PREDICTIVE ANALYSIS FOR DIABETES

Orla Giffen, Anny She, Kyle Fujii, Ipsha Pandey,
Frances Hogg, Elizabeth Iskander, Mani Atwal

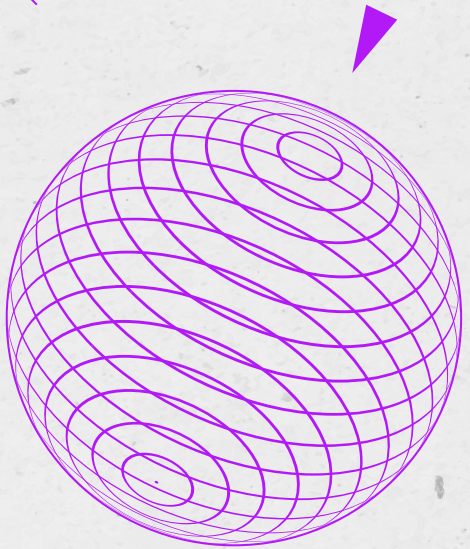


TABLE OF CONTENTS

01

INTRODUCTION

Introduction

Topic

- Relationship between certain factors (healthcare stats and lifestyle choices) and diabetes diagnosis

Goals

- Tool for diabetes diagnosis
- Understand which variables and lifestyle choices affect diagnosis

Stakeholders

- Individuals
- Insurance companies
- Pharmaceutical industry
- Government Health Agencies

Data

- CDC Diabetes Health Indicators

02

DATASET

Dataset

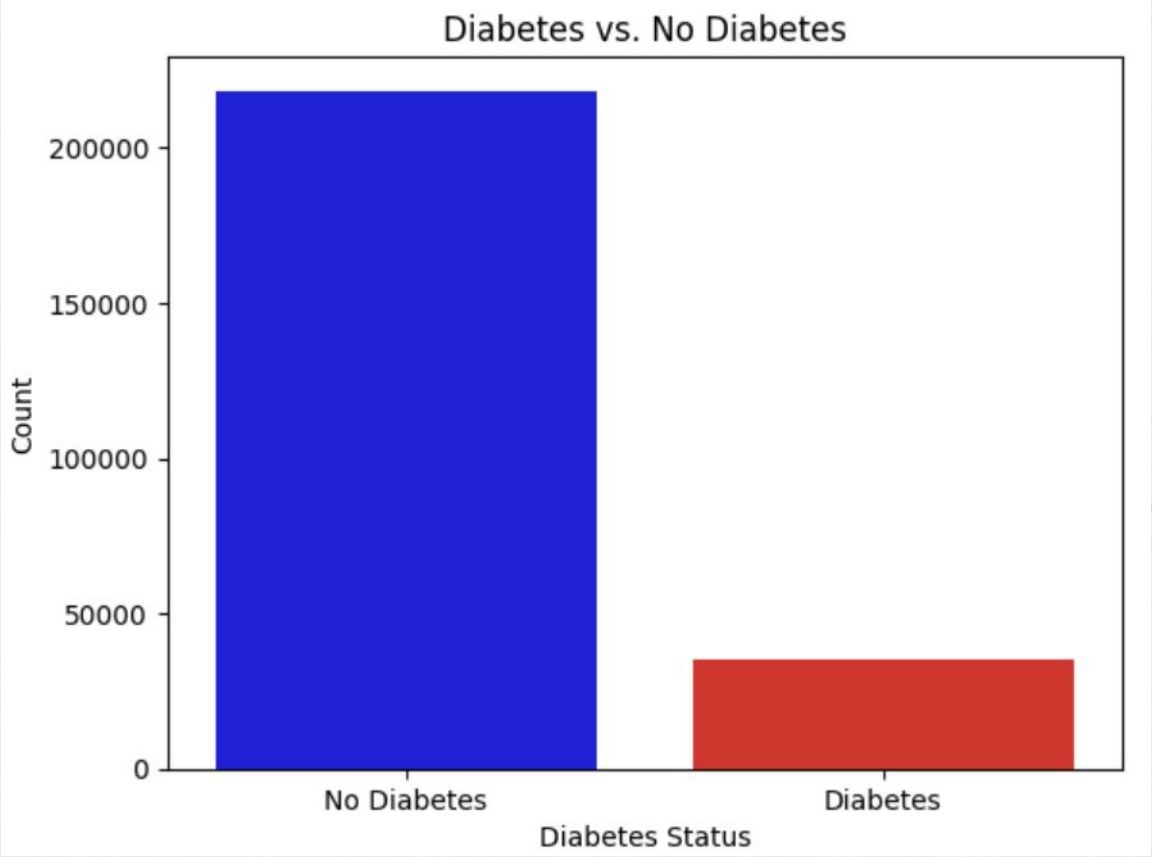
Origin/Collection — Created by CDC to better understand how lifestyle relates to diabetes in the US

Features — 21 features; no missing values

EDA — Understanding the dataset prior to modelling

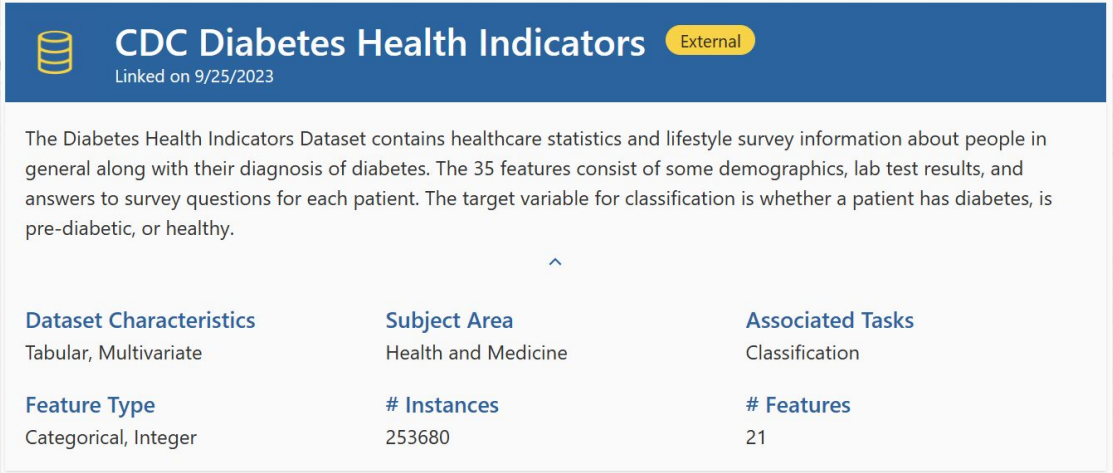
Feature Selection/ Engineering — Improve performance, reduce overfitting

Issues — Class imbalance


Diabetes vs. No Diabetes
200000
150000
100000
50000
0
Count
No Diabetes
Diabetes
Diabetes Status

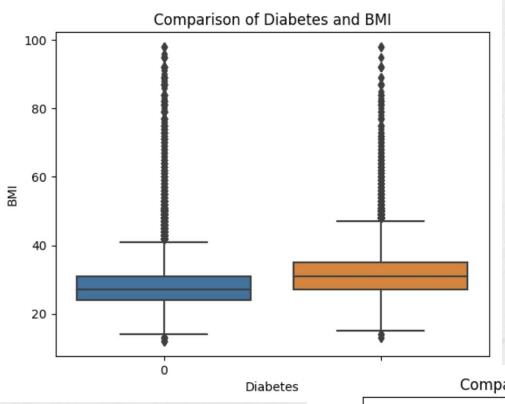
CDC Data

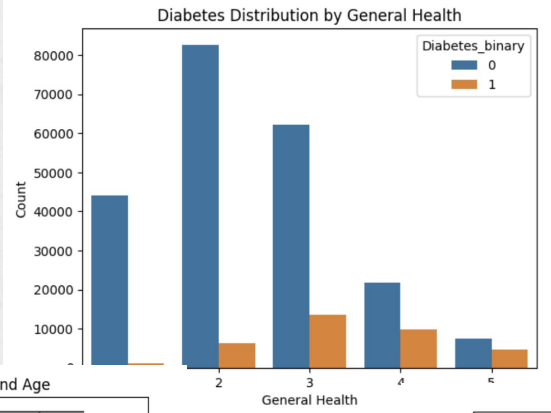
- Number of instances
- Features of dataset
- Purpose of dataset

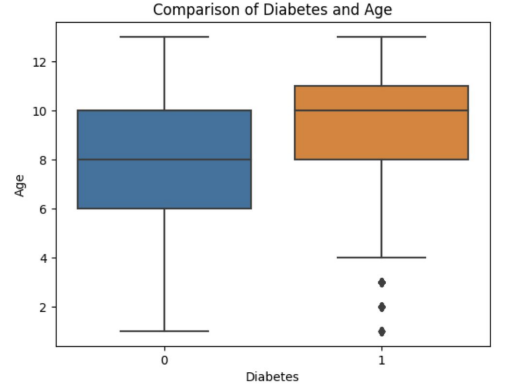


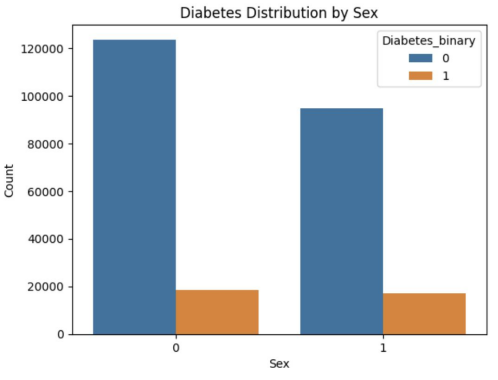
Data Exploration

Visualizations

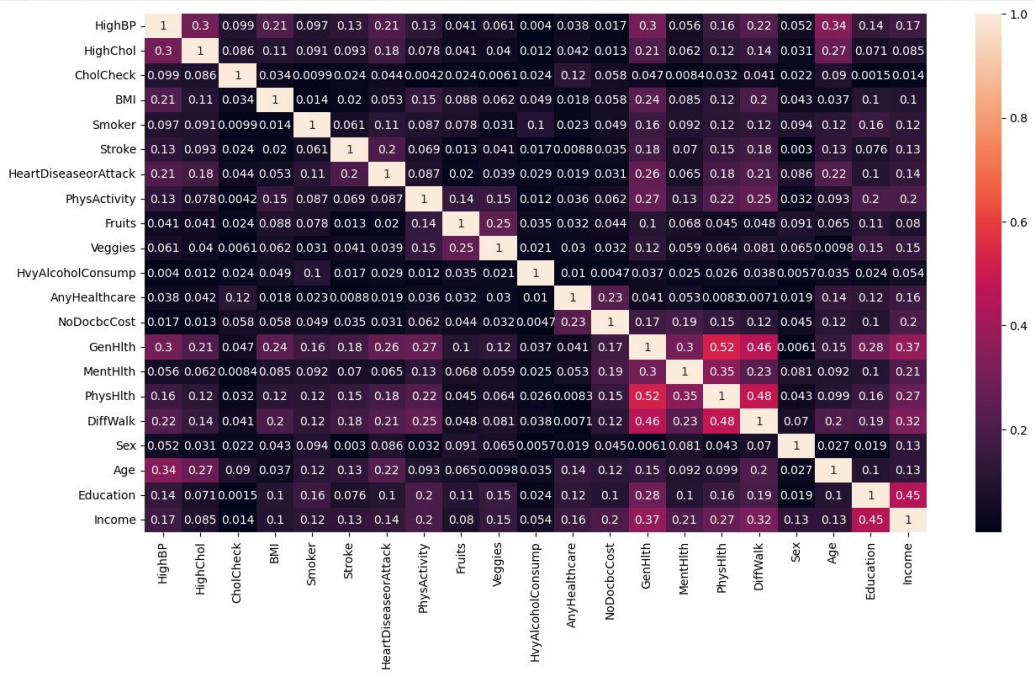








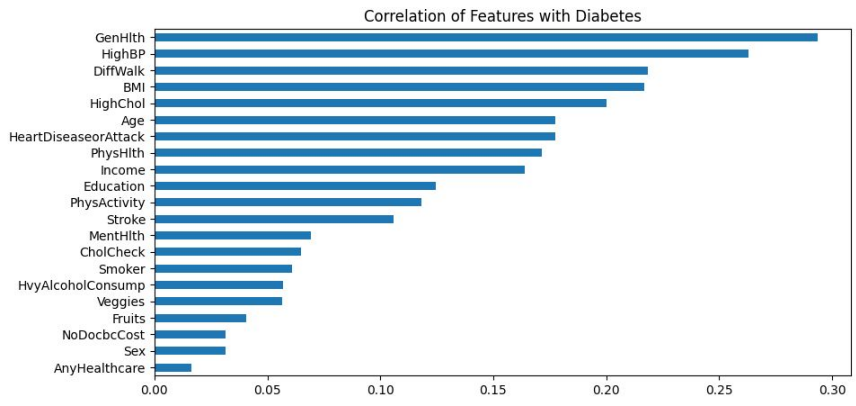
Feature Correlation



- Multicollinearity: Some features are highly correlated with each other
- GenHlth and PhysHlth, GenHlth and DiffWalk, and DiffWalk and Physhealth
- Feature engineering? Use a different model?

Initial Feature Selection

- Graphing absolute correlation of the features with diabetes
- Some features have very low correlation
  - AnyHealthcare
  - Sex
  - NoDocbcCost
  - Fruits



Pre-Processing

- **Class Imbalance:**
  - Much higher proportion of samples in the dataset that did not have diabetes
- **Dropping Features:**
  - Features with low correlation were dropped to reduce overfitting
- **Train/Test Split**
- **Scaling:**
  - Scaled the training and test data

```
# Checking for class imbalance
class_distribution = y['Diabetes_binary'].value_counts(normalize=True)
print(class_distribution)
```

```
0    0.860667
1    0.139333
Name: Diabetes_binary,
```

```
# Dealing with imbalance by oversampling
smote = SMOTE(random_state=42)
X, y = smote.fit_resample(X, y)
```

```
# Scaling the data
scalar = StandardScaler()
X_train = scalar.fit_transform(X_train)
X_test = scalar.transform(X_test)
```

```
#Dropping the variables with low correlation with diabetes
X.drop(columns=['AnyHealthcare', 'Sex', 'NoDocbcCost', 'Fruits'], inplace=True, axis=1)
```

03

METHODS

Models We Tested

1	Logistic Regression	LR model with a max iteration of 1500
2	KNNs	KNNs classifier with 5 clusters
3	Decision Tree	DT with max depth of 15
4	Random Forest	RF with default parameters
5	Keras Nearest Network	Keras NN with chosen parameters

Evaluation on Test Set

Accuracy

How correct our predictions are compared to the truth

- **Logistic Regression:**
  - Accuracy: 0.73655
- **KNNs:**
  - Accuracy: 0.78447
- **Decision Tree:**
  - Accuracy: 0.76424
- **Random Forest:**
  - Accuracy: 0.85886
- **Keras NN:**
  - Accuracy: 0.75114

Recall

Used to decrease the probability of false negative

- **Logistic Regression:**
  - Recall: 0.754597
- **KNNs:**
  - Recall: 0.86421
- **Decision Tree:**
  - Recall: 0.83171
- **Random Forest:**
  - Recall: 0.90882
- **Keras NN:**
  - Recall: 0.81525

Model Selection

Random Forest

- Had the best results:
  Highest accuracy and recall
- Limits Overfitting
- Moving forward with perfecting this model
  - Better feature engineering/selection
  - Parameter tuning

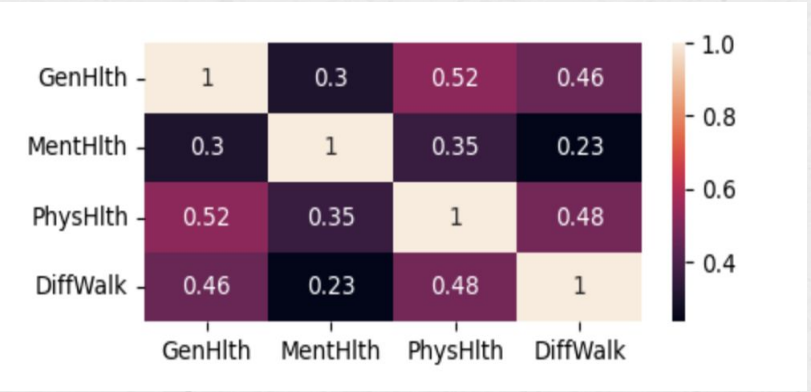
Random Forest

```
1  rf_classifier = RandomForestClassifier(random_state=42)
2  rf_classifier.fit(X_train, y_train)
3  rf_y_pred = rf_classifier.predict(X_test)
```

```
Training Accuracy: 0.9754046270904063
Training Recall: 0.9820796118393573
Test Accuracy: 0.8588636727963909
Test Recall: 0.9088175202589472
```

Perfecting the Model (Attempt 1)

- Selected only the 10 most correlated features to use
- Looked at features with high collinearity
  - Physical health and general health with correlation of .52
  - Both correlated with “Difficulty Walking” at ~.47
- Attempted to remove columns due to multicollinearity
  - Hope to reduce overfitting
- Not bad, but not as good as the original model




```
Training Accuracy: 0.8933771118757406
Training Recall: 0.9088818066863875
Test Accuracy: 0.8161311745711864
Test Recall: 0.8558573035513418
```

Perfecting the Model (Attempt 2)

- Standardized the non-binary variables
- Combined correlated variables instead of removing them
  - Averaged general health, physical health, and difficulty walking
- Did slightly better than our original attempt

```
scaler = StandardScaler()
columns_to_scale = ['BMI', 'GenHlth', 'MentHlth', 'PhysHlth', 'Age', 'Education', 'Income']
df[columns_to_scale] = scaler.fit_transform(df[columns_to_scale])
```

```
# Combine variables with high colinearity: GenHlth, PhysHlth, DiffWalk
df["Gen+PhysHlth+DiffWalk"] = (df['GenHlth'] + df['PhysHlth'] + df['DiffWalk'])/3
df = df.drop(['PhysHlth', "GenHlth", 'DiffWalk'], axis=1)
```

```
Training Accuracy: 0.9754046270904063
Training Recall: 0.9820796118393573
Test Accuracy: 0.8588636727963909
Test Recall: 0.9088175202589472
```

Original

```
Training Accuracy: 0.9919675725809683
Training Recall: 0.988739679469941
Test Accuracy: 0.9079739849314128
Test Recall: 0.9106310690755492
```

New Attempt

04

RESULTS

RESULTS

- Our best model had:
  - 90.797% Accuracy
  - 91.063% Recall
- The health features that were most correlated with diabetes were:
  - General Health, High Blood Pressure, Difficulty Walking, BMI, High Cholesterol
- Lifestyle features (negatively) correlated with diabetes included:
  - Vegetables, Education, Physical Activity, and Income
- Age also had a high positive correlation
  - More likely to have diabetes as you age
  - Should get tested more often

```
Training Accuracy: 0.9919675725809683
Training Recall: 0.988739679469941
Test Accuracy: 0.9079739849314128
Test Recall: 0.9106310690755492
```

Conclusions

Implications

- **Medical**
  - Machine Learning can be helpful in medical settings
  - Concerns:
    - Issues with bias and which type of risks to favor
    - Risk assessment tool, not a diagnosis tool
- **Insurance Market**
  - With more patient risk information known, less adverse selection in the insurance market
  - So, more fair pricing

Real-world usage

- **Individuals:**
  - Those with a family history of diabetes could use this model as a tool to determine their diabetes risk
- **Doctors/Hospitals:**
  - High-risk patients according to the model can be closely monitored
- **Insurance Companies:**
  - They can use the model to assess the risk associated with a client and properly charge them
- **CDC:**
  - Can use the lifestyle features correlated with diabetes to advise citizens on preventative actions

THANK YOU

CREDITS: This presentation template was created by **Slidesgo**,
including icons by **Flaticon** and infographics & images by **Freepik**