Analysing The Data from the NLSS 2018-2019 For South-West Nigeria

Komolafe Elisha Ayobami

EEG/2015/061

November 26, 2021

Outline

- Introduction
- NLSS Data
- Data Preprocessing
- Methodology
- Results
- Conclusion

Introduction

"Health is a state of complete physical, mental and social well-being and not merely the absence of disease or infirmity." [who, 1946]

The most significant Principles of the W.H.O. are:

- The enjoyment of the highest attainable standard of health is one of the fundamental rights of every human being without distinction of race, religion, political belief, economic or social condition.
- The health of all peoples is fundamental to the attainment of peace and security and is dependent on the fullest co-operation of individuals and States.
- Governments have a responsibility for the health of their peoples which can be fulfilled only by the provision of adequate health and social measures.

Introduction to the NLSS Dataset

The NLSS is a nationwide intervention and collaborative effort of multiple agencies carried out between 2018 and 2019 to provide:

- Provide critical information for production of a wide range of socio-economic and demographic indications for benchmarking and monitoring of SDGs.
- Monitor progress in the populations welfare.
- Provide statistical evidence and measure the impact on households from current and future Government policies.

Highlights From the NLSS Dataset

- Malaria is the most reported health illness reported by Nigerians.
- Most Nigerians visit a chemist for medical treatment.
- it takes longer to reach and receive treatment at a Hospital compared to a chemist.
- Most Nigerians suffering from Minor illnesses don't seek Medical treatment.
- Very few Nigerians visit an Hospital in a year.

NLSS Data

Table: Data from the NLSS Report.

Strata	Hospital		Clinic		Chemist	
	Male	Female	Male	Female	Male	Female
NIGERIA	130.7	127.7	76.7	94.9	50.6	49.2
Urban	107.8	110.3	60	101	44.6	40.7
Rural	145.1	139.8	81.5	93.2	53	52.3
Ekiti	126.3	103.7	160.6	71.1	22.8	27.1
Lagos	63	62.2	34.5	41.2	18.4	19.2

Table: Portion of the Displayed Input without Preprocessing

index	s03q01	s03q02	s03q03	...	s03q25
1	1. YES	NaN	2. NO	...	1. No, no difficulty
2	1. YES	NaN	2. NO	...	1. No, no difficulty
3	1. YES	NaN	1. YES	...	1. No, no difficulty
4	2. NO	2	2. NO	...	1. No, no difficulty
5	2. NO	2	2. NO	...	1. No, no difficulty

Figure: Data importing

```
#%%

#data preprocessing, cleaning and a little sorting
health_data_raw=pd.read_stata('Data Sets_Other/sect3_health.dta'
    )
health_data=pd.read_stata('Data Sets_Other/sect3_health.dta',
    convert_categoricals=False,preserve_dtypes=False,
    convert_missing=False)
health_data=health_data.fillna(0)
health_data
#health_data.describe()
#health_data_raw
#health_data_raw.describe()
#sea.pairplot(health_data)

#%%
```

Figure: Data selection

```
#%%

#data for the south west for past 30 days
southwest_total_data=health_data.loc[health_data['zone']==6,['
    state','sector','s03q03','s03q04_1','s03q04_2',
        's03q05','s03q06_1','s03q06_2','s03q07a','s03q08','s03q09
            ', 's03q10_1',
        's03q10_2', 's03q11__1', 's03q11__2', 's03q11__3',
        's03q11__4', 's03q11__5', 's03q12', 's03q13',
        's03q14', 's03q15', 's03q16a', 's03q16b', 's03q17', '
            s03q18', 's03q18b']]
#state_zone=health_data.loc[health_data['zone']<=6,['state','
    zone']]
#state_zone=state_zone.loc[state_zone['state']==24,:]
#sea.pairplot(southwest_total_data)
southwest_total_data
```

Data Preprocessing

Figure: mapping variables

```
#%%

#changing the values of 1 and 2 to yes and no respectively.
val2str=southwest_total_data.loc[:,'s03q03']

for i in southwest_total_data.index:
    if val2str[i]==1.0:
        southwest_total_data['s03q03'][i]='YES'
    else:
        southwest_total_data['s03q03'][i]='NO'



southwest_total_data.head(30)
```

Figure: Final Data output

Table 2.3: Table showing the Data after Mapping using 2.15

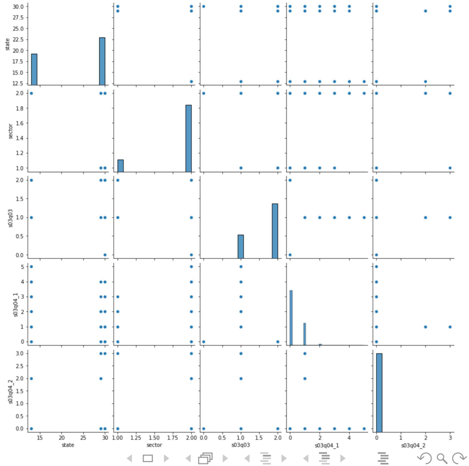
state	sector	s03q03	s03q04_1
2109	13	2	NO
2110	13	2	NO
2111	13	2	NO
2112	13	2	NO
2113	13	2	YES
2114	13	2	YES
2115	13	2	YES
2116	13	2	NO
2117	13	2	NO
2118	13	2	YES

Data Visualization

Figure: Visualization of Data chunks

```
#%%

#data plotting
sw_plot_1=southwest_total_data.loc[:,['state','sector','s03q03',
    's03q04_1','s03q04_2']]
sea.pairplot(sw_plot_1)


#%%

#data plotting
sw_plot_2=southwest_total_data.loc[:,['s03q05','s03q06_1','
    s03q06_2','s03q07a','s03q08']]
sea.pairplot(sw_plot_2)

#%%

#data plotting
sw_plot_3=southwest_total_data.loc[:,['s03q09', 's03q10_1','
    s03q10_2', 's03q11__1', 's03q11__2']]
sea.pairplot(sw_plot_3)
```

Figure: Plot of the first 5 variables



VIF Analysis

Figure: Code to Calculate VIF.

```
#%%

#test for colinearity if needed (vif)
vifdatax=health_data.loc[health_data['zone']==6,['zone','state',
    'sector','s03q04_1','s03q04_2',
        's03q05','s03q06_1','s03q06_2','s03q07a','s03q08','s03q09
           ', 's03q10_1',
        's03q10_2', 's03q11__1', 's03q11__2', 's03q11__3',
        's03q11__4', 's03q11__5', 's03q12', 's03q13',
        's03q14', 's03q15', 's03q16a', 's03q16b', 's03q17', '
           s03q18', 's03q18b']]


vif_data=pd.DataFrame()
vif_data['feature']=vifdatax.columns

vif_data['VIF']=[vif(vifdatax.values,i) for i in range(len(
    vifdatax.columns))]
print(vif_data)
```

Figure: First VIF Result.

	feature	VIF
0	zone	126.379359
1	state	1.283020
2	sector	1.155342
3	s03q04_1	2.411468
4	s03q04_2	1.071587
5	s03q05	15.748960
6	s03q06_1	1.710809
7	s03q06_2	1.260246
8	s03q07a	3.671035
9	s03q08	8.521257
10	s03q09	3.909443
11	s03q10_1	11.165224
12	s03q10_2	1.208677
13	s03q11__1	3.129934
14	s03q11__2	1.283692
15	s03q11__3	1.641351
16	s03q11__4	1.169566
17	s03q11__5	1.055591
18	s03q12	13.549558
19	s03q13	2.960139
20	s03q14	1.327161
21	s03q15	1.554728
22	s03q16a	3.966910
23	s03q16b	24.194978
24	s03q17	9.254953
25	s03q18	1.852098
26	s03q18b	2.136655

VIF Analysis 1

Figure: Final VIF Result.

	features	vif without s03q12
0	state	6.096657
1	sector	6.326565
2	s03q04_1	2.730057
3	s03q04_2	1.072848
4	s03q06_1	2.041564
5	s03q06_2	1.365661
6	s03q07a	4.516364
7	s03q08	10.664574
8	s03q09	4.298487
9	s03q10_1	4.224135
10	s03q10_2	1.245882
11	s03q11__1	2.679302
12	s03q11__2	1.272134
13	s03q11__3	1.463381
14	s03q11__4	1.166484
15	s03q11__5	1.034655
16	s03q13	2.183749
17	s03q14	1.378289
18	s03q15	1.585710
19	s03q16a	2.384502
20	s03q17	9.520753
21	s03q18	1.949016
22	s03q18b	2.535808

Conlusions

- Zone has high collinearity.
- By removing the Highest Co linear Variables, the data can be free of collinearity.

Methodology

Figure: Dividing the Data into Training and Test sets.

```
#%%

#split data using 80/20 method and train and test
southwest_train, southwest_test= split(southwest_total_data,
    test_size=0.2)
southwest_train
southwest_test

#%%
```

Specifications

- Only Data from South-West Nigerian States[a].
- Testing Data is 20% of the available Data.
- The Models to be Trained are: Logistic Regression, LDA, QDA.
- k chosen for k-fold validation is 7.

[a] Ekiti,Lagos,Ogun,Ondo,Osun and Oyo

Figure: Training Logistic Regression Model.

```
        's03q05','s03q06_1','s03q06_2','s03q07a','s03q08','s03q09
           ', 's03q10_1',
        's03q10_2', 's03q11__1', 's03q11__2', 's03q11__3',
        's03q11__4', 's03q11__5', 's03q12', 's03q13',
        's03q14', 's03q15', 's03q16a', 's03q16b', 's03q17', '
           s03q18', 's03q18b']]
swtrainy= southwest_train.loc[:, ['s03q03']]
visit_doc_model.fit(swtrainx, swtrainy)
print(visit_doc_model.coef_)
#visit_doc_model.n_features_in_
#visit_doc_model
```

Figure: Results from Testing.

```
              precision    recall  f1-score   support

NO                 0.98      0.98      0.98       288
YES                0.95      0.95      0.95       135

accuracy                               0.97       423
macro avg          0.96      0.96      0.96       423
weighted avg       0.97      0.97      0.97       423

array([[281,   7],
       [  7, 128]], dtype=int64)
```

Methodology. (LDA Model)

Figure: Training Logistic Regression Model.

```
#%%

#lda
visitdoc_ldamodel=LinearDiscriminantAnalysis()
visitdoc_ldamodel.fit(swtrainx, swtrainy)
visitdoc_ldamodel
visitdoc_ldamodel.coef_
```

Figure: Results from Testing.

```
              precision    recall  f1-score   support

NO              1.00      1.00      1.00       288
YES             1.00      1.00      1.00       135

accuracy                            1.00       423
macro avg       1.00      1.00      1.00       423
weighted avg    1.00      1.00      1.00       423

array([[288,   0],
       [  0, 135]], dtype=int64)
```

Methodology. (QDA Model)

Figure: Training Logistic Regression Model.

```
#%%

#qda
visitdoc_qdamodel=QuadraticDiscriminantAnalysis()
visitdoc_qdamodel.fit(swtrainx, swtrainy)
visitdoc_qdamodel
visitdoc_qdamodel.get_params()
```

Figure: Results from Testing.

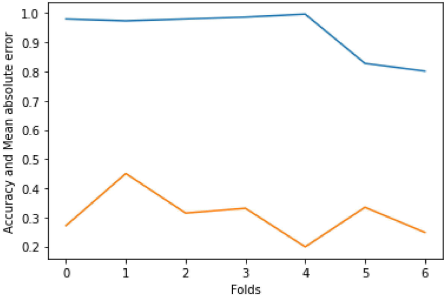
```
              precision    recall  f1-score   support

NO                 1.00      0.91      0.95       288
YES                0.83      1.00      0.91       135

accuracy                               0.94       423
macro avg          0.92      0.95      0.93       423
weighted avg       0.95      0.94      0.94       423

array([[261,  27],
       [  0, 135]], dtype=int64)
```

K-Fold Cross-Validation

Figure: Code for Cross-Validation

```
#%%

#cross validation with 7 folds for log regression
k=7
folds=KFold(n_splits=k)
accuracy=[]
metrep=[]

for train_index,test_index in folds.split(kcvx):
    #x_train=[],x_test=[],y_train=[],y_test=[]
    x_train,x_test= kcvx.iloc[train_index,:],kcvx.iloc[
        test_index,:]
    y_train,y_test= kcvy.iloc[train_index,:],kcvy.iloc[
        test_index,]

    #print(train_index,test_index)
    visit_doc_model.fit(x_train,y_train)
    y_pred=visit_doc_model.predict(x_test)
    metrep.append(metrics.classification_report(y_test,y_pred))
    accuracy.append(metrics.accuracy_score(y_test,y_pred))
    #mae.append(metrics.mean_absolute_error(y_test,y_pred))
```

Figure: Plot of Cross-Validation on Log. Regression Model



K-Fold Cross-Validation 1

Figure: Plot of Cross-Validation on LDA Model

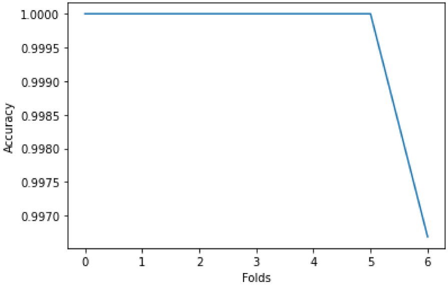
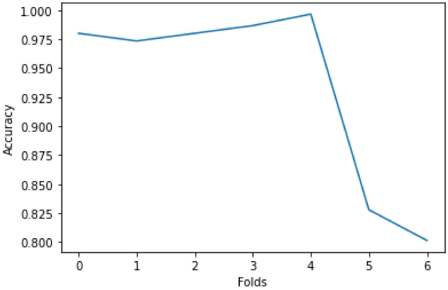


Figure: Plot of Cross-Validation on QDA Model



PCA Analysis

Figure: Code snippet for 80% variance

```
#%%

#pca method on single 80/20 data used originally using 85%
   variance retained
pca=PCA(0.85)
pca.fit(swtrainx)
#pca.n_components_
pcatrainx=pca.transform(swtrainx)
pcatestx=pca.transform(swtestx)
pcatrainx
#pcatestx
pca.explained_variance_ratio_
```

Figure: Result of all the PCAtests.

```
85%-array([0.96016506])
90%-array([0.96016506])
95%-array([0.96016506])
97%-array([0.96016506, 0.02108479]
99%-array([0.96016506, 0.02108479, 0.01870875])
5_components-array([9.60165059e-01, 2.10847903e-02, 1.87087459e
   -02, 2.55160744e-05,
5.81451411e-06])
10_components-array([9.60165059e-01, 2.10847903e-02, 1.87087459e
   -02, 2.55160744e-05,
5.81451411e-06, 3.73393949e-06, 2.35224596e-06, 2.14133718e-06,
6.24471077e-07, 5.63498855e-07])
```

Results

Conclusion from 7-fold Cross-Validation

- LDA shows the Highest accuracy across all the folds.
- Logistic Regression shows high accuracy but the range is larger than LDA.
- QDA shows the worst accuracy and has the highest accuracy range.

Conclusions from PCA

- 95% variance can only be represented with 1 variable.
- 97% variance is represented by 2 variables.
- 99% variance is represented by 3 variables.
- 3 variables are the minimum that can totally represent the system.

Conclusions

- A LDA Model fits the Data the best, with Logistic Regression second.
- the 3 most influential variables can represent the entire model, using dimension reduction.
- The Data has a Large amount of ‘No’ variables, which can sometimes lead to the models not fitting

Thank You for Listening.

(1946).
Constitution of the World Health Organization.