Heart Disease Prediction

Malachai Cravens, Marco Lopez
Kwadwo Asante, Yarely Vargas

CONTENTS

01

About our data

INTRODUCTION

According to the CDC, heart disease is one of the leading causes of death for most races in the US. About half of all Americans (47%) have at least 1 of 3 key risk factors for heart disease: high blood pressure, high cholesterol, and smoking. Driven by the silent approach heart disease has on its victims, we took it upon ourselves to search for relations within the indicators in our dataset.

Data Source

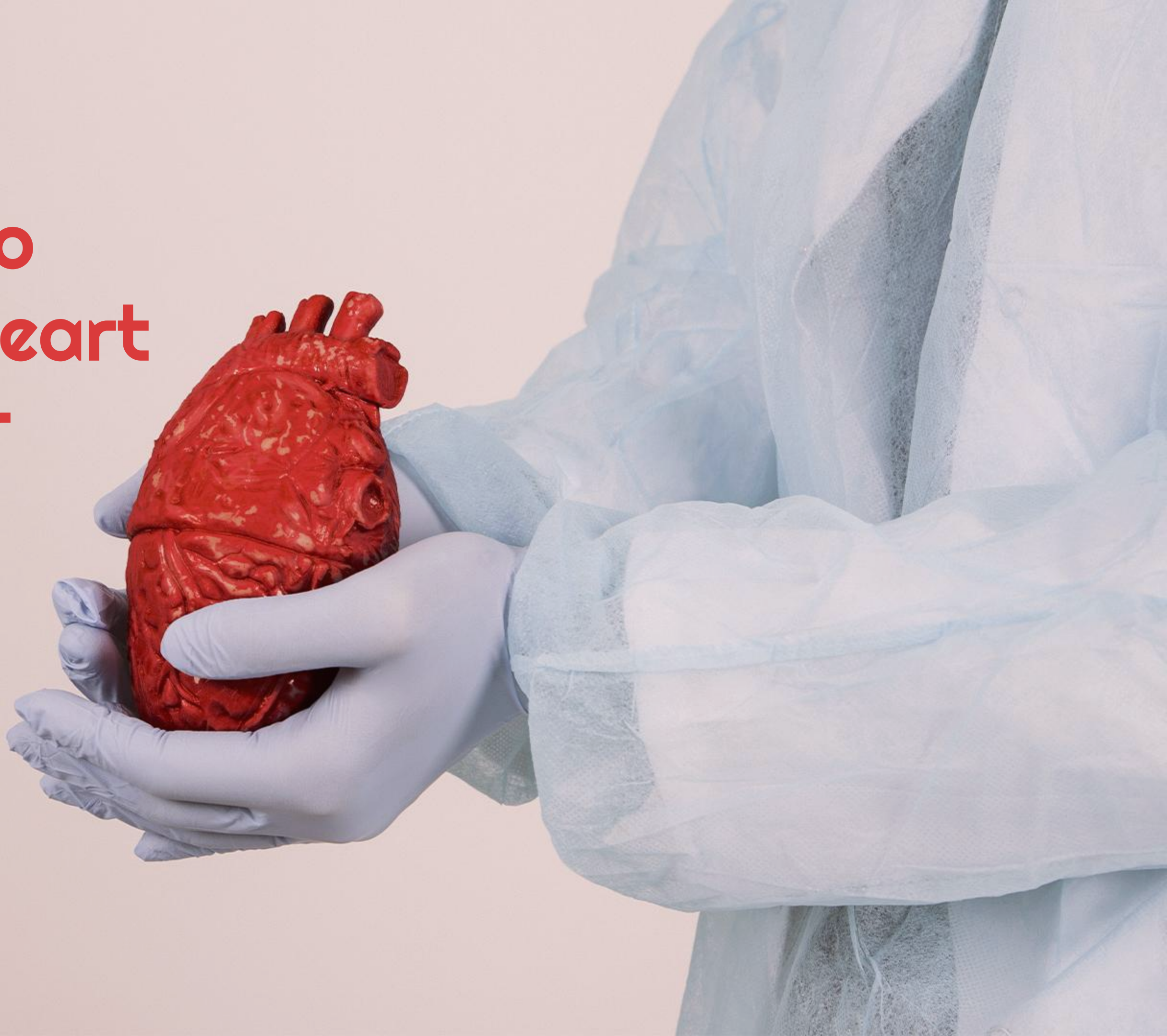
We used a Kaggle dataset.

Originally the data came from the BRFSS (Behavioral Risk Factor Surveillance System) which conducts annual telephone surveys to gather data on the health status of U.S. residents. The CDC uses this data and makes it available to the public.

The dataset provided, only has 17 columns out of the 279 columns available in the original survey.

Dataset is the most recent as of 02/15/2022 but has data from 2020

Heart Disease: respondents who have coronary heart disease (CHD) or myocardial infarction (MI)



Our data

Total Rows

There are 319,795 rows in our dataset

Imbalanced Data

9.3% of respondents had heart disease other 90.7% did not

Categorical Data

13 columns where categorical data (label encode)

Numerical

4 columns where numerical data (we scaled the data)

Sleep number

of sleep was dropped as it had lowest correlation

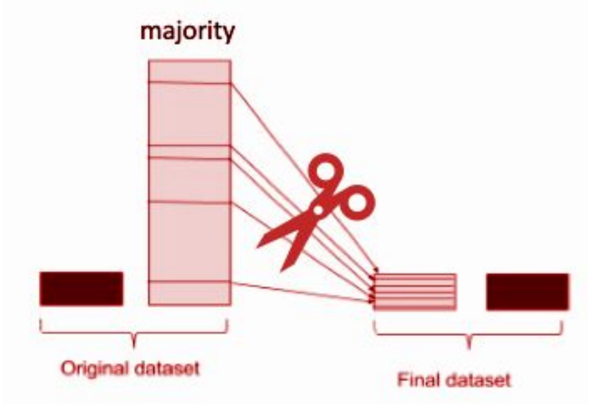
Tableau

Visuals will show Demographic makeup of our residents as well as gauge how health they are.

UNDERSAMPLING vs OVERSAMPLING

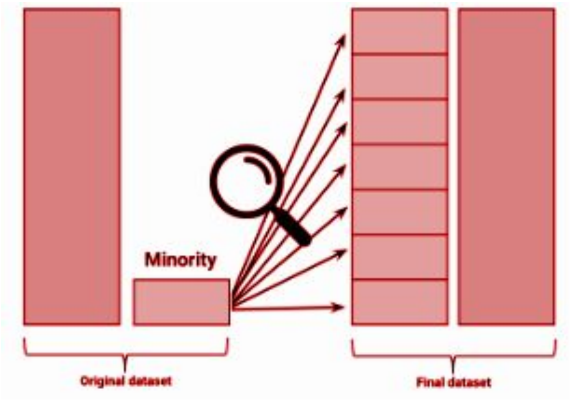
RANDOM UNDERSAMPLE

Randomly deletes examples from the majority group up to the count of the minority group



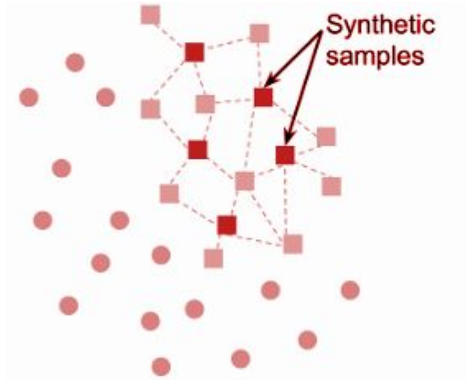
RANDOM OVERSAMPLE

Randomly duplicates the minority group to match the majority group



SMOTE (Synthetic Minority Oversampling Technique)

Synthesizes new examples by selecting a minority class instance at random and finding its K nearest minority class in order to generate a combination of the two chosen instances



Undersample

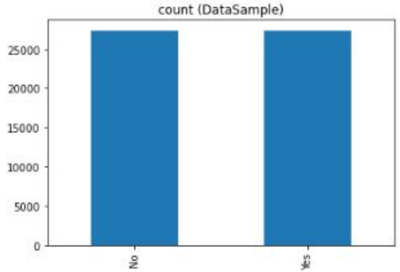
Data

```
# Heart Disease Counts
NoDisease_count, Disease_count = df_og.HeartDisease.value_counts()

# Separate the yes and no
NoDisease = df_og[df_og['HeartDisease'] == 'No']
Disease = df_og[df_og['HeartDisease'] == 'Yes']

# print the shape
print('No Disease:', NoDisease.shape)
print('Disease:', Disease.shape)
```

```
No Disease: (292422, 18)
Disease: (27373, 18)
```

```
NoDisease_under = NoDisease.sample(Disease_count)
df = pd.concat([NoDisease_under, Disease], axis=0)
print(df['HeartDisease'].value_counts())
# plot the count after under-sampeling
df['HeartDisease'].value_counts().plot(kind='bar', title='count (DataSample)')
```

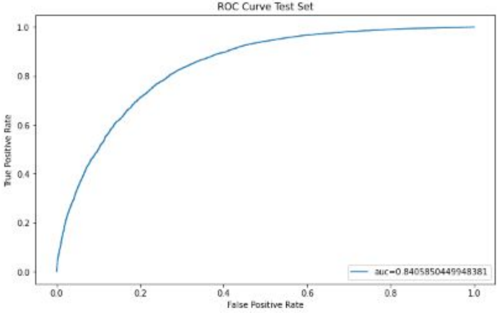
```
No     27373
Yes    27373
Name: HeartDisease, dtype: int64

<AxesSubplot:title={'center':'count (DataSample)'}>
```



Results - LGBM model

```
lgbm = LGBMClassifier(random_state=42)
lgbm = evaluateModel(lgbm, X_train, y_train, X_test, y_test)
```

```
TRAINING SET
              precision    recall  f1-score   support

           0       0.81      0.74      0.77     20548
           1       0.76      0.82      0.79     20511

    accuracy                           0.78     41059
   macro avg       0.78      0.78      0.78     41059
weighted avg       0.78      0.78      0.78     41059

[[15173  5375]
 [ 3636 16875]]

Testing SET
              precision    recall  f1-score   support

           0       0.79      0.73      0.76      6825
           1       0.75      0.81      0.78      6862

    accuracy                           0.77     13687
   macro avg       0.77      0.77      0.77     13687
weighted avg       0.77      0.77      0.77     13687

[[4959 1866]
 [1327 5535]]
```



Oversample - RandomOverSampler

Data

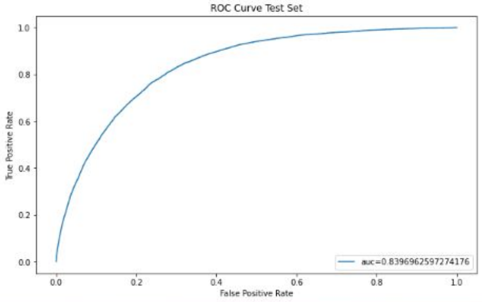
```python
from imblearn.over_sampling import RandomOverSampler
from collections import Counter
ros = RandomOverSampler(random_state=42)
X_resampled, y_resampled = ros.fit_resample(X_train, y_train)

Counter(y_resampled)
```

```
Counter({0: 219418, 1: 219418})
```

Results- LGBM Model

```python
In [97]: lgbm = LGBMClassifier(random_state=42)
         lgbm = evaluateModel(lgbm, X_resampled, y_resampled, X_test, y_test)
```

```
TRAINING SET
              precision    recall  f1-score   support

           0       0.81      0.73      0.76    219418
           1       0.75      0.82      0.79    219418

    accuracy                           0.77    438836
   macro avg       0.78      0.77      0.77    438836
weighted avg       0.78      0.77      0.77    438836

[[159170  60248]
 [ 38518 180900]]

Testing SET
              precision    recall  f1-score   support

           0       0.98      0.72      0.83     73004
           1       0.22      0.81      0.34      6945

    accuracy                           0.73     79949
   macro avg       0.60      0.76      0.59     79949
weighted avg       0.91      0.73      0.79     79949

[[52764 20240]
 [ 1339  5606]]
```



Oversample - SMOTE

Data

```python
# Resample the training data with SMOTE
from imblearn.over_sampling import SMOTE
X_resampled_smote, y_resampled_smote = SMOTE(random_state=1).fit_resample(
    X_train, y_train
)
Counter(y_resampled_smote)
```

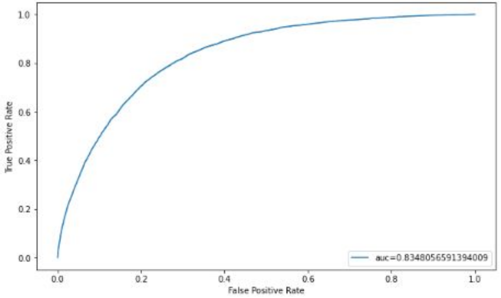
```
Counter({0: 219418, 1: 219418})
```

Results - Logistic Regression

```python
# Initialize the model
lr = LogisticRegression()
lr = evaluateModel(lr, X_resampled_smote, y_resampled_smote, X_test, y_test)
```

```
TRAINING SET
              precision    recall  f1-score   support

           0       0.78      0.75      0.76    219418
           1       0.76      0.80      0.78    219418

    accuracy                           0.77    438836
   macro avg       0.77      0.77      0.77    438836
weighted avg       0.77      0.77      0.77    438836

[[163552  55866]
 [ 44816 174602]]

Testing SET
              precision    recall  f1-score   support

           0       0.97      0.74      0.84     73004
           1       0.22      0.78      0.35      6945

    accuracy                           0.75     79949
   macro avg       0.60      0.76      0.59     79949
weighted avg       0.91      0.75      0.80     79949

[[54248 18756]
 [ 1540  5405]]
```

ROC Curve Test Set



02

Understanding Model Results

Confusion Matrix

		Actuals	
		No Heart Disease	Heart Disease
Predicted	No Heart Disease	✓	✗
	Heart Disease	✗	✓

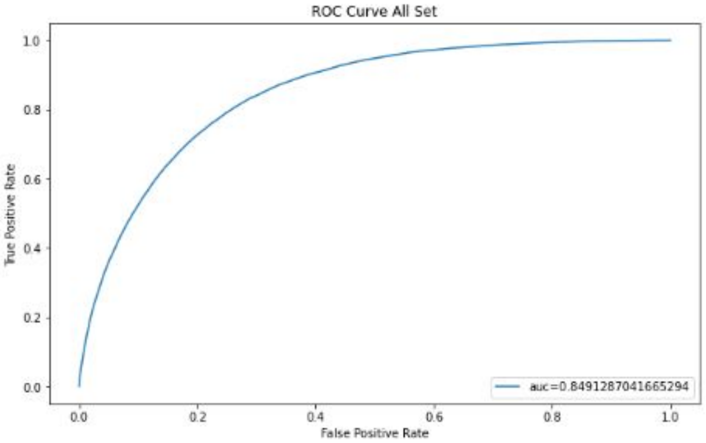
Type 2 error:
False Negative
Sick people predict healthy

Type 1 error:
False positive
Healthy people predicted sick

LGBM Classifier model trained using Undersampling

Model Results on Original Data

```
              precision    recall  f1-score   support

           0       0.98      0.72      0.83    292422
           1       0.22      0.82      0.34     27373

    accuracy                           0.73    319795
   macro avg       0.60      0.77      0.59    319795
weighted avg       0.91      0.73      0.79    319795

[[211872  80550]
 [  4963  22410]]
```



Analysis

The LGBM model had the best results with an AUC of 84% and recall of 82% for respondents with Heart Disease.

Overall accuracy between all *5 models used using Undersampling and oversampling techniques ranged between 77% and 84%.

*Logistic Regression, XGB, LGBM, RandomForest, AdaBoost

03

Research Questions

Questions for project

Question 1

What Health Conditions affect Heart Disease?

Question 3

Does mental health affect heart disease

Question 2

Do Male or Females have a higher risk of heart disease?

Question 4

Are people of a certain race more likely to have heart disease?

Tableau:

Tableau was helpful in exploring the demographic makeup of our residents. As well as observing the overall health our our residents.

Or color scheme in our plots are from the Number of people that reported having heart disease.

Our Filters are based on whether or residents said they have heart disease or not. As well as Race, and BMI count

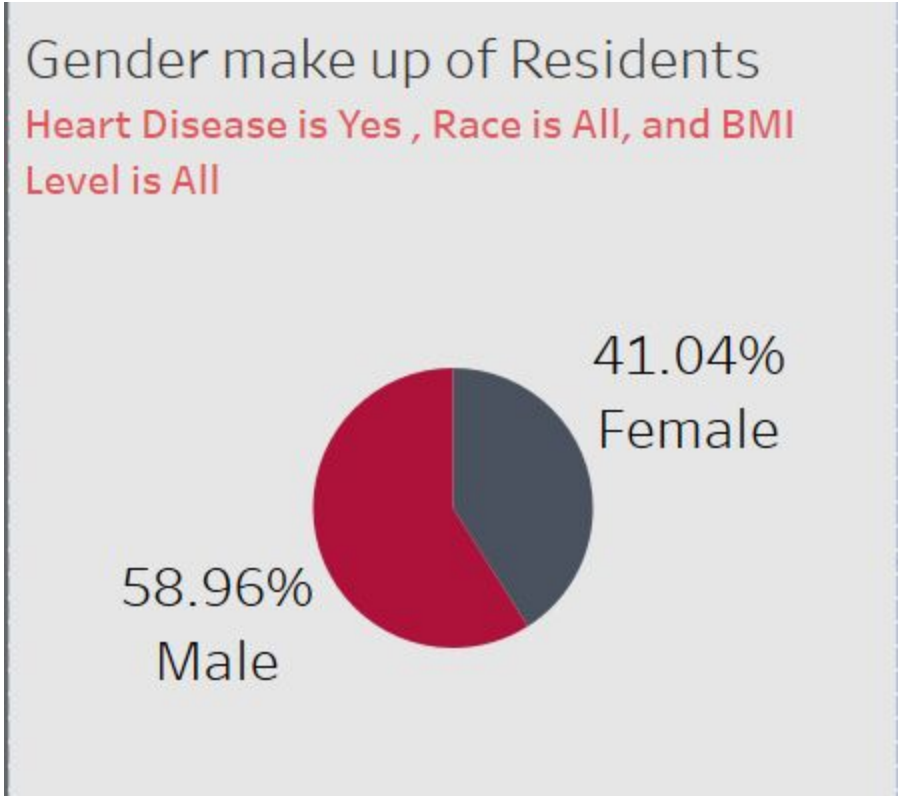


Tableau:

We found a huge difference in our race values

Age Group was already given to us

Most of the people that reported a heart disease were older than 65.

Tableau:

Tableau Issues:

A lot of Categorical (Dimension) Columns

Finding the best way to plot our Numeric (Measure) Columns

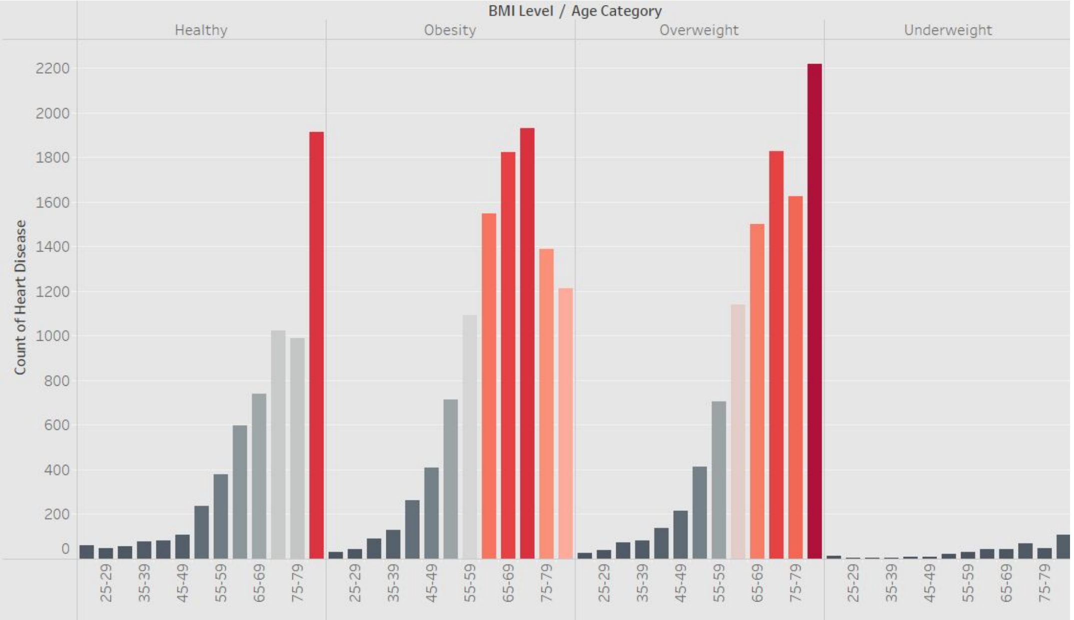
Created Calculated fields for the BMI, the physical and mental health columns

```
IF ([Physical Health] >=0) AND ([Physical Health] < 11) THEN
"PH 0-10"
ELSEIF ([Physical Health] >=11) AND ([Physical Health] < 21) THEN
"PH 11-20"
ELSE
"PH 21-30"
END
```

```
IF ([Mental Health] >=0) AND ([Mental Health] < 11) THEN
"MH 0-10"
ELSEIF ([Mental Health] >=11) AND ([Mental Health] < 21) THEN
"MH 11-20"
ELSE
"MH 21-30"
END
```

```
IF ([BMI] >=0) AND ([BMI] < 18.5) THEN "Underweight"
ELSEIF ([BMI] >=18.5) AND ([BMI] < 25) THEN "Healthy"
ELSEIF ([BMI] >=25) AND ([BMI] < 30) THEN "Overweight"
Else "Obesity"
END
```

Physical ..	Mental Health 0	1	2	3	4	5	6	7	8	9	10
0		275	419	270	133	309	31	123	23	7	228
1	386	60	42	19	11	24	2	6			10
2	696	53	109	41	28	44	9	17	1		37
3	488	36	51	59	17	45	4	21	3		22
4	284	14	31	18	20	18	8	7	3	1	16
5	500	25	53	36	11	80	6	13	4	1	33
6	97	4	13	8	3	5	6	3	1		9
7	258	11	25	24	7	13	3	30	3		13
8	53	4	10	4	4	7	1	3	5	2	5
9	13		1	1	1	3		4		3	2
10	435	12	34	27	8	53	1	4	2	1	73



We can now filter the number of days for the mental and physical Health

We collected the BMI Levels from the CDC website

[About Adult BMI | Healthy Weight, Nutrition, and Physical Activity | CDC]

We know our age groups but now we can see how healthy they are based on their BMI count.

04

Website Exploration & Heart Disease Prediction

05

Limitations & Future Work

Future Work

Machine Learning Models

- Try to Narrow down columns to get best results when training model.
- Explore different solutions such as SMOTEE to balance data or find better dataset with more respondents who have Heart Disease.
- Explore other Machine Learning Models to see if there are better results
- Add the % predictor to the results on our Webpage

Limitations

Limitations

- Dataset is heavily imbalanced
- Too much categorical data (yes & no) limited to few visualizations.
- Oversampling model over 100MB not able to submit to Github/heroku
- Data is limited to coronary heart disease (CHD) or myocardial infarction (MI). Would like to predict what type of heart disease recipients may have.

CONCLUSIONS

LGBM is our best trained model to use with outside data. Oversampling our data have a potential of overfitting thus, would not accurately predict on non original data. Next step load the next 2021 dataset using the model and see how it compares.

REFERENCES

- https://www.kaggle.com/code/arkalodh/prediction-of-heart-disease-easy
- https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease
- https://www.cdc.gov/heartdisease/risk_factors.htm
- https://en.wikipedia.org/wiki/Precision_and_recall
- https://www.kaggle.com/code/arkalodh/prediction-of-heart-disease-easy
- https://www.kaggle.com/code/houssemeddinedhahri/eda-prediction-with-7-models/notebook
- https://world-heart-federation.org/
- https://public.tableau.com/app/profile/shreerangscp/viz/HeartDisease_16303068486410/HeartDiseaseAnalysis
- https://public.tableau.com/app/profile/jeff.ho4188/viz/HeartDiseaseResearchDashboard/Dashboard

THANK YOU!
Stay Healthy!
Keep in Touch!