DSC-530: Predictive Modeling
Week 8 Assignment: Final Predictive Modeling Project (GLM)
Author: Jonathan Ibifubara Pollyn

College of Science, Engineering and Technology, Grand Canyon University

This dataset is collected from kaggle.com called advertising.csv, and it was last updated by fayomi. It is a dataset for practice Data Analysis and Logistic Regression Prediction; the data as of the date of the project is four years old. The data is used to predict the number of times a company ad will be clicked based on the advertisement. The predictor used in the Age and Area income in this data while the target is the clicked on Ad.

In [217…
```
#Importing the required packages
import pandas as pd
import numpy as np
import statsmodels.api as sm
from scipy import stats
import statsmodels.tools.tools as stattools
import matplotlib.pyplot as plt
import math
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

In [206…
```
#Get Dataset needed for the project
getData = pd.read_csv('C:/School/DSC-530/Final Projects Datasets/advertising.csv')
getData.head(5)
```

Out[206…

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Male	Country	Timestamp	Clicked on Ad
0	68.95	35	61833.90	256.09	Cloned 5thgeneration orchestration	Wrightburgh	0	Tunisia	2016-03-27 00:53:11	0
1	80.23	31	68441.85	193.77	Monitored national standardization	West Jodi	1	Nauru	2016-04-04 01:39:02	0
2	69.47	26	59785.94	236.50	Organic bottom-line service-desk	Davidton	0	San Marino	2016-03-13 20:35:42	0
3	74.15	29	54806.18	245.89	Triple-buffered reciprocal time-frame	West Terrifurt	1	Italy	2016-01-10 02:31:19	0
4	68.37	35	73889.99	225.58	Robust logistical utilization	South Manuel	0	Iceland	2016-06-03 03:36:18	0

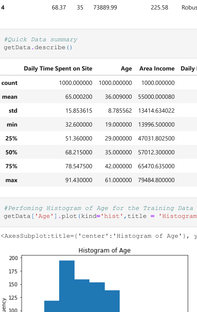
In [207…
```
#Quick Data summary
getData.describe()
```

Out[207…

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Male	Clicked on Ad
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	65.000200	36.009000	55000.000080	180.000100	0.481000	0.50000
std	15.853615	8.785562	13414.634022	43.902339	0.499889	0.50025
min	32.600000	19.000000	13996.500000	104.780000	0.000000	0.00000
25%	51.360000	29.000000	47031.802500	138.830000	0.000000	0.00000
50%	68.215000	35.000000	57012.300000	183.130000	0.000000	0.50000
75%	78.547500	42.000000	65470.635000	218.792500	1.000000	1.00000
max	91.430000	61.000000	79484.800000	269.960000	1.000000	1.00000

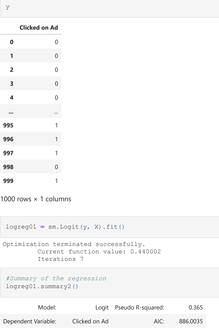
In [209…
```
#Perfoming Histogram of Age for the Training Data
getData['Age'].plot(kind='hist',title = 'Histogram of Age')
```

Out[209…
```
<AxesSubplot:title={'center':'Histogram of Age'}, ylabel='Frequency'>
```



In [219…
```
#Obtaining a jointplot that shows the 'Daily Time Spent on Site' vs. 'Daily Internet Usage'""
sns.jointplot(data=getData,x='Daily Time Spent on Site',y='Daily Internet Usage',color='blue')
```

Out[219…
```
<seaborn.axisgrid.JointGrid at 0x208f9246a00>
```



Interpretation of the graph is found on the final report documentation

In [210…
```
#Isolating the predictor variables
X = pd.DataFrame(getData[['Age', 'Area Income']])
X = sm.add_constant(X)
```

In [211…
```
X.head(5)
```

Out[211…

	const	Age	Area Income
0	1.0	35	61833.90
1	1.0	31	68441.85
2	1.0	26	59785.94
3	1.0	29	54806.18
4	1.0	35	73889.99

In [212…
```
#Isolating the target variables
y = pd.DataFrame(getData[['Clicked on Ad']])
```

In [213…
```
y
```

Out[213…

	Clicked on Ad
0	0
1	0
2	0
3	0
4	0
...	...
995	1
996	1
997	1
998	0
999	1

1000 rows × 1 columns

In [214…
```
logreg01 = sm.Logit(y, X).fit()
```

```
Optimization terminated successfully.
         Current function value: 0.440002
         Iterations 7
```

In [216…
```
#Summary of the regression
logreg01.summary2()
```

Out[216…

Model:	Logit	Pseudo R-squared:	0.365
Dependent Variable:	Clicked on Ad	AIC:	886.0035
Date:	2021-10-25 14:52	BIC:	900.7268
No. Observations:	1000	Log-Likelihood:	-440.00
Df Model:	2	LL-Null:	-693.15
Df Residuals:	997	LLR p-value:	1.1490e-110
Converged:	1.0000	Scale:	1.0000
No. Iterations:	7.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
const	0.0916	0.5399	0.1697	0.8653	-0.9666	1.1498
Age	0.1626	0.0126	12.9008	0.0000	0.1379	0.1874
Area Income	-0.0001	0.0000	-12.5857	0.0000	-0.0001	-0.0001

Descriptive form of the final logistic regression

In [202…
```
#phat_Clicked_on_Ad = (exp(-0.0091640+0.1626(age)-0.0001(Area Income))/1+exp(-0.0916+0.1626(age)-0.0001(Area Incom
```

Interpretation of the coefficient is found on the final report

In [229…
```
math.exp(0.1626)
```

Out[229…
```
1.1765659696162973
```

In [230…
```
math.exp(-0.0001)
```

Out[230…
```
0.9999000049998333
```

Predict the possibility of increasing the number of ad clicks if the a person if 5 years from now

In [203…
```
math.exp(0.1705*5)
```

Out[203…
```
2.3455032865488583
```

Predict the possibility of increasing the number of ad clicks if the area income is increased by 5,000

In [204…
```
math.exp(-0.0001*5000)
```

Out[204…
```
0.6065306597126334
```

Interpretation is found on the final report

In [220…
```
#Obtaing the predicted values
ypred = logreg01.predict(X)
```

In [221…
```
#Getting the actual values
yture = getData['Clicked on Ad']
```

Building Poisson regression model to predict how a person's age influences their decision to click on the Ad and if the Area of income plays a role.

In [222…
```
Possion_Reg = sm.GLM(y, X, family = sm.families.Poisson()).fit()
```

In [223…
```
Possion_Reg.summary()
```

Out[223…

Generalized Linear Model Regression Results

Dep. Variable:	Clicked on Ad	**No. Observations:**	1000
Model:	GLM	**Df Residuals:**	997
Model Family:	Poisson	**Df Model:**	2
Link Function:	log	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-758.24
Date:	Tue, 26 Oct 2021	**Deviance:**	516.47
Time:	07:32:37	**Pearson chi2:**	452.
No. Iterations:	5		
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.9793	0.280	-3.496	0.000	-1.528	-0.430
Age	0.0415	0.005	8.556	0.000	0.032	0.051
Area Income	-2.495e-05	3.1e-06	-8.050	0.000	-3.1e-05	-1.89e-05

In [224…
```
#The descriptive form for the final Poisson regression model.
#Clicked_on_Ad = exp(-0.9793+0.0415(Age)-2.495e-05(Area Income))
```

In [225…
```
#Obtaining the predicted values from the model.
ypred_poisson = Possion_Reg.predict(X)
```

In [226…
```
ypred_poisson
```

Out[226…
```
0      0.343718
1      0.246843
2      0.246883
3      0.319219
4      0.254427

995    0.220034
996    0.446943
997    1.084752
998    0.290584
999    0.524932
Length: 1000, dtype: float64
```

In [227…
```
#Getting the actual values
ytrue = getData['Area Income']
```