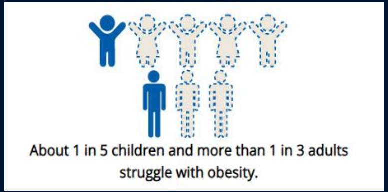
Analyzing Risk Factors Associated with Obesity Using Machine Learning

Final Presentation

DATA 606 - Capstone in Data Science
UMBC
Group Members:
Sandra Pinto, Shruthi Boban, Siyu Ma

Introduction

- ❖ Our project aims to analyze the relationship between obesity and different risk factors such as BMI, Race, Gender, physical activities, mental health, education level, and etc.
- ❖ Obesity constitutes a major public health concern in the U.S. and Globally
    - ➢ About 1 in 5 children and more than 1 in 3 adults struggle with obesity in the U.S. (CDC)
    - ➢ Adults with obesity have higher risk for developing Heart disease, Type 2 diabetes, and some types of cancer (CDC)
    - ➢ According to the “World Health Organization” (WHO), 30% of global death will be caused by lifestyle diseases by 2030.
- ❖ There are limited number of studies using machine learning to analyze obesity related datasets in the U.S.



Research Questions/Approach

Research Questions:

- ❖ Which variables are risk factors related to obesity?
- ❖ What are the correlations between different risk factors and BMI?
    - ➢ Is mental health an important factor that correlates with obesity?
- ❖ Which machine learning model can classify the dataset more accurately?

Approach:

- ❖ Conduct EDA to find the relationship of different factors and produce visualizations
- ❖ Find the most accurate model for our dataset.
    - ➢ Classification models (e.g Random Forest, Support Vector Machines (SVM), Logistic Regression, Decision Trees, and XGBoost)

Literature/Industry Research Review

- The Technology and Health Departments of the University of Agder (Norway) identified potential risk factors associated with obesity/overweight using machine learning methods such as Support Vector Machines (SVM), Decision Trees, and Logistic regression models. (Chatterjee et al, 2021)
- The University of Bologna (Italy) used ML techniques to test for the predictive effects of emotional and affective variables over BMI values. (Delnevo et al, 2021)
- The Daffodil International University in Dhaka (Bangladesh) applied 9 prominent ML algorithms to predict the risk of obesity on the data collected from many varieties of people of different ages suffering from obesity and non-obesity. (Ferdowsy F. et al, 2021)

Dataset

- **SEQN:** Respondent sequence number
- **Gender:** 1 = Male, 2 = Female
- **Age:** Age in years
- **Race:** 1= Mexican American, 2 = Other Hispanic, 3 = None-Hispanic White, 4 = None-Hispanic Black, 6 = None-Hispanic Asian, 7 = Other Race - Including Multi-Racial
- **Country of birth**: 1 = Born in 50 US states or Washington, DC, 2 = Other
- **Education level - Adults 20+**: 1 = Less than 9th grade, 2 = 9-11th grade(Includes 12th grade with no diploma), 3 = High School graduate/GED or equivalent, 4 = Some college or AA degree, 5 = College graduate or above,
- **Ratio of family income to poverty**: numerical values from 0 to 5.00
- Weight in kg
- Height in cm
- Body mass index - BMI
- Doctor told you have diabetes: 1 = Yes, 2 = No, 3 = Borderline
- **Moderate work activity:** 1 = Yes, 2 = No
- **Moderate recreational activities:** 1 = Yes, 2 = No
- **Feeling down, depressed, or hopeless**: 0 = Not at all, 1 = Several days, 2 = More than half the days, 3 = Nearly every day
- **Poor appetite or overeating:** 0 = Not at all, 1 = Several days, 2 = More than half the days, 3 = Nearly every day
- **Sleep hours - weekdays or workdays:** range of values
- **Sleep hours - weekends**: range of values
- **Do you now smoke cigarettes?** 1= Every day, 2 = Some days, 3 = Not at all

Source:
CDC - National Center for Health Statistics.
National Health and Nutrition Examination Survey
March 2017 to 2020 Pre-pandemic

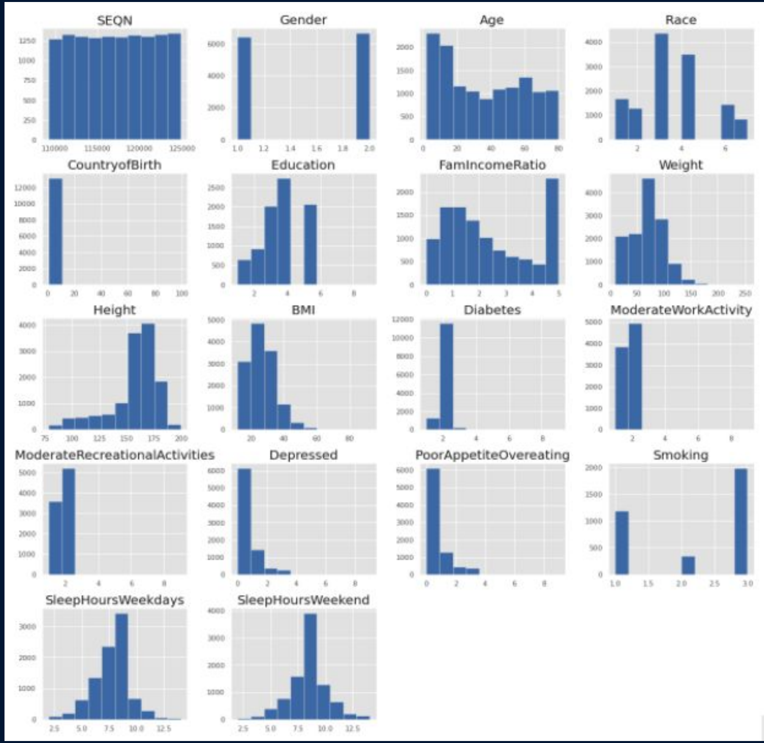
Combined 7 different raw datasets

Size: 12.4MB - XPT. files

	SEQN	Gender	RIDAGEYR	Race	CountryofBirth	Education	FamIncomeRatio	Weight	BMI	BMICategory	Diabetes	ModerateWorkActivity	ModerateRecreationalActivities	Depressed	PoorAppetiteOvereating	SleepHoursWeekdays	SleepHoursWeekend
0	109263.0	1.0	2.0	6.0	1.0	4.0	4.66	65.42638	26.656847	2.0	2.0	2.0	2.0	5.397605e-79	5.397605e-79	7.64092	8.361768
1	109264.0	2.0	13.0	1.0	1.0	4.0	0.83	42.20000	17.600000	2.0	2.0	2.0	2.0	5.397605e-79	5.397605e-79	7.64092	8.361768
2	109265.0	1.0	2.0	3.0	1.0	4.0	3.06	12.00000	15.000000	2.0	2.0	2.0	2.0	5.397605e-79	5.397605e-79	7.64092	8.361768
3	109266.0	2.0	29.0	6.0	2.0	5.0	5.00	97.10000	37.800000	2.0	2.0	2.0	1.0	5.397605e-79	5.397605e-79	7.50000	8.000000
4	109267.0	2.0	21.0	2.0	2.0	4.0	5.00	65.42638	26.656847	2.0	2.0	2.0	2.0	5.397605e-79	5.397605e-79	8.00000	8.000000

EDA & Visualizations

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13137 entries, 1 to 15559
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   SEQN                            13137 non-null  int64
 1   Gender                          13137 non-null  int64
 2   Age                             13137 non-null  int64
 3   Race                            13137 non-null  int64
 4   CountryofBirth                  13137 non-null  int64
 5   Education                       8381 non-null   float64
 6   FamIncomeRatio                  11443 non-null  float64
 7   Weight                          13137 non-null  float64
 8   Height                          13137 non-null  float64
 9   BMI                             13137 non-null  float64
 10  Diabetes                        13137 non-null  int64
 11  ModerateWorkActivity            8790 non-null   float64
 12  ModerateRecreationalActivities  8790 non-null   float64
 13  Depressed                       8203 non-null   float64
 14  PoorAppetiteOvereating          8202 non-null   float64
 15  Smoking                         3521 non-null   float64
 16  SleepHoursWeekdays              9188 non-null   float64
 17  SleepHoursWeekend               9183 non-null   float64
dtypes: float64(12), int64(6)
memory usage: 1.9 MB
```



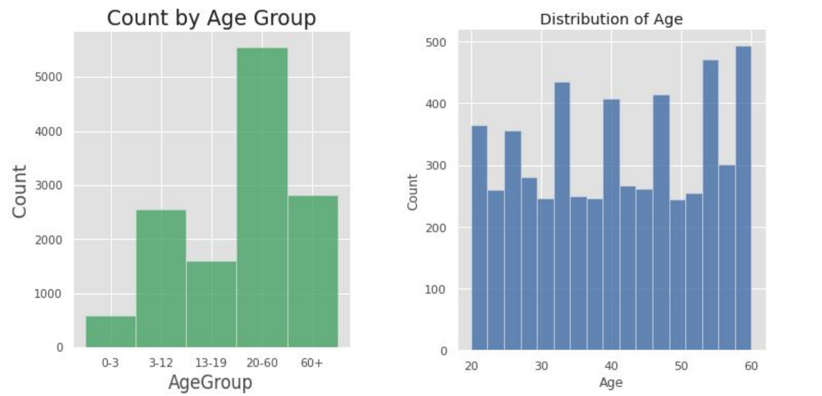
Basic histogram shows data distribution and frequency counts.

EDA & Visualizations

Dataset contains infants, children, teenagers, adults, and seniors.

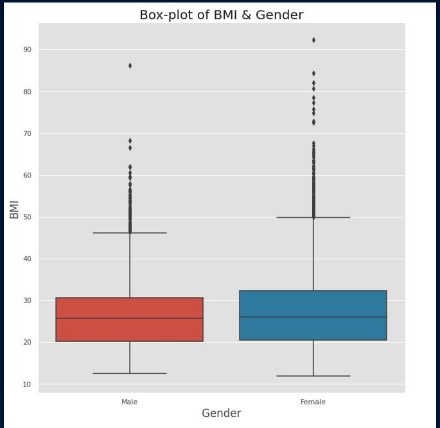
- Added a new column and separated respondents to different age groups.

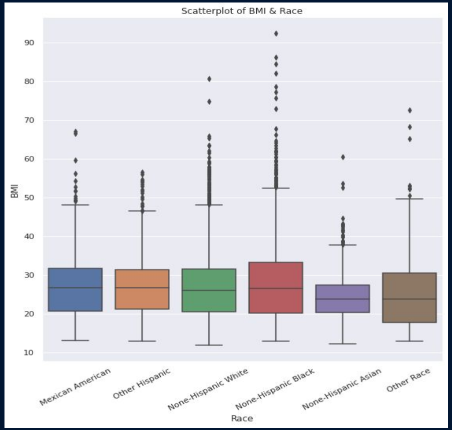
```
# Filter the age, extract respondents from 20-60 years old.
df_age_filter = df[(df['Age'] >= 20) & (df['Age'] <= 60)]
```

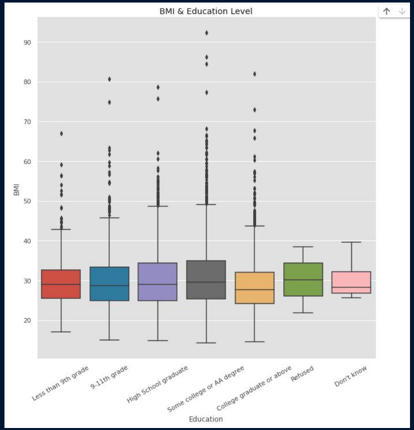


- Respondents below 20 or older than 60 years old - Missing information such as education, family income ratio, diabetes, activities, eating disorder status, and smoking habit.
- Decided to extract the respondents' data between 20 to 60 years old to a new data frame.

EDA & Visualizations





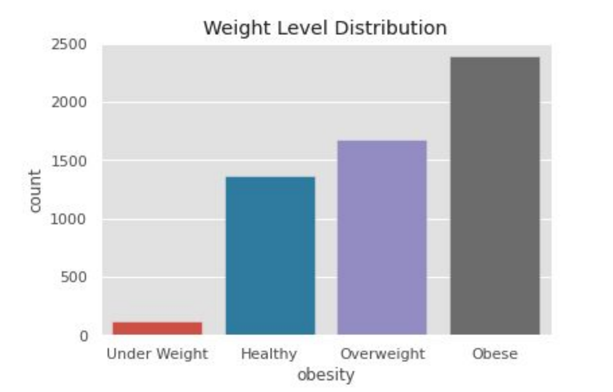


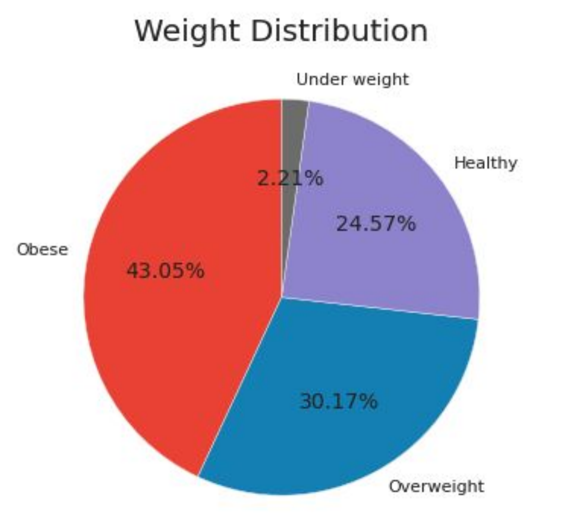
1. Female respondents have slightly higher BMI than Male respondents.
2. None-Hispanic Asians have lower BMI compared to other race groups in our dataset.
3. People with college or above education level tend to have lower BMI.

EDA & Visualizations

Added a column “obesity” showing weight level for each respondent based on CDC BMI guideline.

BMI	Weight Status
Below 18.5	Underweight
18.5 – 24.9	Healthy Weight
25.0 – 29.9	Overweight
30.0 and Above	Obesity



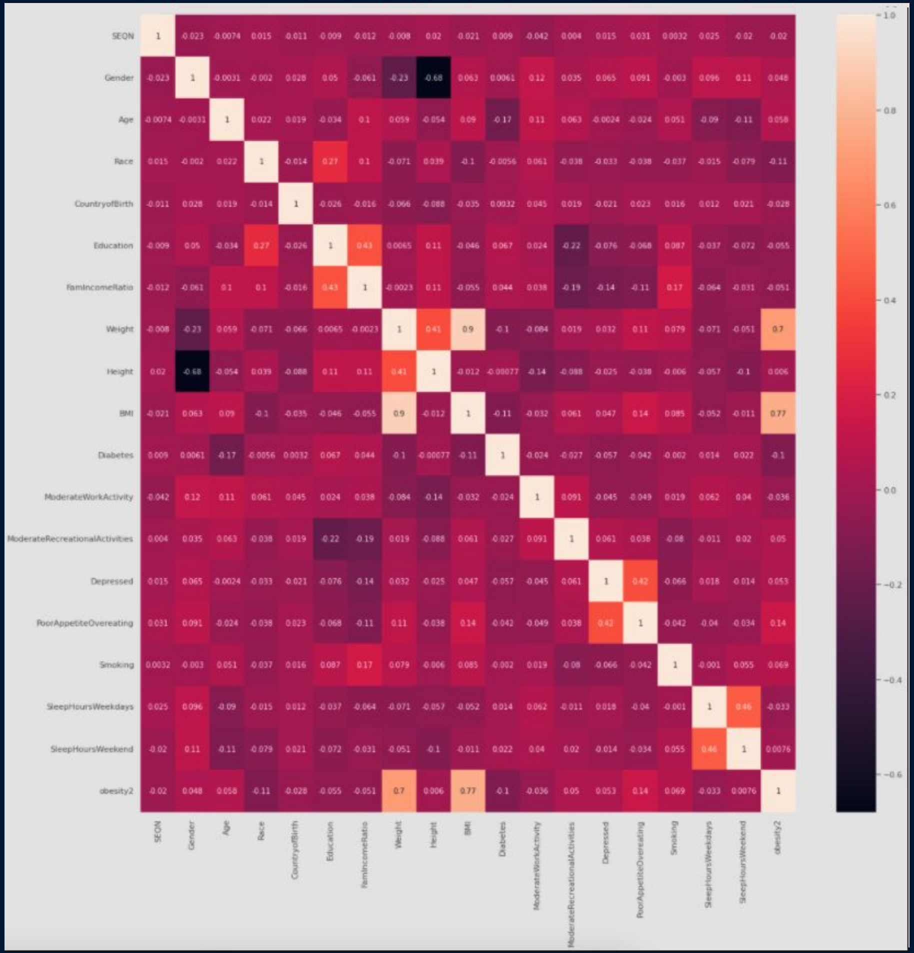


Obese: 43.05%
Overweight: 30.17%
Healthy: 24.57%
Underweight: 2.21%

EDA & Visualizations

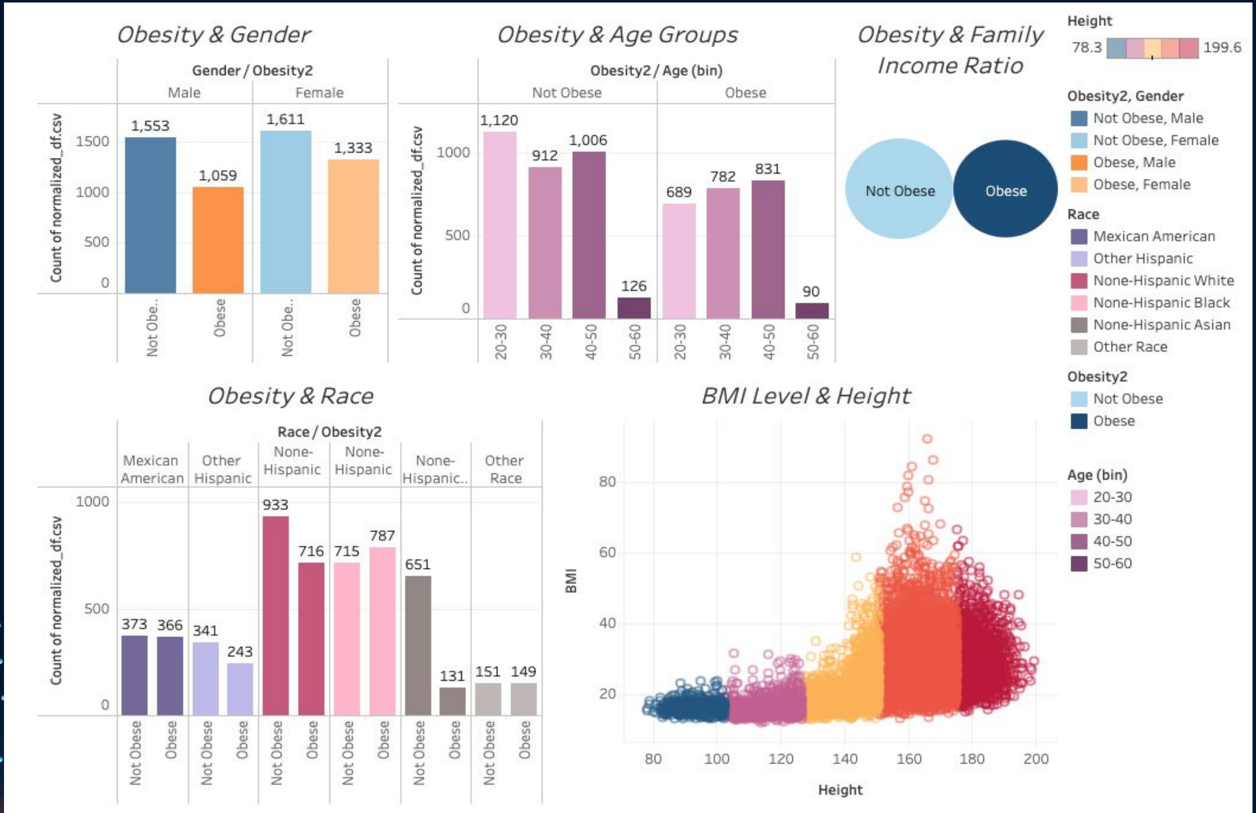
Heatmap shows the correlation between different risk factors.

- Obesity level is highly correlated with weight and BMI.
- After removing BMI and Weight, we can see that obesity level related to Poor appetite/overeating, diabetes, and race.
- High correlation: Poor appetite/overeating and Depressed; Education and Family income ratio; Height and Gender;



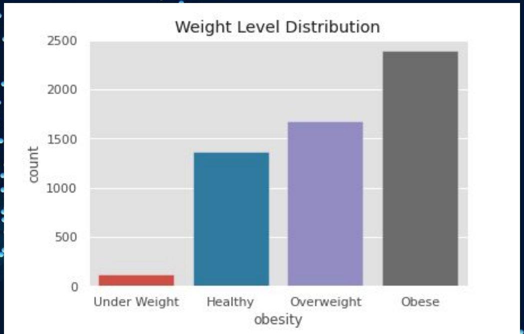
Dashboard

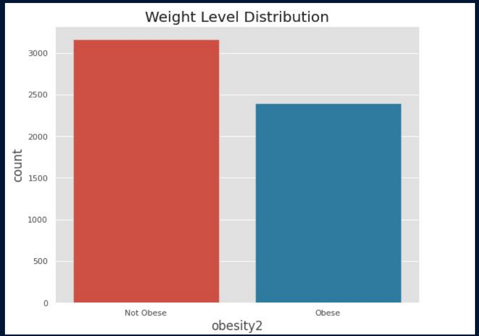
https://public.tableau.com/views/Book1_16381205995830/Dashboard1?:language=en-US&:display_count=n&:origin=viz_share_link



Preparation & Model Construction

- After checking weight distributions we discovered that there was a class imbalance thus, we combined the respondents from the underweight, healthy, and overweight groups together and kept the obese group separate. (1 = Not Obese, 2 = Obese)
- Dropped Weight and BMI from the dataset: Weight and BMI are highly correlated with obesity/overweight
- Normalized the data using mean-max transformation which scaling each variable to the range (0, 1).
- We split the data into training and testing sets at a 80% to 20% ratio.
- For our modeling section, we used a total of 6 models: Baseline, Random Forest, Logistic Regression, SVM, Decision Trees, and XGBoost to predict accuracy and feature importance of risk factors





Baseline Model

- We decided to create a baseline classification model as a benchmark
    - A simple model that provides reasonable results on a task or a metric you would hope any model could beat.
- DummyClassifier is a classifier that makes predictions using simple rules. We use this to build a baseline model to compare with other models.
- The baseline model has a 57.64% accuracy, which indicates the lowest possible prediction we can get. We expect to get higher accuracy from other models we selected.

```
# Baseline classification accuracy
from sklearn.dummy import DummyClassifier

baseline_classifier = DummyClassifier(strategy = "most_frequent")
baseline_classifier.fit(X_train,Y_train)

# predicting
Y_pred_base = baseline_classifier.predict(X_test)

# accuracy calculation
from sklearn import metrics

print("Accuracy:", metrics.accuracy_score(Y_test, Y_pred_base))
```

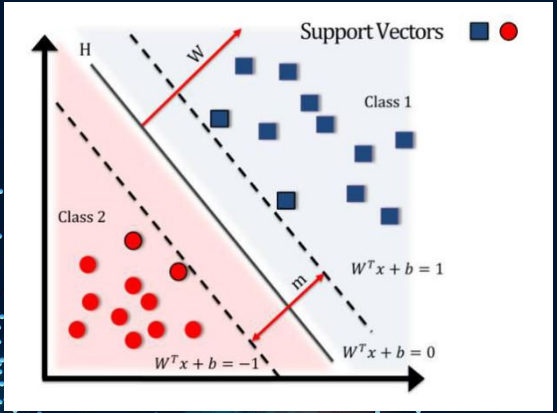
```
Accuracy: 0.5764388489208633
```

SVM Model

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression, and outliers detection.Given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

The accuracy rate of the SVM model is 63.49%, the cross validation score is 60.91%.

The precision rate of the SVM model is 64%, the false-positive rate is 87% which indicates is not good fit.



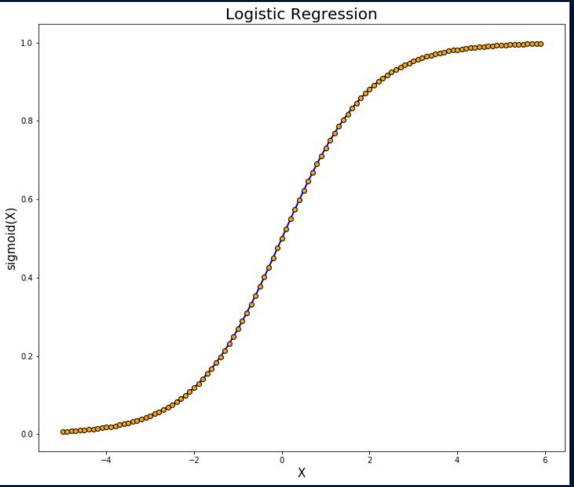
	precision	recall	f1-score	support
0	0.63	0.87	0.73	641
1	0.64	0.32	0.43	471
accuracy			0.63	1112
macro avg	0.64	0.59	0.58	1112
weighted avg	0.64	0.63	0.60	1112

Logistic Regression Model

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. It is mainly used when the target variable is categorical.

- The accuracy is 61.33%, cross-validation is 60.62%, the accuracy and cross-validation score are not high, but they are close to each other.
- The precision rate is 57%, and the false positive rate is 81% which shows this is not a good model for our dataset.

	precision	recall	f1-score	support
0	0.63	0.81	0.71	641
1	0.57	0.35	0.43	471
accuracy			0.61	1112
macro avg	0.60	0.58	0.57	1112
weighted avg	0.60	0.61	0.59	1112



Decision Tree Model

The goal of using a Decision Tree model is to create a training model that can be used to predict/classify the value of the target variable by learning simple decision rules inferred from training data.

- Decision Tree Model result shows the accuracy as 58.13% and cross validation score as 57.05%.
- The precision rate is 51%, false positive rate is 63% which shows this is not a good model for our dataset.

	precision	recall	f1-score	support
0	0.64	0.63	0.64	641
1	0.51	0.51	0.51	471
accuracy			0.58	1112
macro avg	0.57	0.57	0.57	1112
weighted avg	0.58	0.58	0.58	1112

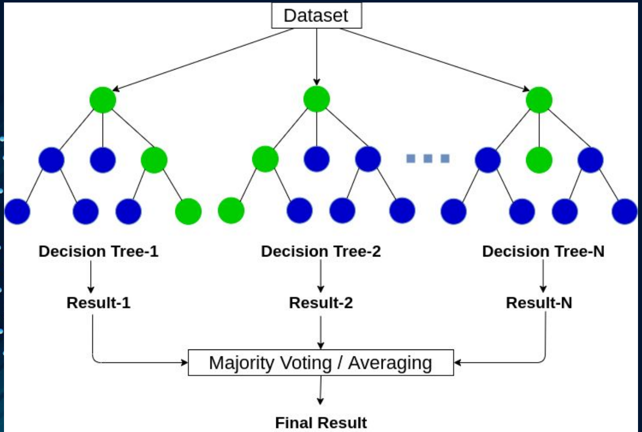
Random Forest Model

A random forest is a supervised machine learning algorithm that is constructed from decision tree algorithms.

- Does not suffer overfitting, cancel biases from taking average of predictions.

	precision	recall	f1-score	support
0	0.66	0.77	0.71	641
1	0.59	0.45	0.51	471
accuracy			0.63	1112
macro avg	0.62	0.61	0.61	1112
weighted avg	0.63	0.63	0.62	1112

Accuracy of Random Forest Model = 63.40%
Cross-validation score = 63.62%



XGBoost Model

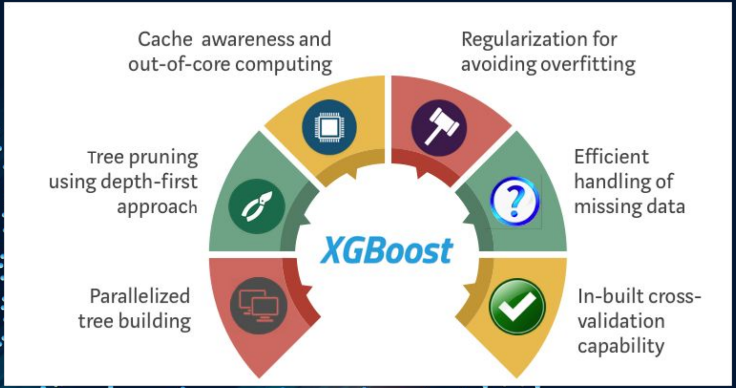
XGBoost is a decision-tree-based ensemble ML algorithm.

- Uses gradient boost framework
- Delivers more accurate approximations by using the second order derivative of the loss function.

	precision	recall	f1-score	support
0	0.67	0.78	0.72	641
1	0.61	0.47	0.53	471
accuracy			0.65	1112
macro avg	0.64	0.63	0.63	1112
weighted avg	0.64	0.65	0.64	1112

Accuracy of XGBoost Model = 64.93%

Cross-Validation score = 63.48%

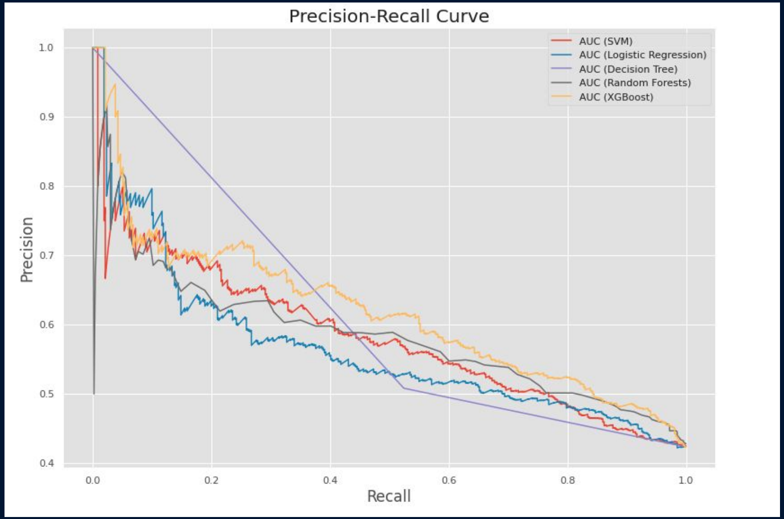


Model Evaluation

Precision: how much was correctly classified as positive out of all the positives.

Recall: the ratio between how much was correctly identified as positive to all the actual positives.

F1-score: the weighted average between precision and recall.

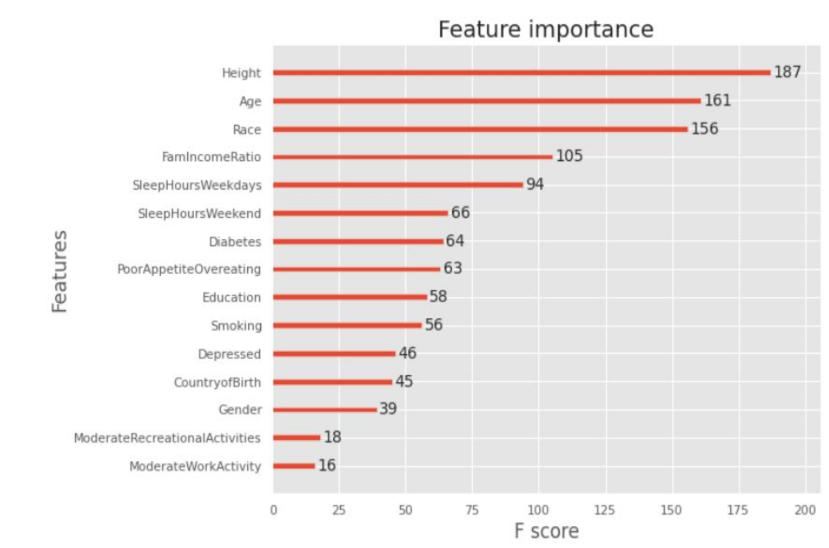


Algorithms	Accuracy	Precision	Recall	F1-score
SVM	63.49%	64%	32%	43%
Logistic Regression	61.33%	57%	35%	43%
Decision Tree	58.36%	51%	51%	51%
Random Forest	63.40%	59%	45%	51%
XGBoost	64.93%	61%	47%	53%

```
AUC of Logistic Regression: 0.56
AUC of SVM: 0.58
AUC of Random Forest: 0.58
AUC of Decision Tree: 0.61
AUC of XGBoost: 0.62
```

Conclusion

Feature Importance- XGBoost Model



- Although the accuracy levels from our models were considerably low, we do have a significant improvement compared to the baseline model.
- The XGBoost Model provided the best accuracy score compared to the other models so we decided to check the feature importance
- The top 6 risk factors are Height, Age, Race, Family income ratio, Sleep hours on weekdays, and Sleep hours on weekends.
- In our literature review, we learned that depression can affect obesity levels. However, based on our analysis we can not say that mental health is highly affecting obesity level.

Limitations & Future Study

Limitations

- Limited access to robust open source healthcare datasets due to US laws such as HIPAA (protects sensitive patient health information).
- The accuracy levels from our models were considerably low but higher than baseline model

Future Study

- Apply Neural Network with Backpropagation in order to self learn and improve the accuracy while feeding in new data.
- Find a better dataset to do in depth research and build prediction models for other relevant disease
    - Build a web interface/tool for disease prediction such as Diabetes

References

https://blog.ml.cmu.edu/2020/08/31/3-baselines/

https://blog.insightdatascience.com/always-start-with-a-stupid-model-no-exceptions-3a22314b9aaa

https://xgboost.readthedocs.io/en/latest/python/python_api.html

https://www.datacamp.com/community/tutorials/random-forests-classifier-python

THANK YOU!!