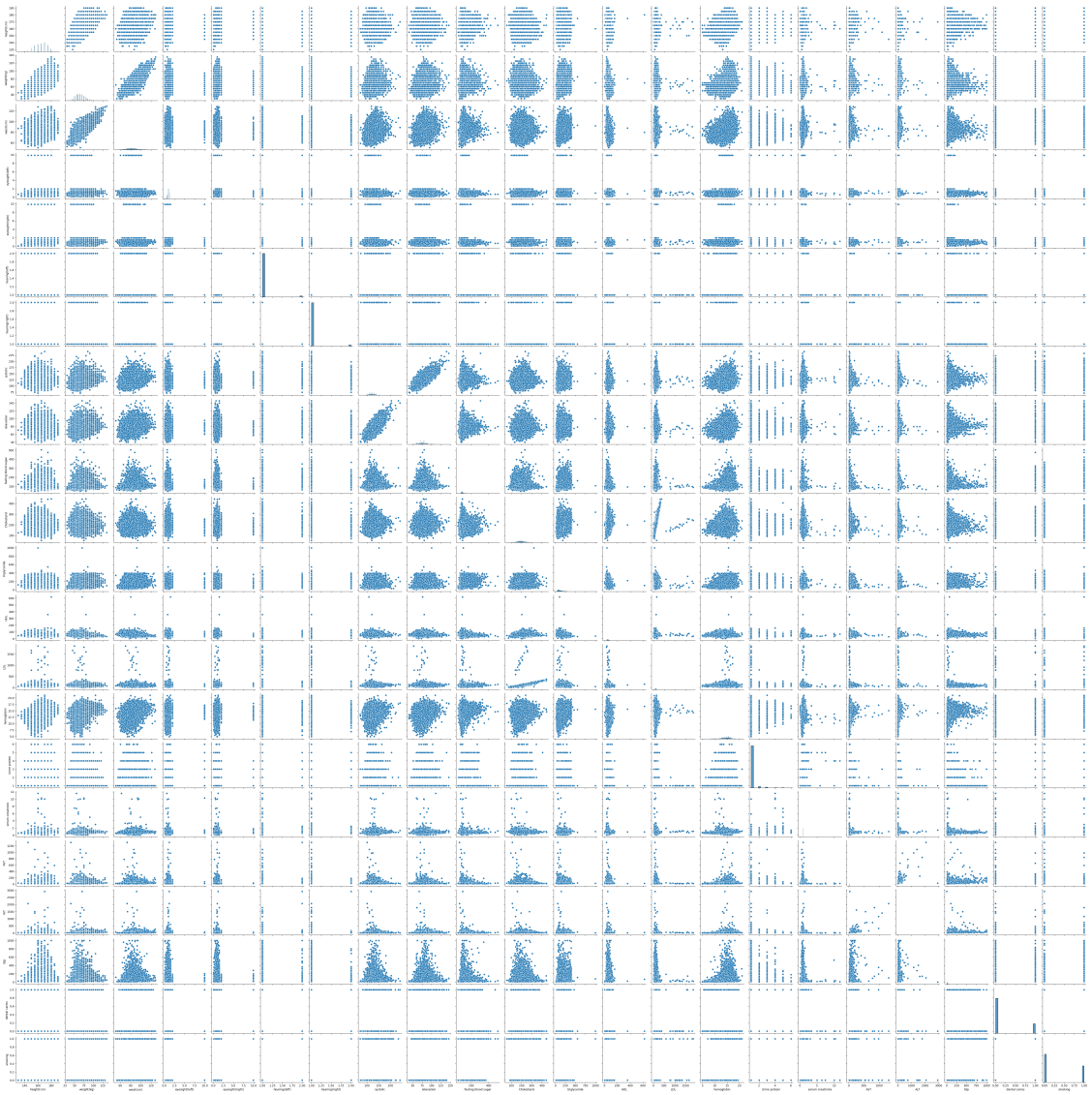
[23]: <Axes: xlabel='hemoglobin', ylabel='smoking'>



Multivariate analysis

[24]:
```
sns.pairplot(z)
```

[24]: <seaborn.axisgrid.PairGrid at 0x2e489059f40>



Modelling

[25]:
```
z.columns
```

[25]:
```
Index(['gender', 'age', 'height(cm)', 'weight(kg)', 'waist(cm)',
       'eyesight(left)', 'eyesight(right)', 'hearing(left)', 'hearing(right)',
       'systolic', 'relaxation', 'fasting blood sugar', 'Cholesterol',
       'triglyceride', 'HDL', 'LDL', 'hemoglobin', 'Urine protein',
       'serum creatinine', 'AST', 'ALT', 'Gtp', 'oral', 'dental caries',
       'tartar', 'smoking'],
      dtype='object')
```

[26]:
```
b.corr()["smoking"].sort_values(ascending = False)
```

[26]:
```
smoking                1.000000
hemoglobin             0.400678
height(cm)             0.396675
weight(kg)             0.302780
triglyceride           0.251799
Gtp                    0.236619
waist(cm)              0.226259
serum creatinine       0.216812
relaxation             0.108309
dental caries          0.103857
fasting blood sugar    0.100279
ALT                    0.097338
systolic               0.073109
eyesight(right)        0.063017
eyesight(left)         0.061204
AST                    0.059253
Urine protein          0.014267
hearing(right)        -0.018855
hearing(left)         -0.023209
Cholesterol           -0.028548
LDL                   -0.045220
age                   -0.162557
HDL                   -0.178470
Name: smoking, dtype: float64
```

[27]:
```
X = z[["hemoglobin", "smoking"]]
Y = z["smoking"]
```

[28]:
```
x_train, x_test, y_train, y_test = train_test_split(X,  Y, train_size = 0.7,␣
 ↪test_size = 0.3, random_state = 100)
```

[29]:
```
x_train = x_train.drop(["smoking"], axis = 1)
x_test = x_test.drop(["smoking"], axis = 1)
```

[30]:
```
y_train = np.array(y_train).reshape(-1, 1)
y_test = np.array(y_test).reshape(-1, 1)
```

[31]:
```
n = RandomForestClassifier(n_estimators = 500)
n.fit(x_train, y_train)
```

[31]:
```
RandomForestClassifier(n_estimators=500)
```

Evaluating training dataset

[32]:
```
y_predict_train = n.predict(x_train)
print(cr(y_true = y_train, y_pred = y_predict_train))
```

```
              precision    recall  f1-score   support
```

```
              0       0.77      0.72      0.74     24643
              1       0.57      0.63      0.60     14341

       accuracy                           0.69     38984
      macro avg       0.67      0.67      0.67     38984
   weighted avg       0.69      0.69      0.69     38984
```

[33]:
```
cm(y_true = y_train, y_pred = y_predict_train)
```

[33]:
```
array([[17699,  6944],
       [ 5285,  9056]], dtype=int64)
```

Evaluating testing dataset

[34]:
```
n = RandomForestClassifier(n_estimators = 500)
n.fit(x_test, y_test)
```

[34]:
```
RandomForestClassifier(n_estimators=500)
```

[35]:
```
y_predict_test = n.predict(x_test)
print(cr(y_true = y_test, y_pred = y_predict_test))
```

```
              precision    recall  f1-score   support

           0       0.76      0.74      0.75     10594
           1       0.57      0.60      0.58      6114

    accuracy                           0.69     16708
   macro avg       0.66      0.67      0.66     16708
weighted avg       0.69      0.69      0.69     16708
```

[36]:
```
cm(y_true = y_test, y_pred = y_predict_test)
```

[36]:
```
array([[7809, 2785],
       [2469, 3645]], dtype=int64)
```

[ ]:

[ ]: