ASSOCIATION BETWEEN DEPRESSION AND POVERTY

DANIEL LAGOS

ANA 500

OCTOBER 23, 2022

HTTPS://GITHUB.COM/DANLAGOS/ANA500-WEEK-4

PROBLEM STATEMENT

- Background: Using California Health Interview Survey (CHIS) 2020 to search for and quantify the association between depression and poverty levels after controlling for general health, and current smoking habits in adults in the state of California.
- Objective
  - How does poverty levels affect depression in adults?
  - What are the effects of general health and smoking habits, and do they interact with poverty levels?
  - Cam a machine learning model predict depression based on poverty levels after accounting for general health, and smoking habits?

HYPOTHESIS FORMULATION

- H_0: There is no statistically significant association between depression and poverty levels after controlling for general health, and current smoking habits
- H_A: There is a statistically significant association between depression and poverty levels after controlling for general health, and current smoking habits

TOP-DOWN PROGRAM DESIGN

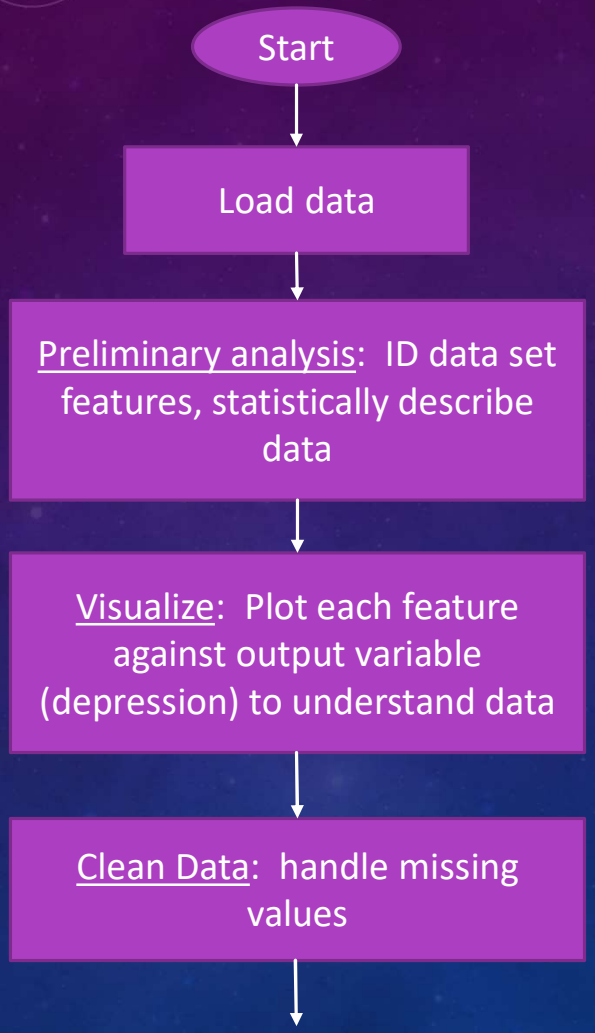
- Overall Task: Develop Python program that takes CHIS 2020 as input (poverty, general health, smoking habits) and outputs predicted effect on depression levels.
  - VAR AJ32: captures results of survey question “Feeling depressed in the past 30 days?
  - VAR AJ32 is categorical, numeric. Coding provided in visualization section
- Steps to take
  - Acquire: identify data set, retrieve data, query data. CHIS has already been selected. Currently in SAS XPT format.
  - Prepare: analyze codebook, explore and process data using python and discovery from codebook.
  - Analyze data: select analytical technique, build model
  - Report: communicate results – not inclusive to code development
  - Act: apply results, connect results with problem statement – not inclusive to code development

HIERARCHY CHART

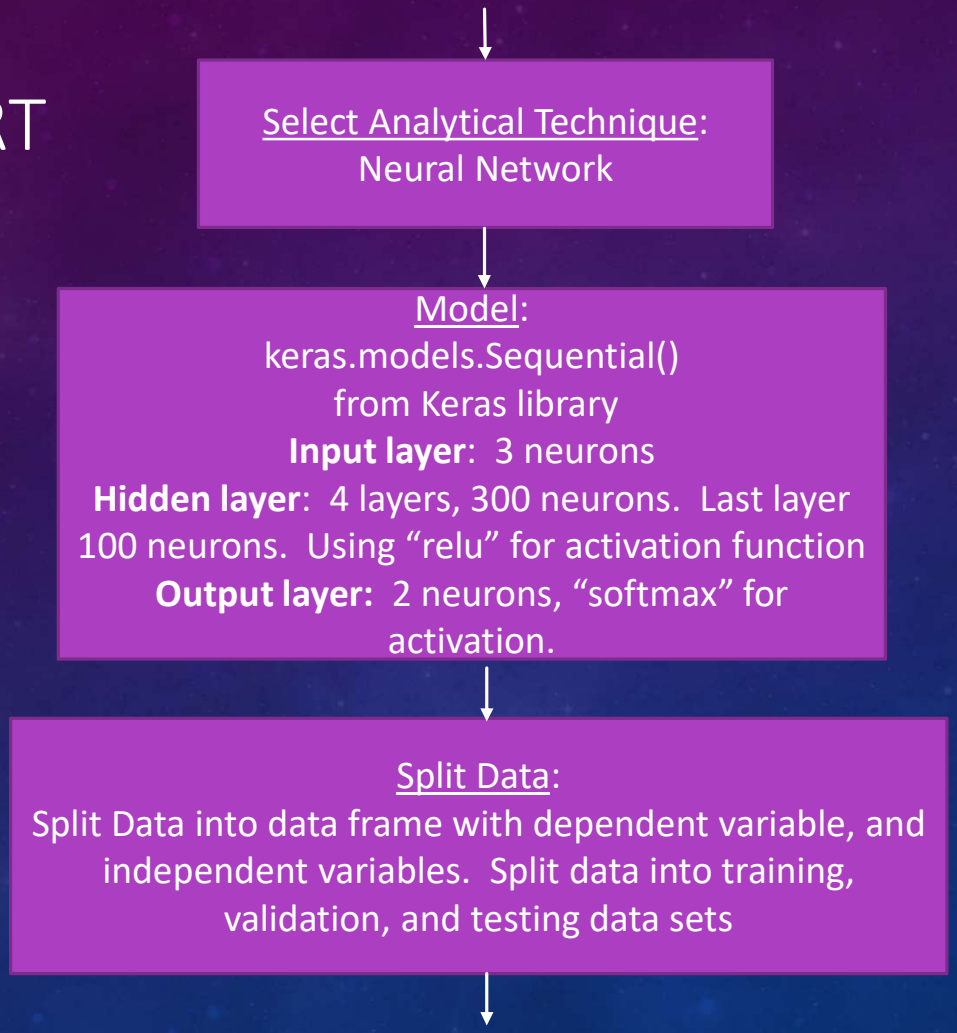
- DEPRESSION = CHIS2020DATASET(POVERTY, GENERAL HEALTH, SMOKING)

- Main Program(output depression, input CHIS dataset with variables for poverty levels, general health, and current smoking habits)

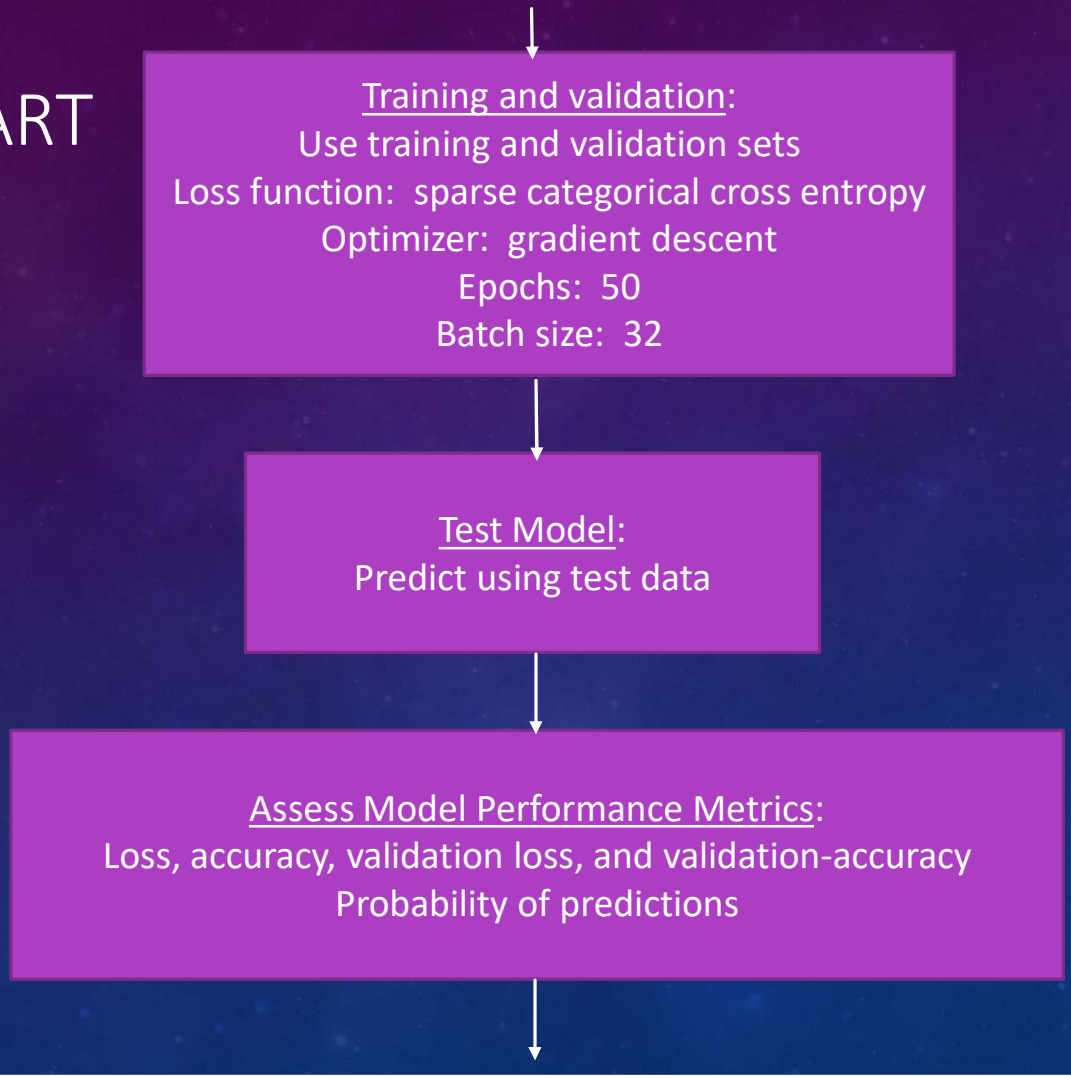
FLOW CHART


Start
Load data
Preliminary analysis: ID data set features, statistically describe data
Visualize: Plot each feature against output variable (depression) to understand data
Clean Data: handle missing values

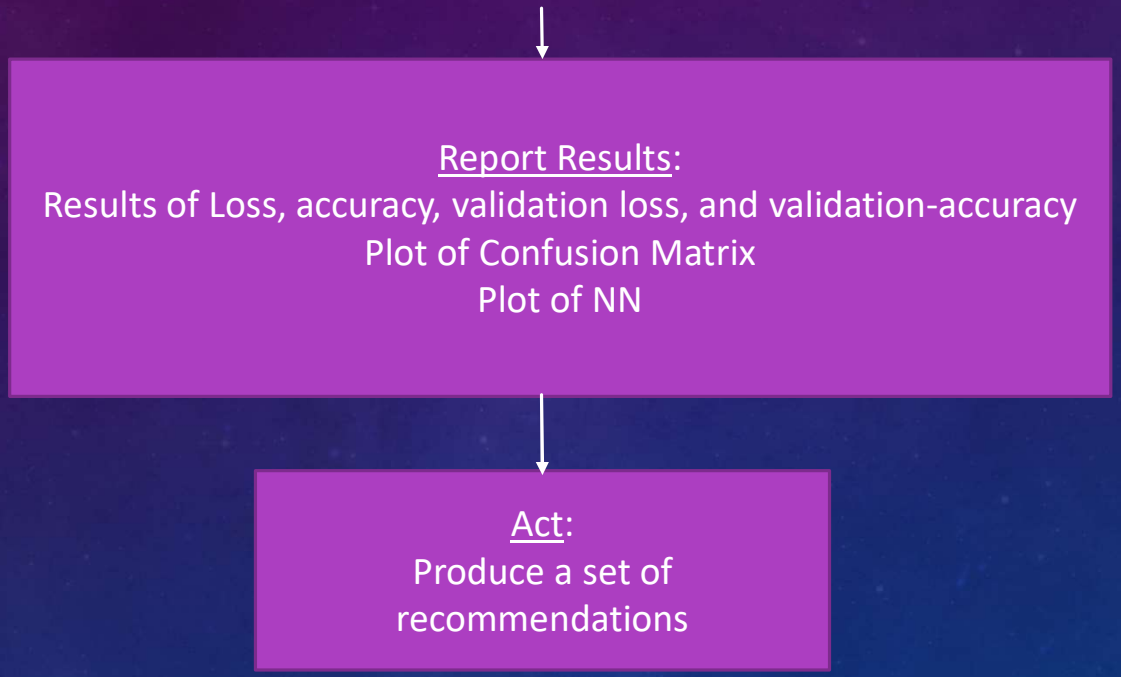
FLOW CHART


Select Analytical Technique:
Neural Network
Model:
keras.models.Sequential()
from Keras library
Input layer: 3 neurons
Hidden layer: 4 layers, 300 neurons. Last layer 100 neurons. Using "relu" for activation function
Output layer: 2 neurons, "softmax" for activation.
Split Data:
Split Data into data frame with dependent variable, and independent variables. Split data into training, validation, and testing data sets

FLOW CHART


Training and validation:
Use training and validation sets
Loss function: sparse categorical cross entropy
Optimizer: gradient descent
Epochs: 50
Batch size: 32
Test Model:
Predict using test data
Assess Model Performance Metrics:
Loss, accuracy, validation loss, and validation-accuracy
Probability of predictions

FLOW CHART


Report Results:
Results of Loss, accuracy, validation loss, and validation-accuracy
Plot of Confusion Matrix
Plot of NN
Act:
Produce a set of recommendations

PSEUDOCODE

- # Start
- # STEP 1 Acquire data: from CHIS website
- # Import libraries: pandas, NumPy, seaborn, matplotlib
- # Read dataset: import data. Must read in SAS XPT file into Python
- #STEP 2: Prepare – conduct analysis of codebook – identify appropriate variables, and determine structure
- # read data: data.head(), data.info()
- # Format data: create new data set with appropriate variables–
  - SMOKING, AB1 (GEN_HEALTH), POVLL (POVERTY), AJ32 (DEPRESSION)
  - Esnure DEPRESSION is binary.
  - Coding: 1 = Depressed, 2 = not-depressed

PSEUDOCODE

- # Remove inappropriate values – AJ32(Variable for depression) contains value of -2 (means proxy skipped). Not applicable, remove.
- # Visualization: make histograms, scatter plots and bar charts, use as appropriate.
- # STEP 3: Analyze
- # Create two new data sets, one that contains Var DEPRESSION (Y data frame), the other that contains POVERTY, GEN_HEALTH, SMOKING (X data frame)
- # Further separate into testing and training data frames (Y_train, Y_test, X_train, X_test)

PSEUDOCODE

- Train classifier
  - Max leaf node = 10
  - Max depth = 5
  - Criterion = entropy, classifiers based on entropy are better with nonbinary categorical variables as compared to gini index.
- Predict on test data
- Assess
  - Accuracy score
  - Precision score
  - Recall Score
  - Plot of decision tree
  - Plot of feature importance's – to determine how algorithm weights the importance of each variable.

ACQUIRE

- Data set was acquired via CHIS website (https://healthpolicy.ucla.edu/chis/data/pages/getchisdata.aspx). This is a well respected and well cited survey repository that conducts surveys to delve into California's public health issues.
- CHIS website proved extensive codebook to assist in analysis and will be relied upon for the data preparation of this project.

PREPARE

- Codebook identifies AJ32, variable for depression with value of -2, meaning proxy missing.
- All other variables are categorical numeric

PREPARE

```
In [3]: df.head()
```

Out[3]:

	AA5C	AA5G	AH37	AH44	AB1	AB17	AB40	AB41	AB18	AB43	...	RAKEDW71	RAKEDW72	RAKEDW73	RAKEDW74	RAKEDW75	RAKEDW76	RAK
0	-1.0	-1.0	-1.0	-1.0	3.0	2.0	-1.0	-1.0	-1.0	-1.0	...	1605.912814	1577.479222	1578.109407	1566.686492	1657.063161	1598.357033	1610
1	-1.0	-1.0	-1.0	1.0	2.0	2.0	-1.0	-1.0	-1.0	-1.0	...	1558.197812	1513.430922	1537.924152	1537.402328	1529.306613	1544.735340	1531
2	-1.0	-1.0	-1.0	-1.0	2.0	2.0	-1.0	-1.0	-1.0	-1.0	...	1211.750536	1212.772970	1202.982303	1225.293304	1248.000638	1216.821869	1233
3	-1.0	-1.0	-1.0	-1.0	3.0	1.0	1.0	2.0	2.0	2.0	...	590.153646	604.817107	596.879484	594.582363	587.412696	590.762356	589
4	-1.0	1.0	-1.0	1.0	4.0	2.0	-1.0	-1.0	-1.0	-1.0	...	194.059102	188.468808	201.853958	195.711101	200.749310	191.936914	206

5 rows × 604 columns

PREPARE

- Confirming that no other variables contain unusable values. AJ32 with values of -2

```
In [4]: df.AJ32.describe() # VAR DEPRESSION

Out[4]: count    21949.000000
        mean         4.631783
        std          0.734293
        min         -2.000000
        25%          5.000000
        50%          5.000000
        75%          5.000000
        max          5.000000
        Name: AJ32, dtype: float64
```

```
In [12]: df.SMOKING.describe() # VAR SMOKING HABITS

Out[12]: count    21944.000000
         mean         2.638307
         std          0.582533
         min          1.000000
         25%          2.000000
         50%          3.000000
         75%          3.000000
         max          3.000000
         Name: SMOKING, dtype: float64
```

PREPARE

- Confirming that no other variables contain unusable values

```
In [13]: df.AB1.describe() # VAR GENERAL HEALTH

Out[13]: count    21944.000000
         mean         2.341232
         std          0.981413
         min          1.000000
         25%          2.000000
         50%          2.000000
         75%          3.000000
         max          5.000000
         Name: AB1, dtype: float64
```

```
In [14]: df.POVLL.describe() # POVERTY LEVELS AS LEVELS OF FEDERAL POVERTY LEVELS

Out[14]: count    21944.000000
         mean         3.423943
         std          0.973866
         min          1.000000
         25%          3.000000
         50%          4.000000
         75%          4.000000
         max          4.000000
         Name: POVLL, dtype: float64
```

PREPARE

- Confirming lack of missing values

```
In [15]: df.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21944 entries, 0 to 21948
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   SMOKING  21944 non-null  float64
 1   AB1      21944 non-null  float64
 2   POVLL    21944 non-null  float64
 3   AJ32     21944 non-null  float64
dtypes: float64(4)
memory usage: 857.2 KB
```

PREPARE

- Clean data

```
In [9]: df = df.loc[:, ['SMOKING', 'AB1', 'POVLL', 'AJ32']]
        df = df[(df['AJ32'] > -1)]

        smokingCurrent = df.SMOKING # CURRENT SMOKING HABITS - CAT
        generalHealth = df.AB1 # GENERAL HEALTH CONDITION - CAT
        povertyFPL = df.POVLL # FPL - CAT
        depression = df.AJ32 # feeling depressed in last 30 days - CAT - DEP VAR

In [10]: df.head()

Out[10]:
```

	SMOKING	AB1	POVLL	AJ32
0	3.0	3.0	3.0	3.0
1	2.0	2.0	3.0	5.0
2	3.0	2.0	4.0	5.0
3	2.0	3.0	3.0	5.0
4	2.0	4.0	2.0	5.0

PREPARE

- Original Data set: Confirming data set is clean: all values over 0, n= 21,944, devoid of variables not in use.

```
In [11]: df.describe()
```

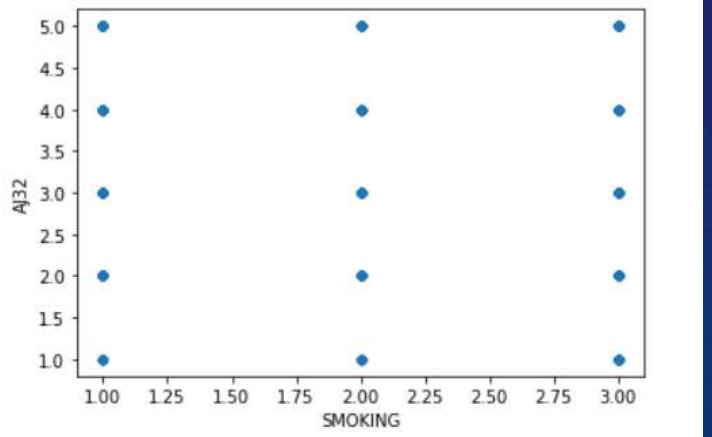
Out[11]:

	SMOKING	AB1	POVLL	AJ32
count	21944.000000	21944.000000	21944.000000	21944.000000
mean	2.638307	2.341232	3.423943	4.633294
std	0.582533	0.981413	0.973866	0.727520
min	1.000000	1.000000	1.000000	1.000000
25%	2.000000	2.000000	3.000000	5.000000
50%	3.000000	2.000000	4.000000	5.000000
75%	3.000000	3.000000	4.000000	5.000000
max	3.000000	5.000000	4.000000	5.000000

PREPARE - VISUALIZATION

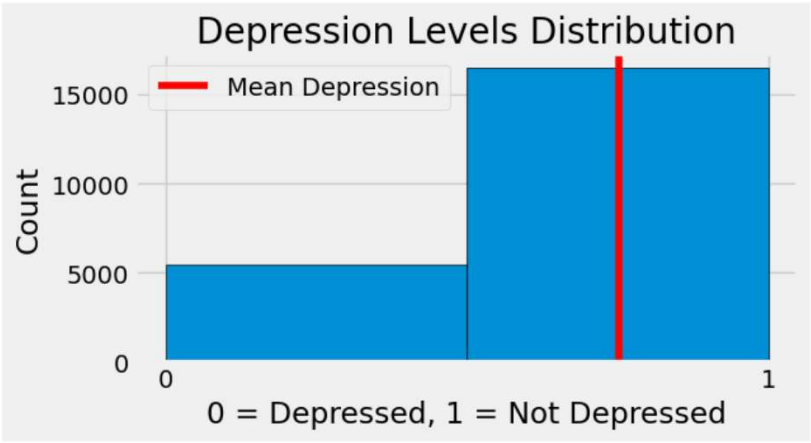
- After conducting data.hist(), data.scatter() and data.plot.box() it has been determined that scatter plots are inappropriate: example of output of scatter plot for smoking vs depression levels, below. All other scatter plots provide similar results, and not insightful, therefore not used.

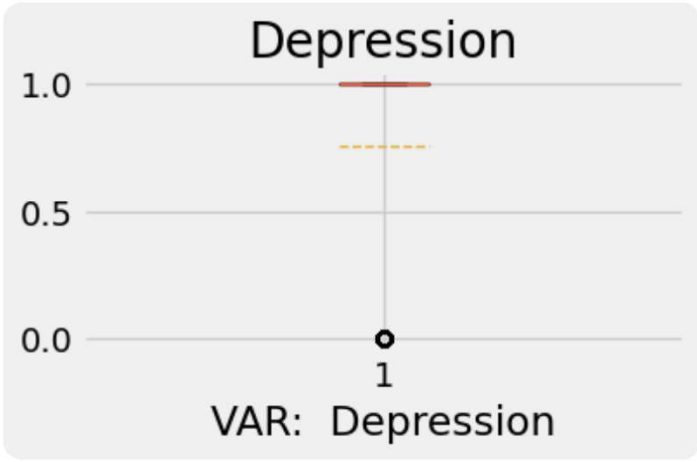
```
In [11]: df.plot.scatter(x='SMOKING', y='AJ32')
Out[11]: <AxesSubplot:xlabel='SMOKING', ylabel='AJ32'>
```



PREPARE - VISUALIZATION

- Histogram of dependent variable DEPRESSION. Majority of observation “not-depressed” compared to “depressed”.
- AJ32 question: Feeling depressed in the past 30 days?
- Changed label of AJ32 to DEPRESSION.
- Coding:
  - 0 = Depressed
  - 1 = Not-Depressed



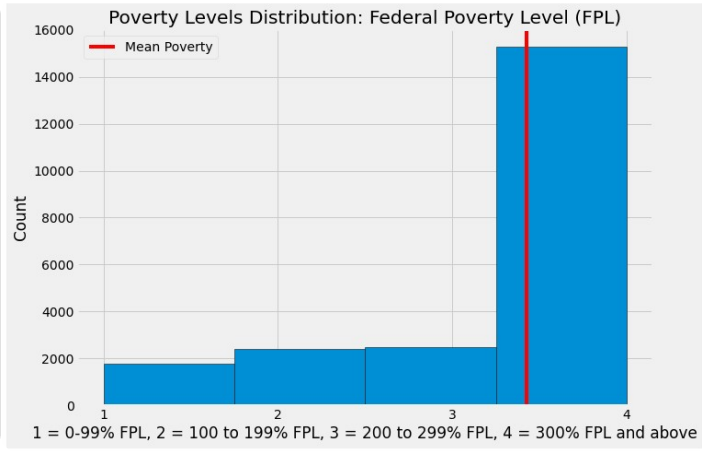


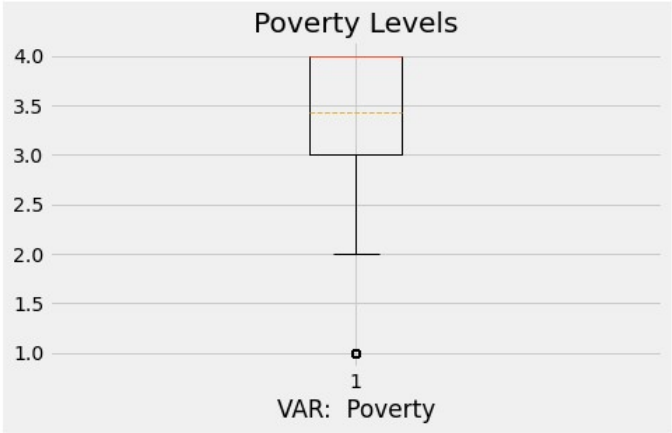
PREPARE - VISUALIZATION

- Observations - DEPRESSION: Very difficult to determine much from the box plot. This may be because in the histogram it appears as if there are relatively few people that are depressed compared to those that are not depressed.
- Histogram shows left skew, meaning the distribution is not normal. Need to keep this in mind if using an algorithm that assumes normal data.

PREPARE - VISUALIZATION

- Histogram & Boxplot of POVERTY,. Left Skew.
- Federal Poverty level (FPL)
- Coding
  - 1 = 0-99% FPL
  - 2 = 100 to 199% FPL
  - 3 = 200 to 299% FPL
  - 4 = 300% FPL and above



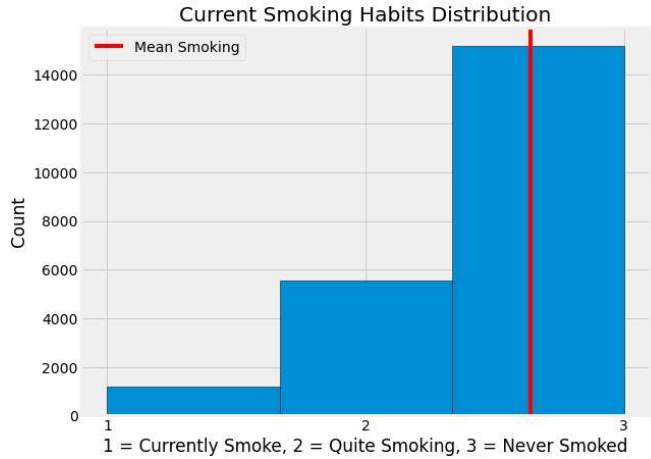


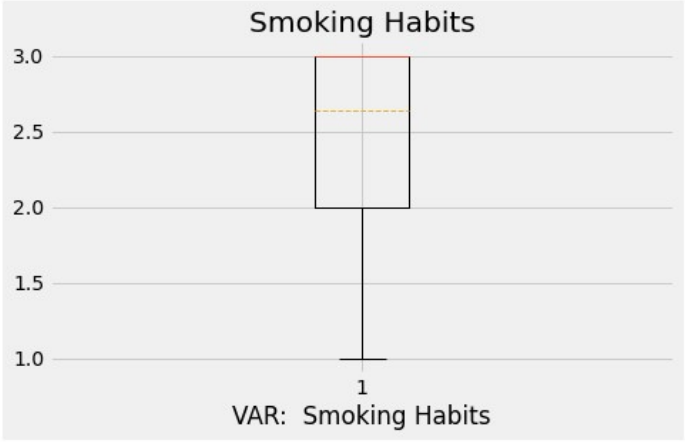
PREPARE – VISUALIZATION

- Observations – POVERTY: Distribution is not normal. Merely pointing out if an algorithm is used that assumes normal distribution.
- Histogram and boxplot confirms skew, confirming lack of normality in distribution
- Income is generally ordinal, because poverty levels is based on Federal Poverty Levels, which is a measure of income, this distribution may have an ordinal component. This may determine the type of algorithm, or may require specific testing.

PREPARE - VISUALIZATION

- Histogram & Boxplot of SMOKING. Left Skew.
- Current smoking habits
- Coding
  - 1 = currently smokes
  - 2 = quit smoking
  - 3 = never smoked regularly



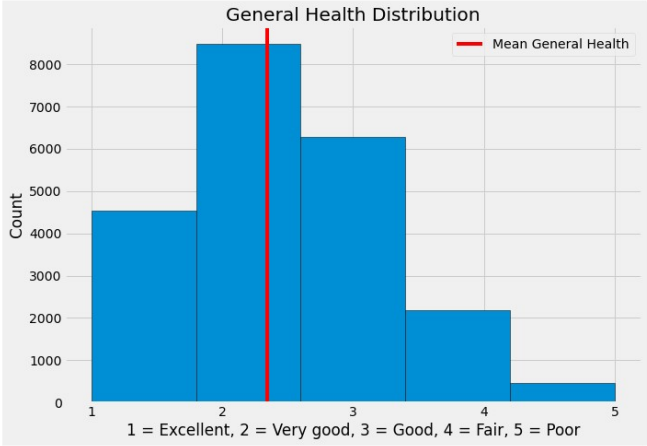


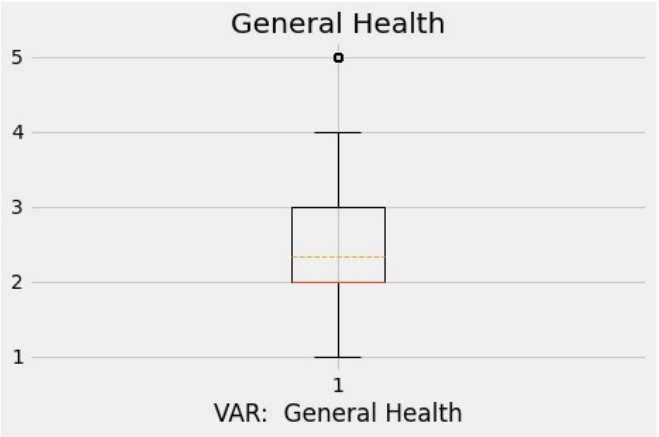
PREPARE – VISUALIZATION

- Observations – SMOKING: As before, histogram and box plot shows left skew. This is not as important as this is not out exposure variable.

PREPARE - VISUALIZATION

- Histogram & Boxplot GENERAL HEALTH, either right skew or might be normal. Difficult to determine.
- General Health Condition
- Coding
  - 1 = excellent
  - 2 = very good
  - 3 = good
  - 4 = fair
  - 5 = poor





PREPARE – VISUALIZATION

- Observations – GENERAL HEALTH: Boxplot and Histogram show that distribution may be normal.

PREPARE – VISUALIZATION

- General Observations based on visualizations We are not too concerned about possible lack of normal distributions to the data. We only worry about this if using an algorithm that requires normal distribution. Right now, the possible lack of normality is merely noted. We only need to act upon this if the algorithm used calls for data with normal distributions.
- POVERTY may be ordinal. Again, this is not a problem right now, but must be noted when determining the type of algorithm to use.
- N = 21,944: This should be sufficient to run an analysis.
- All our data is categorical. It may make sense to transform the data into binary forms to simplify the model and reduce computational power. However, there are only 4 variables in total, with 3 to 4 levels. The resulting model, and computational power required to train model may not be that complex. For now, no changes will be made to structure of data.

PREPARE

- Actions taken:
  - Before Analyze phase was conducted, variables were renamed so that the variable names were less ambiguous.
  - Var DEPRESSION was changed to a binary structure
    - If DEPRESSION = 5 then NEW_DEP = 1 (not depressed)
    - If DEPRESSION <= 5 then NEW_DEP = 0 (depressed)

	SMOKING	GEN_HEALTH	POVERTY	DEPRESSION	NEW_DEP
count	21944.000000	21944.000000	21944.000000	21944.000000	21944.000000
mean	2.638307	2.341232	3.423943	4.633294	0.751868
std	0.582533	0.981413	0.973866	0.727520	0.431938
min	1.000000	1.000000	1.000000	1.000000	0.000000
25%	2.000000	2.000000	3.000000	5.000000	1.000000
50%	3.000000	2.000000	4.000000	5.000000	1.000000
75%	3.000000	3.000000	4.000000	5.000000	1.000000
max	3.000000	5.000000	4.000000	5.000000	1.000000

ANALYZE

- This is a classification problem with labels, thus supervised type of ML. Neural network used, with the following hyperparameters for layers
  - Input layer: 3 neurons
  - Hidden layers: 4 layers, 300 neurons for all layers except for the last, contain 100 neurons. Activation function to be "relu" for all layers
  - Output layer: 2 neurons, one for each class within var NEW_DEP. Activation function to be "softmax"
- Other hyperparameters.
  - Loss: sparse categorical cross entropy
  - Optimizer: gradient descent
  - Epochs: 50
  - Batch size: 32

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 300)               1200

 dense_1 (Dense)             (None, 300)               90300

 dense_2 (Dense)             (None, 300)               90300

 dense_3 (Dense)             (None, 100)               30100

 dense_4 (Dense)             (None, 2)                 202

=================================================================
Total params: 212,102
Trainable params: 212,102
Non-trainable params: 0
_________________________________________________________________
```

```
X_New_train.describe()
```

	SMOKING	GEN_HEALTH	POVERTY
count	11191.000000	11191.000000	11191.000000
mean	2.640961	2.342597	3.428737
std	0.579620	0.979070	0.969458
min	1.000000	1.000000	1.000000
25%	2.000000	2.000000	3.000000
50%	3.000000	2.000000	4.000000
75%	3.000000	3.000000	4.000000
max	3.000000	5.000000	4.000000

```
X_val.describe()
```

	SMOKING	GEN_HEALTH	POVERTY
count	1975.000000	1975.000000	1975.000000
mean	2.626835	2.347848	3.450633
std	0.594631	1.003547	0.960448
min	1.000000	1.000000	1.000000
25%	2.000000	2.000000	3.000000
50%	3.000000	2.000000	4.000000
75%	3.000000	3.000000	4.000000
max	3.000000	5.000000	4.000000

```
X_New_test.describe()
```

	SMOKING	GEN_HEALTH	POVERTY
count	8778.000000	8778.000000	8778.000000
mean	2.637503	2.338004	3.411825
std	0.583520	0.979460	0.982370
min	1.000000	1.000000	1.000000
25%	2.000000	2.000000	3.000000
50%	3.000000	2.000000	4.000000
75%	3.000000	3.000000	4.000000
max	3.000000	5.000000	4.000000

ANALYZE

INDEPENDENT VARIABLES WERE SPLIT IN THE MANNER SHOWN ABOVE

```
: Y_New_train.describe()
```

:

	NEW_DEP
count	11191.000000
mean	0.751765
std	0.432008
min	0.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	1.000000

```
Y_val.describe()
```

	NEW_DEP
count	1975.000000
mean	0.761519
std	0.426263
min	0.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	1.000000

```
: Y_New_test.describe()
```

:

	NEW_DEP
count	8778.000000
mean	0.749829
std	0.433136
min	0.000000
25%	0.000000
50%	1.000000
75%	1.000000
max	1.000000

ANALYZE

DEPENDENT VARIABLE WAS SPLIT IN THE MANNER SHOWN ABOVE

ANALYZE

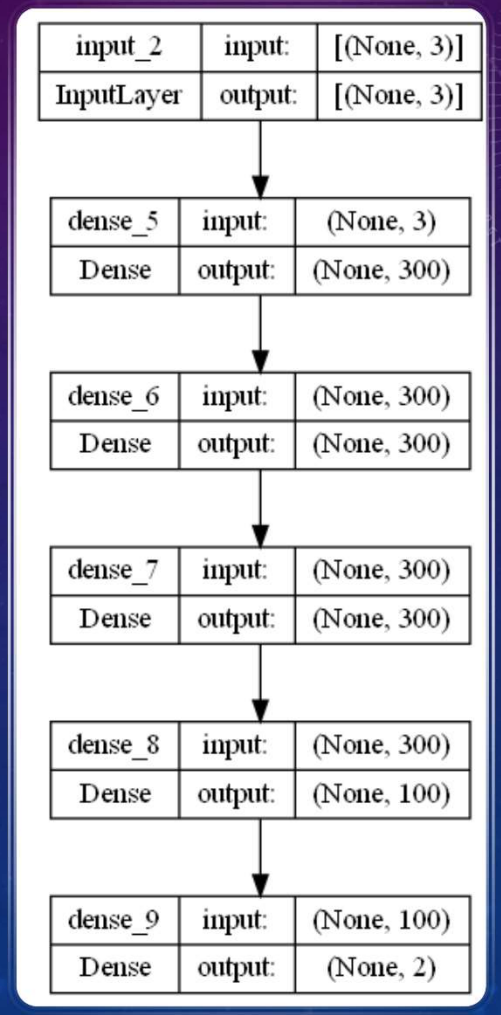
- After conducting training and validation of neural network, loss accuracy, validation loss, and validation accuracy metrics were produced.
- Plot of neural network is based on training and validation set.
- Evaluation was conducted on test set, producing loss and accuracy scores as well as confusion matrix.
- Note: originally this network was conducted with two hidden layers. In an attempt to optimize, 2 hidden layers were added, increasing the number of hidden layers to 4.. This presentation reports on results with additional layers, as there are no significant differences.
- Results in the next section.

REPORT

- Training scores:
  - Loss: 53.88%
  - Accuracy: 75.26%
  - Val loss: 52.75%
  - Val accuracy: 76.35%
- Testing scores:
  - Loss: 54.63%
  - Accuracy: 75.09%

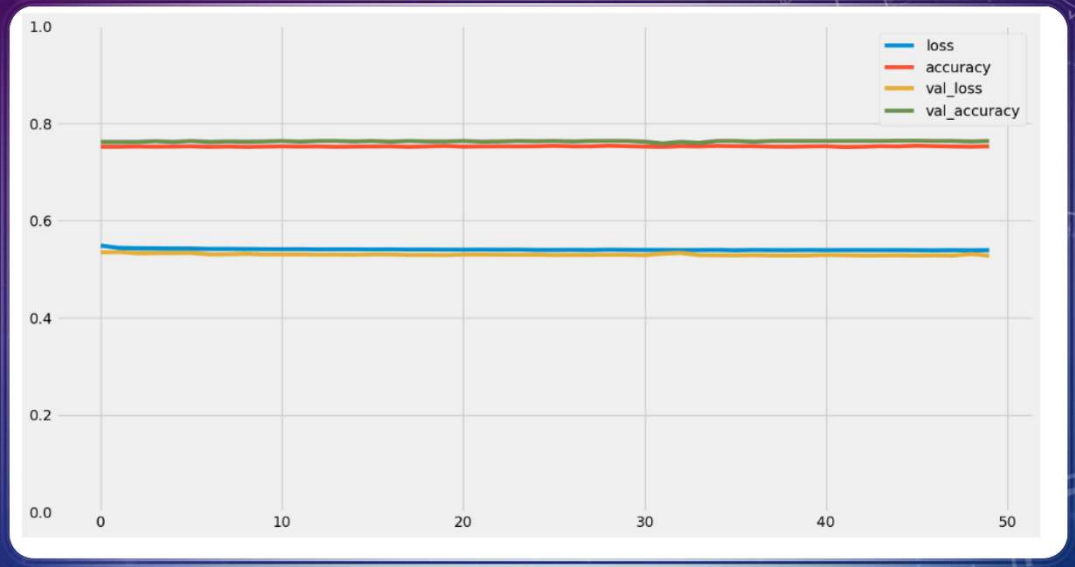
REPORT

- Plot of NN: There is very little complexity.
- As a reminder, this plot is based on training and validation sets.



REPORT

- Model performance chart
  - X-axis indicates epochs
  - Y-axis indicates score
- We have two sets of horizontal lines, indicating no gain after initial epoch.
- Network was unable to improve scores after 50 epochs



Evaluate model on test set

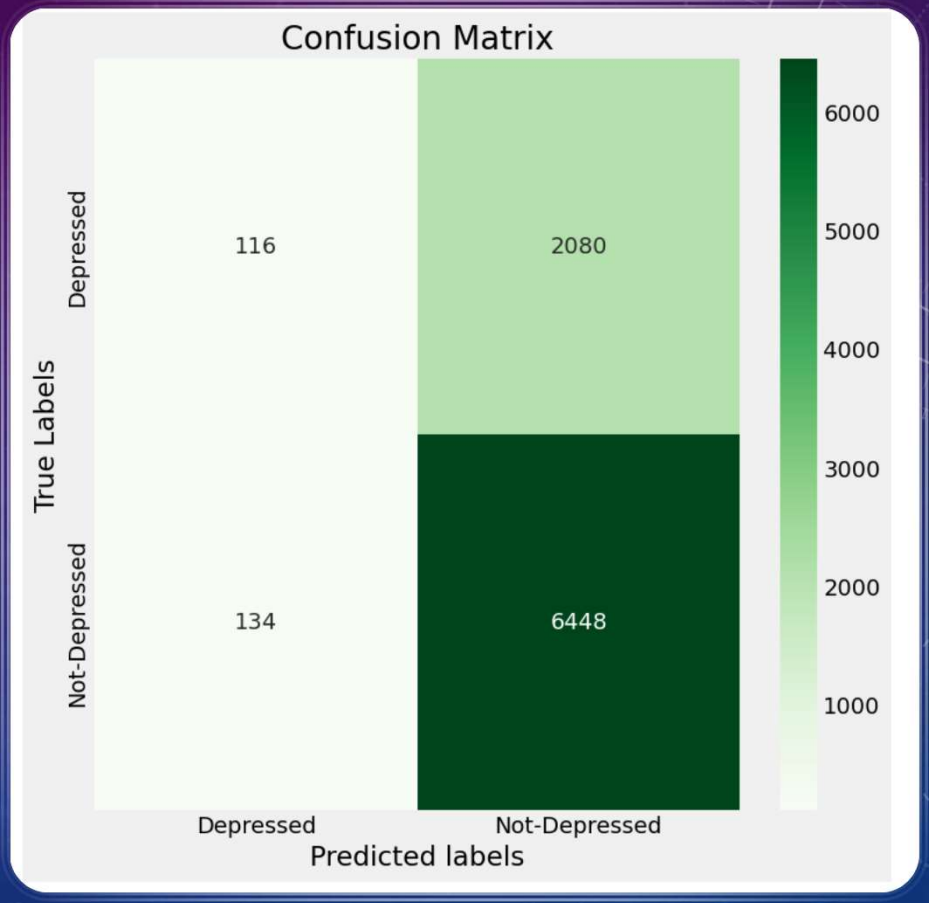
```
]: model.evaluate(X_New_test, Y_New_test)

   275/275 [==============================] - 1s 2ms/step - loss: 0.5463 - accuracy: 0.7509

]: [0.5463082790374756, 0.7508544325828552]
```

REPORT

LOSS: 54.63%, ACCURACY: 75.10%, RESULTS SIMILAR TO TRAINING SCORES

REPORT

- Confusion matrix plot: We would expect to see the upper left box to be a dark green if the model were correctly predicting “Depressed”
- We see that network is mis-categorizing “depressed” as “not-depressed”
- Network is correctly categorizing “not-depressed”



ACT

- Low accuracy scores of 75%, large misclassification of “depressed” as “not-depressed.” In the neural network plot we see that there is very little complexity. All these factors seem to indicate that the three variables are not sufficient in predicting depression.
- If we only consider these results, we must conclude that POVERTY, GEN_HEALTH, and SMOKING do not adequately predict DEPRESSION, forcing us to accept H_0.
- Hyperparameters were adjusted by adding more hidden layers, this did not produce significant changes. Neither the model nor the hyperparameters are the issue.
- It should be noted that a decision tree was previously conducted on the same data and variables. Results here are consistent with results of decision tree. This further confirms that the analytical technique is not the issue with the mis-classifications weakening the model.
- Also, there is a large body of published research that shows strong connections with depression and poverty. We cannot eliminate this variable, unless we ignore publish research.
- It must be further noted that the model is correctly predicting “not-depressed.” We can hypothesize that this is either an accident or that the model requires more complexity by adding more variables.

ACT

- Two courses of action are recommended:
  - Conduct a thorough test on neural networks hyper-parameters, to confirm that changing these will not provide much more improvement to model. Given that we have added to the complexity of the nueral network by adding two hidden layers with 300 neurons each, I do not expect to find much improvement by this course of action. This is merely recommended to be systematic in testing. Recommend using another NN algorithm to find optimal hyperparameters.
  - Conduct PCA of original dataset. Objective is to improve prediction power by finding more variables or eliminating current variables. Our main goal is to address misclassification of depressed" vs "not-depressed"