Cox Deliverable 2

CIS 368

Data Dictionary:

- gender: Male or Female response from participant
- age: Age in years of the individual
- avg_glucose: Average glucose levels of the individual
- bmi: Body mass index of the individual
- smoking_status: is the individual a former smoker, never smoked, or currently smokes
- stroke_poss: is the individual likely to have a stroke

Problem Statement:

How likely are males and females likely to have a stroke based on their smoking status, average glucose levels, and their body mass index.

Predictor Variables:

- gender
- age
- avg_glucose
- bmi
- smoking_status

Target Variable:

- stroke_poss

In [46]:
```
%matplotlib inline
import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import matplotlib.pylab as plt
from sklearn.metrics import confusion_matrix
from sklearn.decomposition import PCA
from sklearn import preprocessing
```

In [47]:
```
stroke = pd.read_csv('stroke.csv')
stroke.head()
```

Out[47]:

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_gl
0	9046	Male	67.0	0	1	Yes	Private	Urban	
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	
2	31112	Male	80.0	0	1	Yes	Private	Rural	
3	60182	Female	49.0	0	0	Yes	Private	Urban	
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	

In [48]:
```
stroke2 = stroke.drop(['id', 'hypertension', 'heart_disease', 'ever_married', 'work_ty
stroke2
```

Out[48]:

	gender	age	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	228.69	36.6	formerly smoked	1
1	Female	61.0	202.21	NaN	never smoked	1
2	Male	80.0	105.92	32.5	never smoked	1
3	Female	49.0	171.23	34.4	smokes	1
4	Female	79.0	174.12	24.0	never smoked	1
...	...	...	...	...	...	...
5105	Female	80.0	83.75	NaN	never smoked	0
5106	Female	81.0	125.20	40.0	never smoked	0
5107	Female	35.0	82.99	30.6	never smoked	0
5108	Male	51.0	166.29	25.6	formerly smoked	0
5109	Female	44.0	85.28	26.2	Unknown	0

5110 rows × 6 columns

In [49]:
```
stroke3 = stroke2.rename(columns={'stroke': 'stroke_possibility'})
stroke3
```

Out[49]:

	gender	age	avg_glucose_level	bmi	smoking_status	stroke_possibility
0	Male	67.0	228.69	36.6	formerly smoked	1
1	Female	61.0	202.21	NaN	never smoked	1
2	Male	80.0	105.92	32.5	never smoked	1
3	Female	49.0	171.23	34.4	smokes	1
4	Female	79.0	174.12	24.0	never smoked	1
...	...	...	...	...	...	...
5105	Female	80.0	83.75	NaN	never smoked	0
5106	Female	81.0	125.20	40.0	never smoked	0
5107	Female	35.0	82.99	30.6	never smoked	0
5108	Male	51.0	166.29	25.6	formerly smoked	0
5109	Female	44.0	85.28	26.2	Unknown	0

5110 rows × 6 columns

In [50]:
```
stroke3['gender'].replace(['Male', 'Female', 'Other'],
                          [0, 1, 2], inplace=True)
stroke3
```

Out[50]:

	gender	age	avg_glucose_level	bmi	smoking_status	stroke_possibility
0	0	67.0	228.69	36.6	formerly smoked	1
1	1	61.0	202.21	NaN	never smoked	1
2	0	80.0	105.92	32.5	never smoked	1
3	1	49.0	171.23	34.4	smokes	1
4	1	79.0	174.12	24.0	never smoked	1
...	...	...	...	...	...	...
5105	1	80.0	83.75	NaN	never smoked	0
5106	1	81.0	125.20	40.0	never smoked	0
5107	1	35.0	82.99	30.6	never smoked	0
5108	0	51.0	166.29	25.6	formerly smoked	0
5109	1	44.0	85.28	26.2	Unknown	0

5110 rows × 6 columns

In [51]:
```
stroke3['smoking_status'].replace(['formerly smoked', 'never smoked', 'smokes', 'Unkno
                          [0, 1, 2, 3], inplace=True)
stroke3
```

Out[51]:

	gender	age	avg_glucose_level	bmi	smoking_status	stroke_possibility
0	0	67.0	228.69	36.6	0	1
1	1	61.0	202.21	NaN	1	1
2	0	80.0	105.92	32.5	1	1
3	1	49.0	171.23	34.4	2	1
4	1	79.0	174.12	24.0	1	1
...	...	...	...	...	...	...
5105	1	80.0	83.75	NaN	1	0
5106	1	81.0	125.20	40.0	1	0
5107	1	35.0	82.99	30.6	1	0
5108	0	51.0	166.29	25.6	0	0
5109	1	44.0	85.28	26.2	3	0

5110 rows × 6 columns

In [52]:
```
stroke4 = stroke3.dropna()
```

In [53]:
```
stroke4
```

Out[53]:

	gender	age	avg_glucose_level	bmi	smoking_status	stroke_possibility
0	0	67.0	228.69	36.6	0	1
2	0	80.0	105.92	32.5	1	1
3	1	49.0	171.23	34.4	2	1
4	1	79.0	174.12	24.0	1	1
5	0	81.0	186.21	29.0	0	1
...	...	...	...	...	...	...
5104	1	13.0	103.08	18.6	3	0
5106	1	81.0	125.20	40.0	1	0
5107	1	35.0	82.99	30.6	1	0
5108	0	51.0	166.29	25.6	0	0
5109	1	44.0	85.28	26.2	3	0

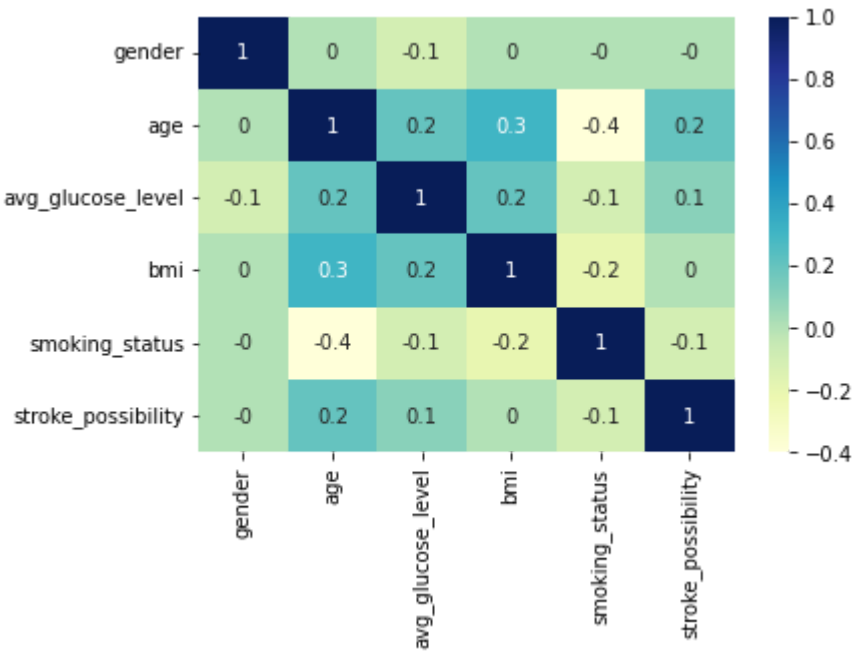
4909 rows × 6 columns

In [56]:
```
stroke4.corr().round(2)
```

Out[56]:

	gender	age	avg_glucose_level	bmi	smoking_status	stroke_possibility
gender	1.00	0.03	-0.05	0.03	-0.04	-0.01
age	0.03	1.00	0.24	0.33	-0.39	0.23
avg_glucose_level	-0.05	0.24	1.00	0.18	-0.11	0.14
bmi	0.03	0.33	0.18	1.00	-0.24	0.04
smoking_status	-0.04	-0.39	-0.11	-0.24	1.00	-0.08
stroke_possibility	-0.01	0.23	0.14	0.04	-0.08	1.00

In [57]:
```
import seaborn as sns
dp = sns.heatmap(stroke4.corr().round(1), cmap="YlGnBu", annot=True)
```



In [69]:
```
pcs = PCA(n_components=2)
pcs.fit(stroke4[['bmi','age']])
```

Out[69]:
```
PCA(n_components=2)
```

In [70]:
```
pcsSummary = pd.DataFrame({'Standard Deviation': np.sqrt(pcs.explained_variance_),
                           'Proportion of variance': pcs.explained_variance_ratio_,
                          'Cumulative proportion': np.cumsum(pcs.explained_variance_rat
pcsSummary
```

Out[70]:

	Standard Deviation	Proportion of variance	Cumulative proportion
0	22.724533	0.905306	0.905306
1	7.349501	0.094694	1.000000

In [71]:
```
pcsSummary = pcsSummary.transpose()
```

In [72]:
```
pcsSummary.columns = ['PC1', 'PC2']
pcsSummary.round(4)
```

Out[72]:

	PC1	PC2
Standard Deviation	22.7245	7.3495
Proportion of variance	0.9053	0.0947
Cumulative proportion	0.9053	1.0000

In [73]:
```
pcs = PCA()

pcs.fit(stroke4.iloc[:, 3:].dropna(axis=0))
```

Out[73]:
```
PCA()
```

In [74]:
```
pcsSummary_df = pd.DataFrame({'Standard Deviation': np.sqrt(pcs.explained_variance_),
                              'Proportion of variance': pcs.explained_variance_ratio_,
                             'Cumulative proportion': np.cumsum(pcs.explained_variance_rat
pcsSummary_df = pcsSummary.transpose()
pcsSummary_df.columns = ['PC{}'.format(i) for i in range(1, len(pcsSummary_df.columns)
pcsSummary_df.round(4)
```

Out[74]:

	PC1	PC2	PC3
PC1	22.7245	0.9053	0.9053
PC2	7.3495	0.0947	1.0000

In [75]:
```
pcsComponents_df = pd.DataFrame(pcs.components_.transpose(),
                                columns=pcsSummary_df.columns,
                                index=stroke4.iloc[:,3:].columns)
```

In [76]:
```
pcsComponents_df.iloc[:]
```

Out[76]:

	PC1	PC2	PC3
bmi	0.999445	0.033320	-0.000651
smoking_status	-0.033308	0.999355	0.013412
stroke_possibility	0.001097	-0.013383	0.999910

One of the predictive approaches I am planning to use to model my data is logistic regression becuase as I was going through the EDA and the PDA for this I was noticing how well it would fit given the data I have. It felt similar to Assignment 6 it terms of progression. I think this will be my primary approach. Another one I want to utilize is a decision tree, I feel this will be beneficial as well given the variables I have could all lead to the possibility of having a stroke and seeing which variables that are omitted could help me see whether there are a combination of variables that would decrease the possibility of a stroke.

In [ ]:
```

```