Diabetes Detection

Using Machine Learning to Make Health Predictions

August 31, 2022

Nashra Khan | Jason Noble | Marie Sanon | Will Wright

Agenda

- Why this Dataset?
- Visualizing the target and features variables
- Data Cleaning
- Machine Learning Algorithms
- Tuning the Models
- The BEST Model
- Web App Demo

Why this dataset?

Diabetes is among the most common chronic diseases in the United States impacting millions of Americans each year.

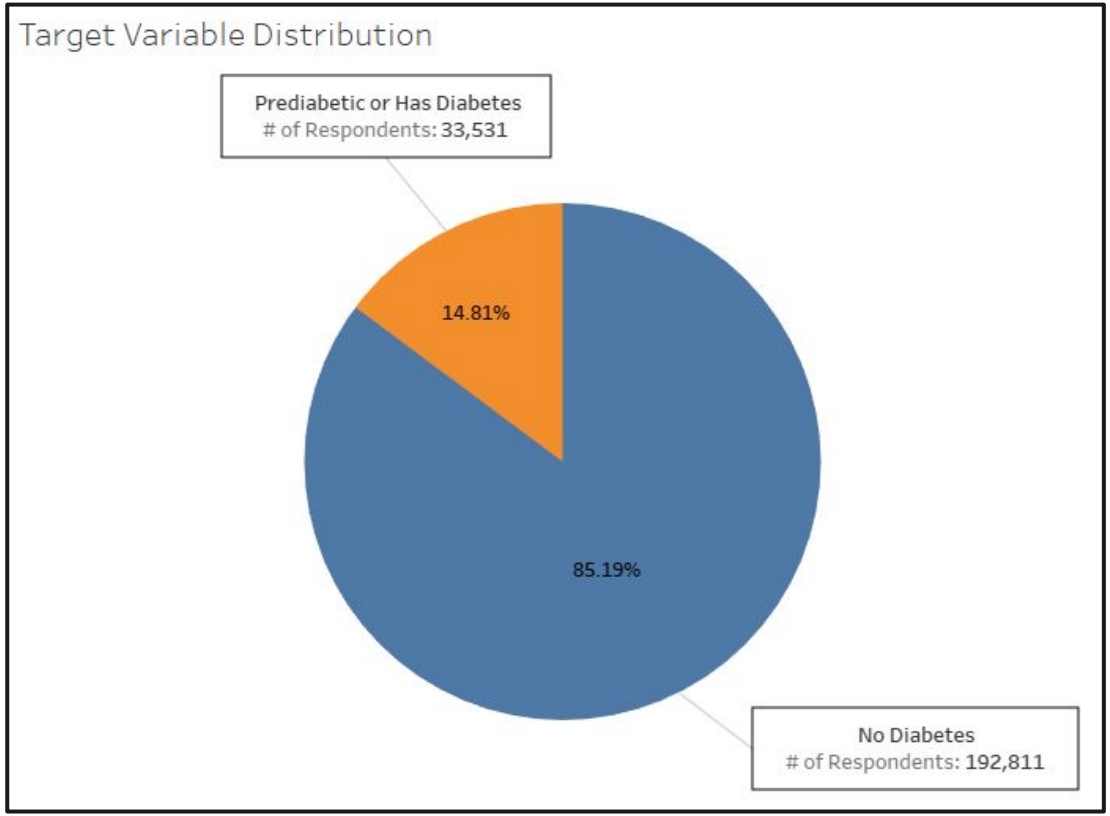
This dataset is the result of a Behavioral Risk Factor Surveillance System (BRFSS) telephone survey that is collected annually by the CDC and was made available on Kaggle.

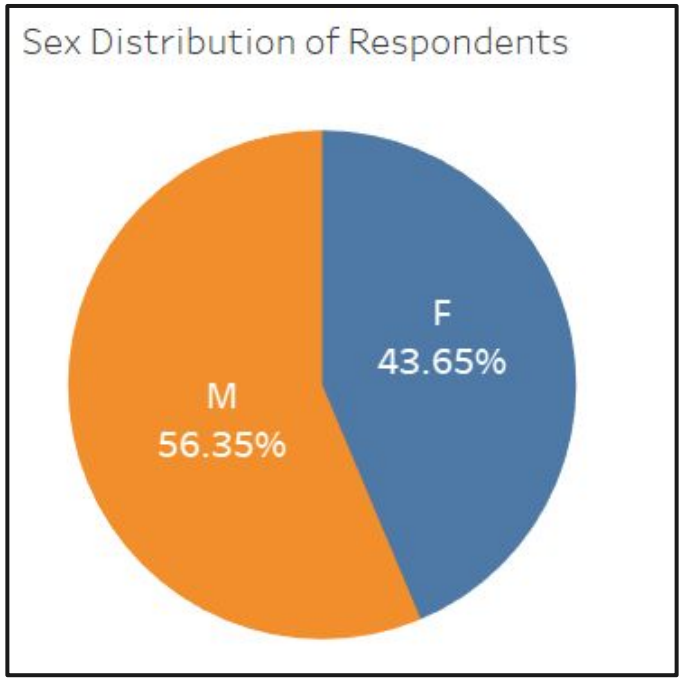
The CDC estimates that 1 in 5 diabetics, and roughly 8 in 10 pre-diabetics are unaware of their risk. Early diagnosis can lead to lifestyle changes and more effective treatment.

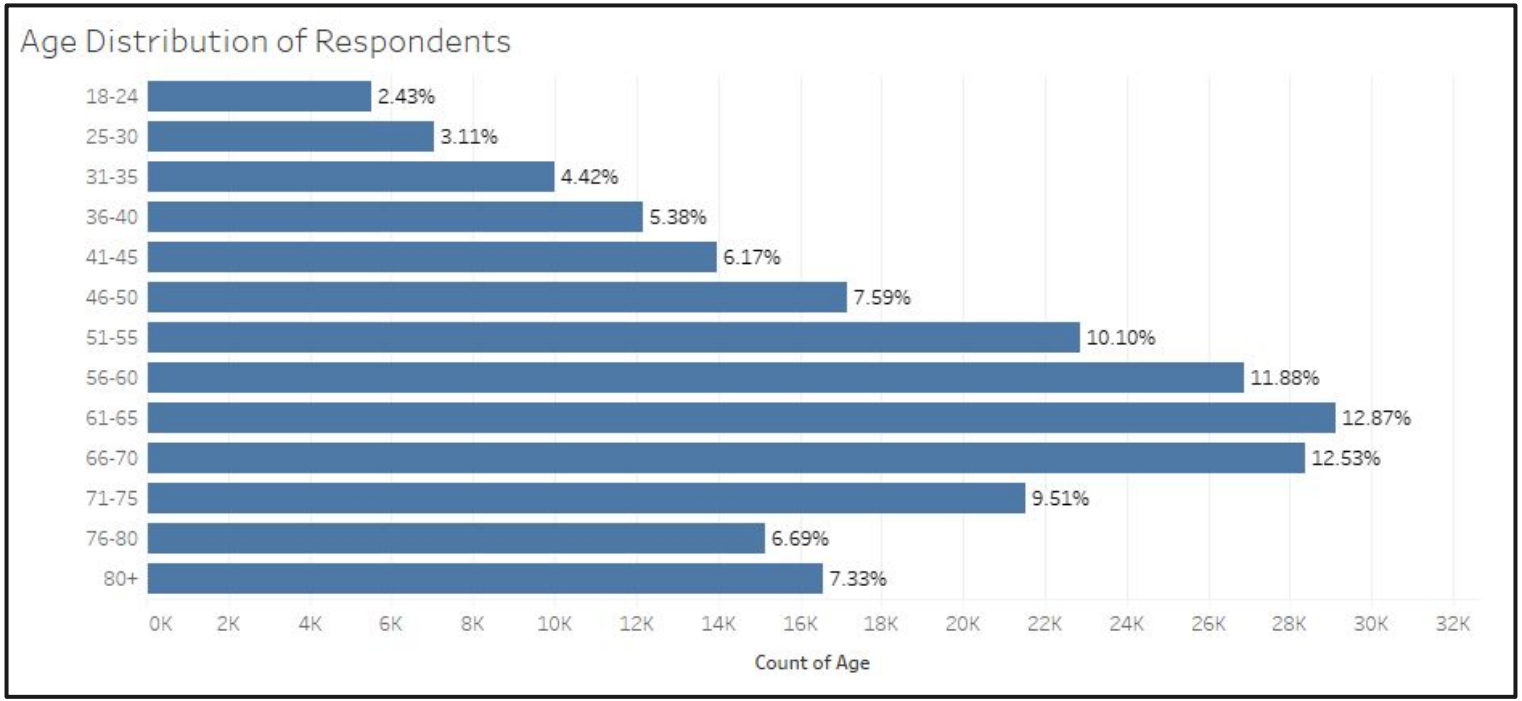
Using this dataset, a predictive model seeks to assist in early detection.

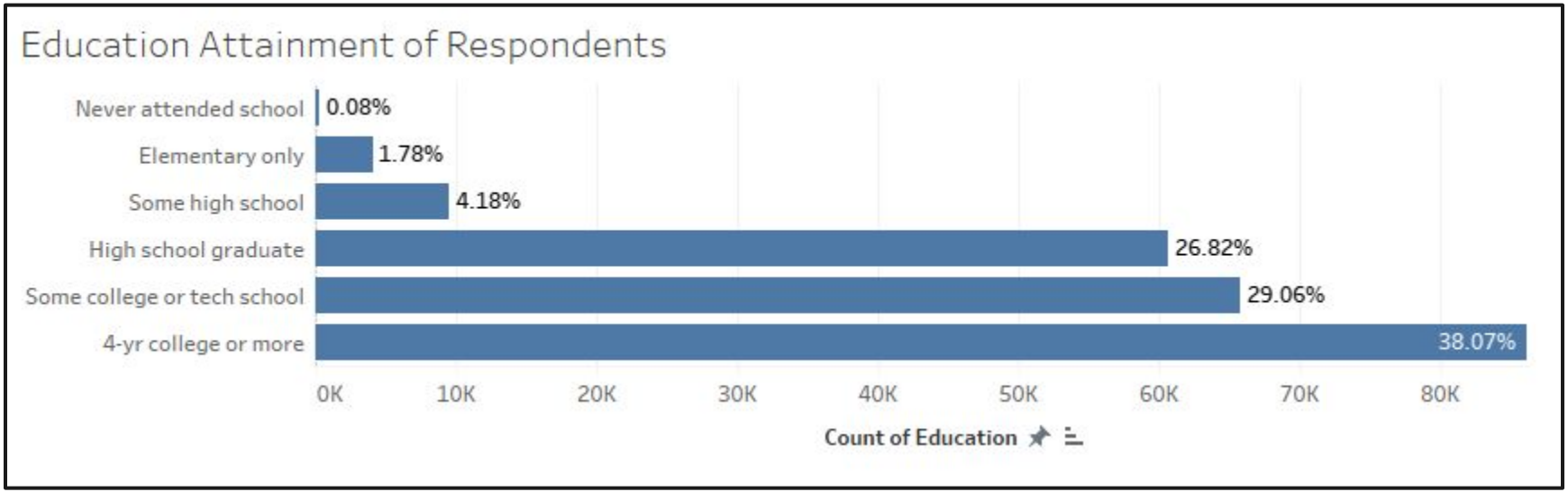
Source: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset?select=diabetes_binary_health_indicators_BRFSS2015.csv

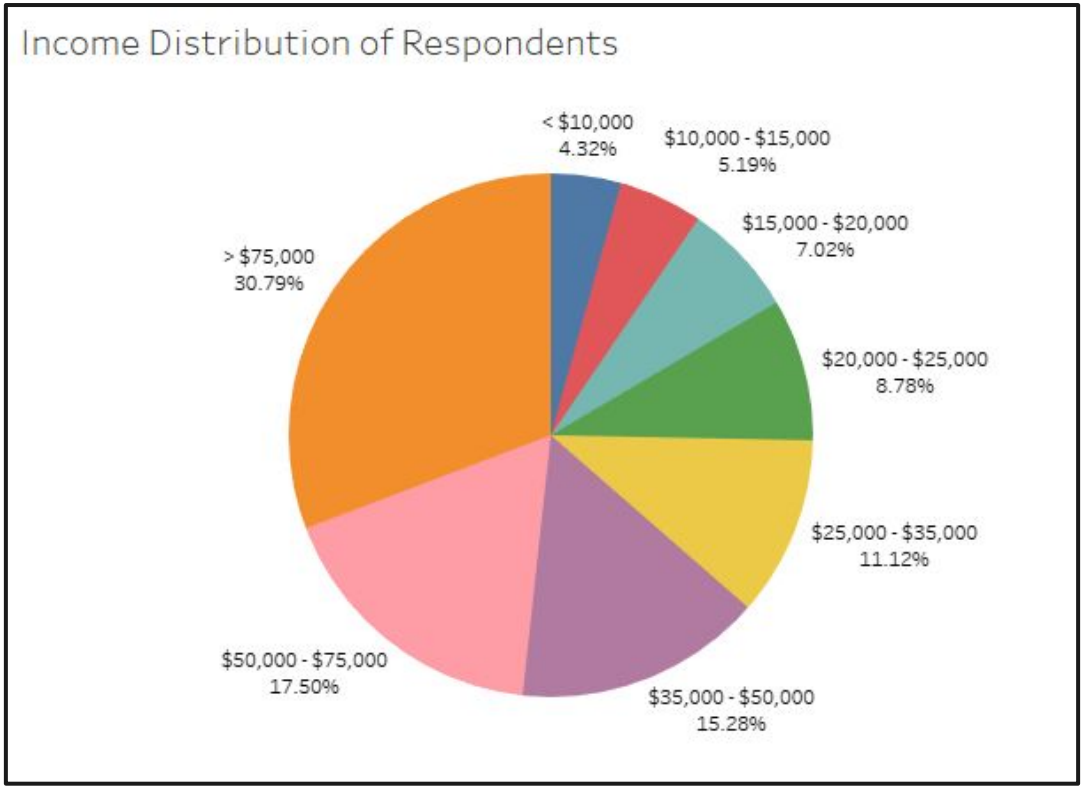
Tableau Visualizations

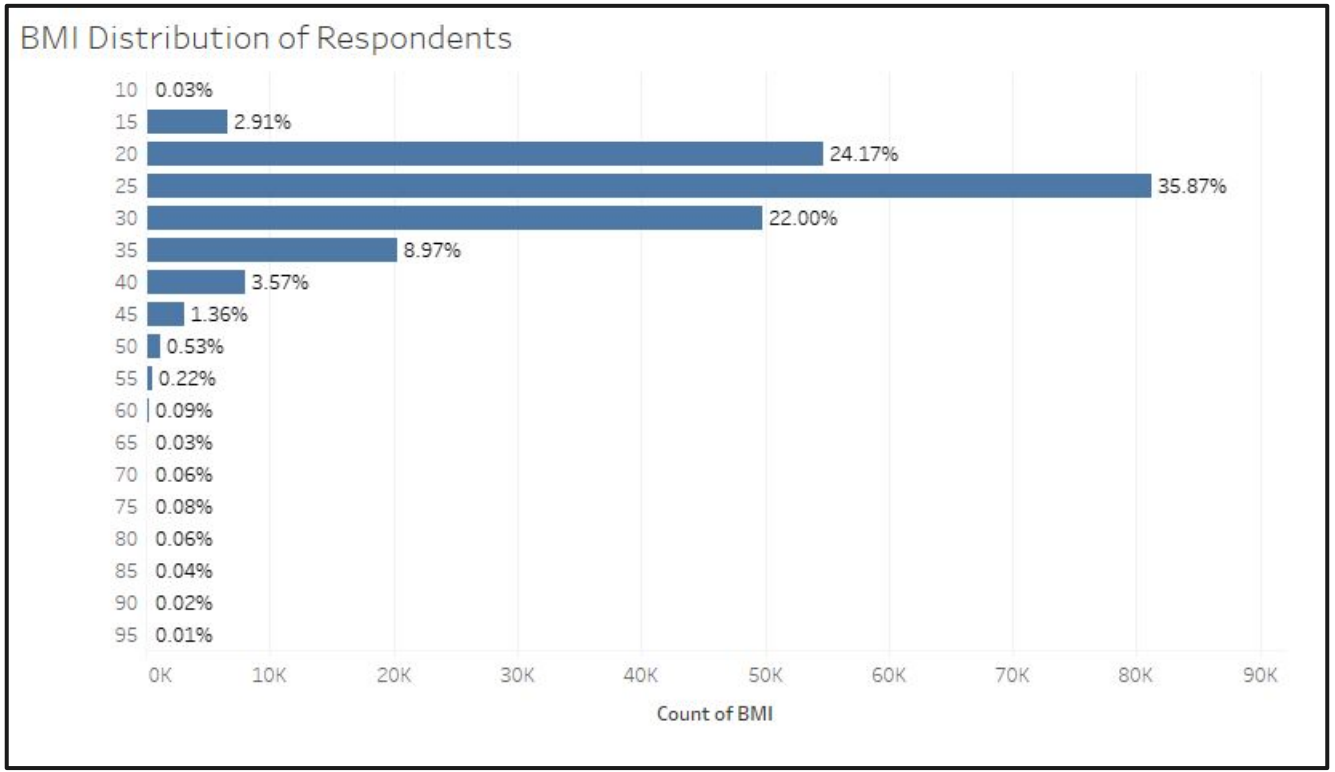

Target Variable Distribution
Prediabetic or Has Diabetes
# of Respondents: 33,531
14.81%
85.19%
No Diabetes
# of Respondents: 192,811


Sex Distribution of Respondents
F
43.65%
M
56.35%


Age Distribution of Respondents
18-24
25-30
31-35
36-40
41-45
46-50
51-55
56-60
61-65
66-70
71-75
76-80
80+
2.43%
3.11%
4.42%
5.38%
6.17%
7.59%
10.10%
11.88%
12.87%
12.53%
9.51%
6.69%
7.33%
0K
2K
4K
6K
8K
10K
12K
14K
16K
18K
20K
22K
24K
26K
28K
30K
32K
Count of Age


Education Attainment of Respondents
Never attended school
0.08%
Elementary only
1.78%
Some high school
4.18%
High school graduate
26.82%
Some college or tech school
29.06%
4-yr college or more
38.07%
0K
10K
20K
30K
40K
50K
60K
70K
80K
Count of Education


Income Distribution of Respondents
< $10,000
4.32%
$10,000 - $15,000
5.19%
$15,000 - $20,000
7.02%
$20,000 - $25,000
8.78%
$25,000 - $35,000
11.12%
$35,000 - $50,000
15.28%
$50,000 - $75,000
17.50%
> $75,000
30.79%


BMI Distribution of Respondents
10 0.03%
15 2.91%
20 24.17%
25 35.87%
30 22.00%
35 8.97%
40 3.57%
45 1.36%
50 0.53%
55 0.22%
60 0.09%
65 0.03%
70 0.06%
75 0.08%
80 0.06%
85 0.04%
90 0.02%
95 0.01%
0K 10K 20K 30K 40K 50K 60K 70K 80K 90K
Count of BMI

Data cleaning steps

- Check for nulls and remove rows that include nulls
  - Our dataset did not have null values
- Check for and remove duplicate entries
  - 24,206 duplicate entries were removed (including target variable)
  - 1,556 duplicate entries were removed (excluding target variable)
- 226,342 entries remained after data cleaning

Algorithms used

- Logistic Regression
- Random Forest Classifier
- AdaBoost Classifier

Libraries used

- Tableau: visualizations
- Pandas: data manipulation
- Sklearn: ML algorithms
- Pickle / Streamlit: app

Tuning the models

```
def model_tester(model, X, y):
    X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
    clf = model.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    print(classification_report(y_test, y_pred))
    print(f'Training Score: {clf.score(X_train, y_train)}')
    print(f'Testing Score: {clf.score(X_test, y_test)}')
```

```
# Look at different Logistic Regression models and find bester performing for further tuning
model_tester(LogisticRegression(random_state=42), X, y)
model_tester(LogisticRegression(random_state=42, max_iter=500), X, y)
model_tester(LogisticRegression(random_state=42, max_iter=1000), X, y)
model_tester(LogisticRegression(random_state=42, max_iter=10000), X, y)
```

Logistic Regression

- Evaluated different values for max_iter

Random Forest Classifier

- Evaluated different values for n_estimators and bootstrap setting

AdaBoost Classifier

- Evaluated different values for n_estimators and learning_rate

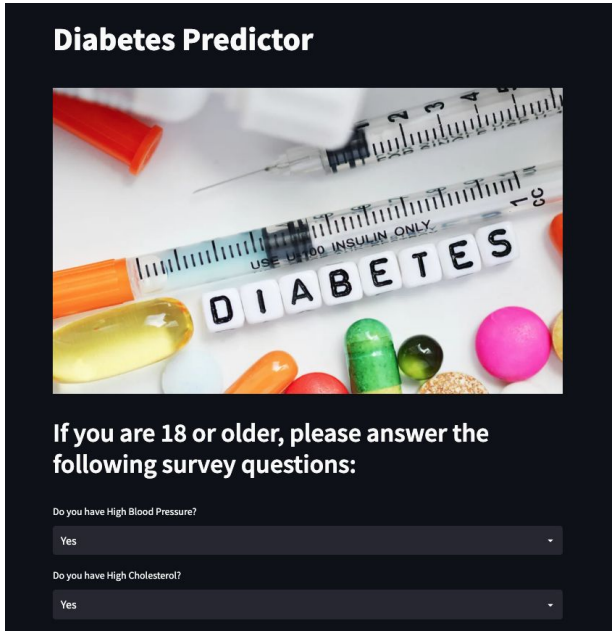
Overall

- Looked at results using dataset with non-binary and non-binned features scaled
- Looked at results using unscaled dataset

The Best Model - AdaBoost Classifier

- Unscaled dataset
- Hyperparameters
  - n_estimators = 1000
  - Learning_rate = 0.1
- Results
  - Training Score: 0.85691
  - Testing Score: 0.85848
  - Recall: 0.97
- Future Work
  - Use a balanced dataset. Our target variable was unbalanced towards the non-diabetic result (85-15).
  - See if XGBoost performs better than AdaBoost.

Web App made with Streamlit



Questions?