Unraveling the Diabetes Dilemma: Impact of Obesity and Inactivity

Akhil Kumar Reddy Bhoomi Reddy - 02080263
Shivakumar Pasem - 02077631
Ben George Samuel - 02078919
Aishwarya Ganesh Bhat - 02097374

The issues:

Within this report, we embark on an exploration of the intricate relationship between lifestyle factors, namely obesity and physical inactivity, and their profound impact on the prevalence of diabetes. The primary objective is to decipher the nuanced interplay of these factors on the stage of diabetes.

The Centers for Disease Control and Prevention (CDC) in the United States maintains an up-to-date database, which we have meticulously analyzed for data encompassing obesity and inactivity rates, along with the percentage of diabetics for each county in the US for the year 2018. Through sophisticated feature engineering and insightful data analysis, our aim is to uncover answers to fundamental questions:

- What lifestyle elements contribute significantly to the diabetes narrative?
- Do these lifestyle elements collectively exert an influence on diabetes?
- Is there a discernible correlation between obesity and inactivity?

Findings:

We examined 15 variables from the dataset including information such as the name of the county and state, Federal Information Processing System (FIPS) Codes, diabetic, obesity and inactivity rate of each of the counties for the year 2018. From these variables we were able to extract 354 data of counties with the complete information on its diabetic, inactivity and obesity rates.

Drawing from this data provided by the US CDC, a compelling revelation emerges: the inactivity rate demonstrates statistical significance nearly 1.5 times more than obesity in its correlation with the rise in diabetic rates. However, it is imperative to acknowledge the existence of a positive correlation between obesity and inactivity rates. Consequently, obesity remains a paramount concern that necessitates strategic attention.

Discussion:

According to the research, the obesity and inactivity rate of a county are crucial factors to understand the diabetic rate. However, the models that we created by these factors can predict only 40% of the variability in the diabetic rate. This disclosure underscores the complexity of health outcomes and

suggests that while obesity and inactivity are significant, there exist other contributing factors or nuanced dynamics that remain unexplained by the current models.

The acknowledgment of the models' limited predictive power opens avenues for further inquiry. Researchers may explore additional variables, consider interactions between different factors, or refine existing models to improve predictive accuracy. This understanding not only informs the ongoing research on the interplay between lifestyle choices and diabetes but also emphasizes the need for a holistic approach that considers a broader spectrum of influences on health outcomes within a county.

Appendix A: Method

The dataset utilized in this research encompassed information from 3143 counties, detailing their diabetic rates for the year 2018. The dataset was segregated into spreadsheets containing 1361 counties with inactivity rates and 364 counties with obesity rates. These spreadsheets were then uploaded to Google Colab and merged based on counties with data for all three factors. As part of feature engineering, three additional features—'Interaction,' '% OBESE_squared,' and '% INACTIVE_squared'—were derived to capture potential nonlinear relationships between the original features.

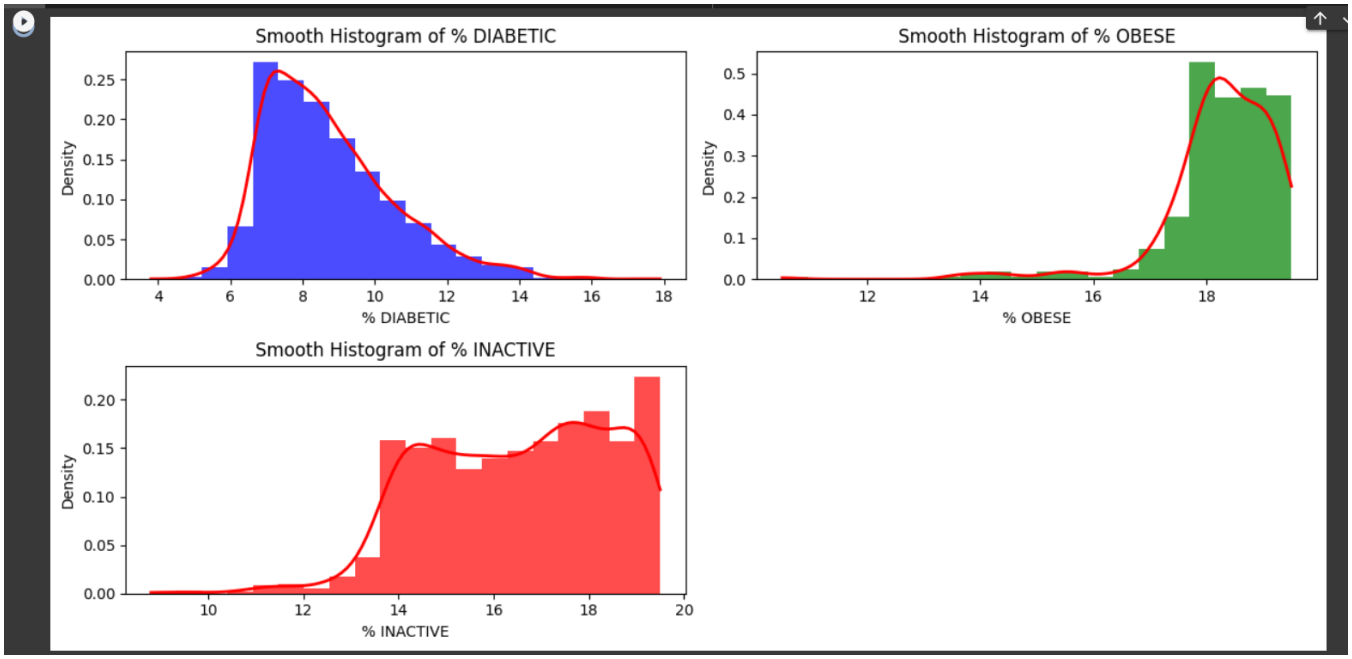
Feature engineering initiated with the creation of interaction terms (Interaction) and squared terms (% OBESE_squared, % INACTIVE_squared). These additions bolstered the feature set, enabling the model to comprehend potential nonlinear correlations. Subsequently, the dataset underwent division into training and testing sets using the train_test_split function, facilitating an evaluation of the model's generalization to new, unseen data. To ensure uniform feature scaling, a preprocessing step was applied, utilizing the StandardScaler() to maintain a consistent scale across all features and prevent any single feature from dominating the model.

The Ridge regression model was instantiated using the Ridge() class, incorporating an additional regularization term (L2 regularization) to prevent overfitting, making it suitable for scenarios with potentially collinear features. Hyperparameter tuning was executed through grid search (GridSearchCV) to identify the optimal value for the regularization parameter (alpha). The grid consisted of different alpha values, and the search was conducted with 5-fold cross-validation. The best alpha value and corresponding negative mean squared error from the grid search were then printed. The negative sign was used because GridSearchCV minimizes the scoring metric, and mean squared error is conventionally presented as a positive value. The best Ridge model, along with the optimal alpha, was obtained, and predictions were made on the testing set. The code concluded by calculating and printing the Mean Squared Error (MSE) and R-squared values to evaluate the model's performance on the testing set.

Following a similar methodology, a polynomial regression model with Ridge regularization was orchestrated, and its performance was assessed using cross-validation and testing data. The process began with the creation of polynomial features using PolynomialFeatures with a specified degree of 2, capturing cubic relationships between the input features. A pipeline was established, comprising a numerical preprocessor, the polynomial transformer, and the Ridge regression model. The alpha parameter, governing the regularization strength, was set to 100, with an option for adjustment. The pipeline underwent 5-fold cross-validation using cross_val_score, with the mean squared error (MSE) as

the evaluation metric. Negative MSE scores were averaged to compute the mean MSE across folds, indicating the model's performance during cross-validation. Following cross-validation, the model was fitted to the training data. Subsequently, predictions were made on the testing set, and the mean squared error on the testing set was calculated using the mean_squared_error function. The final step involved computing the R-squared value (r2_score) to assess how well the model explained the variance in the testing data.

Appendix B: Results :



Before combining the data:

Skewness of % DIABETIC: 0.97
Kurtosis of % DIABETIC: 1.03

Skewness of % OBESE: -2.69
Kurtosis of % OBESE: 12.32

Skewness of % INACTIVE: -0.34
Kurtosis of % INACTIVE: -0.55

% DIABETIC:

- Skewness: 0.97

Interpretation: The skewness value of 0.97 indicates that the distribution of % DIABETIC data is moderately right-skewed. this means that there is a slight tail on the right side of the distribution, and most data points are concentrated towards the lower end of the scale. It suggests that there may be more data points with lower values for % DIABETIC.

- Kurtosis: 1.03:

Interpretation: The kurtosis value of 1.03 implies that the % DIABETIC data has slightly heavier tails and a slightly less pronounced peak compared to a normal distribution (kurtosis of 3). This suggests that while there may be some outliers, the distribution is relatively close to a normal distribution in terms of tailedness and peakedness.

% OBESE:

- Skewness: -2.69

Interpretation: The skewness value of -2.69 indicates that the distribution of % OBESE data is heavily left-skewed. this means that there is a long tail on the left side of the distribution, and most data points are clustered towards the higher end of the scale. It suggests that there may be a larger number of lower values

- Kurtosis: 12.32

Interpretation: The kurtosis value of 12.32 implies that the % OBESE data has very heavy tails and a pronounced peak. This high kurtosis indicates that the distribution has more outliers or extreme values than a typical normal distribution (which has a kurtosis of 3). The distribution is leptokurtic, meaning it has heavier tails and is more peaked than a normal distribution.

% INACTIVE:

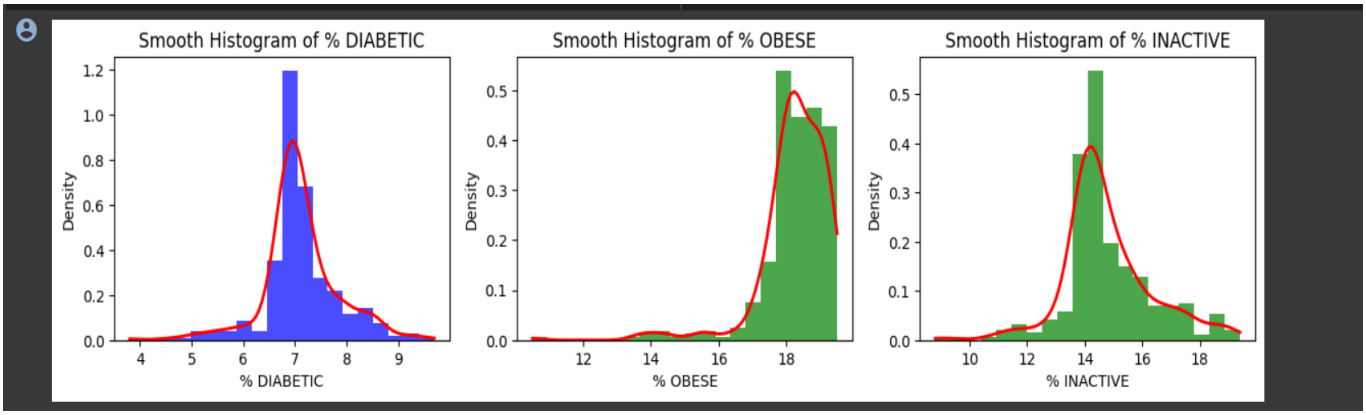
- Skewness: -0.34

Interpretation: The skewness value of -0.34 suggests that the distribution of % INACTIVE data is slightly left-skewed. this means that there is a slight tail on the left side of the distribution, but the majority of data points are concentrated towards the higher end of the scale. It indicates a tendency for more values to be on the higher side of the scale.

- Kurtosis: -0.55

Interpretation: The kurtosis value of -0.55 indicates that the % INACTIVE data has lighter tails and a less pronounced peak compared to a normal distribution (kurtosis of 3). This suggests that the distribution is relatively flatter and has fewer outliers compared to a typical normal distribution.

After combining the data:



Results from Histogram:

For % DIABETIC, it is moderately right-skewed with a distribution relatively close to normal. For % OBESE, it is heavily left-skewed with very heavy tails and a pronounced peak. For % INACTIVE, it is slightly left-skewed.

- Skewness of % OBESE: -2.75

A negative skewness value (like -2.75) suggests that the data is left-skewed or negatively skewed. In a negatively skewed distribution, the tail on the left side is longer or fatter than the right side, and the majority of the data points are concentrated on the right side.

- Kurtosis of % OBESE: 12.93

Kurtosis measures the "tailedness" or the concentration of data points in the tails of the distribution. A high positive kurtosis value (like 12.93) indicates that the data has heavy tails and is more peaked around the mean compared to a normal distribution. This suggests that the distribution has more extreme values (outliers) than a normal distribution.

- Skewness of % INACTIVE: 0.43

Skewness measures the asymmetry of the distribution of data.
A positive skewness value (like 0.43) suggests that the data is right-skewed or positively skewed. In a positively skewed distribution, the tail on the right side is longer or fatter than the left side, and the majority of the data points are concentrated on the left side.

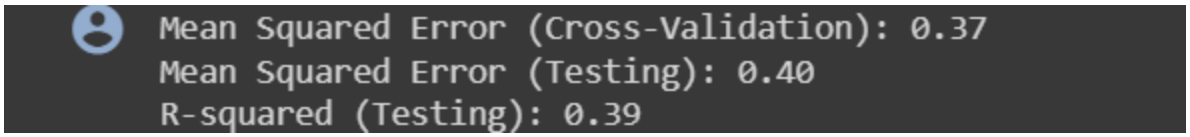
- Kurtosis of % INACTIVE: 1.61

Kurtosis measures the "tailedness" or the concentration of data points in the tails of the distribution. A positive kurtosis value (like 1.61) indicates that the data has slightly heavier tails than a normal distribution but is less peaked compared to the distribution of % OBESE.

The negative skewness of % OBESE suggests that it has a left-skewed distribution with a longer left tail. Additionally, the high positive kurtosis of % OBESE indicates that it has heavy tails and is more peaked. On the other hand, % INACTIVE is right-skewed but has slightly lighter tails and is less peaked compared to % OBESE.

```
Best Alpha: 0.1
Best Negative Mean Squared Error: 0.32
Mean Squared Error (Testing): 0.43
R-squared (Testing): 0.36
```

We found the best alpha value using Regularization. The grid search determined that the best regularization parameter (alpha) for the Ridge Regression model is 0.1. This value is selected based on cross-validation and is used to control the strength of regularization. Smaller alpha values indicate less regularization.



The provided results indicate the performance of a predictive model, and we can use them to assess whether the model exhibits bias or variance.

1. Mean Squared Error (Cross-Validation): 0.37

A MSE of 0.37 suggests that, on average, the model's predictions are relatively close to the true values in the cross-validation dataset. Lower MSE values generally indicate lower bias in the model because it is fitting the data well.

2. Mean Squared Error (Testing): 0.40

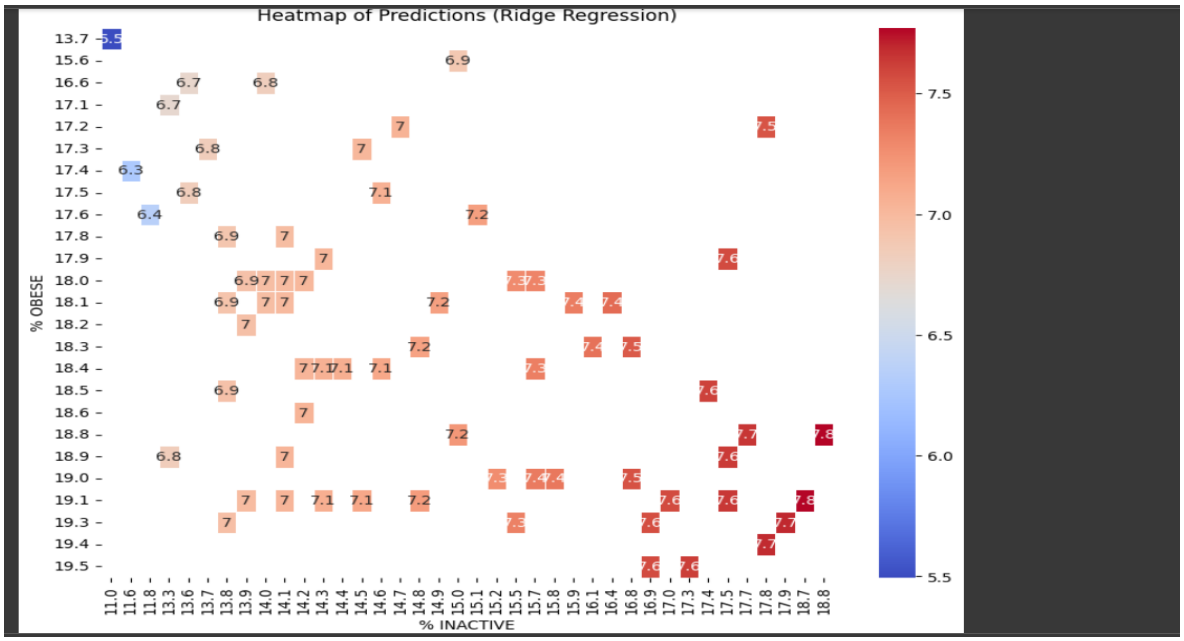
A testing MSE of 0.40 indicates that the model's performance is consistent with the cross-validation results, meaning it generalizes well to unseen data. Again, lower values indicate lower bias.

3. R-squared (Testing): 0.39

- The R-squared value (also known as the coefficient of determination) measures the proportion of the variance in the dependent variable that is explained by the model. It ranges from 0 to 1, where higher values indicate a better fit.
- An R-squared of 0.39 suggests that the model explains approximately 39% of the variance in the testing dataset. While this indicates some ability to capture the underlying patterns, it also suggests that there is still unexplained variance, which could be due to model bias or noise.

Conclusion:

- The model appears to have relatively low bias because it performs well both in cross-validation and on the testing dataset, as indicated by the low MSE values.
- However, there is still some unexplained variance in the testing dataset, as indicated by the R-squared value of 0.39. This could be due to either model variance or the inherent noise in the data.



From the heat map we can clearly see that if %INACTIVE is more than 16.1 and %OBESE is more than 18.1, then 7.5% of people will be diabetic.

Appendix C: Code

```
# Load data from each sheet into separate DataFrames
xls = pd.ExcelFile('cdc-diabetes-2018.xlsx')
sheet_names = xls.sheet_names  # Get the names of all sheets

# Perform inner joins on the 'FIPS' column to combine the DataFrames
combined_data = pd.merge(dfs[0], dfs[1], on='FIPS', how='inner')
combined_data = pd.merge(combined_data, dfs[2], on='FIPS', how='inner')

# Calculate skewness and kurtosis for each variable
skewness_obese = skew(df['% OBESE'])
kurtosis_obese = kurtosis(df['% OBESE'])

skewness_inactive = skew(df['% INACTIVE'])
kurtosis_inactive = kurtosis(df['% INACTIVE'])

# Print the results
print(f'Skewness of % OBESE: {skewness_obese:.2f}')
print(f'Kurtosis of % OBESE: {kurtosis_obese:.2f}')
print(f'Skewness of % INACTIVE: {skewness_inactive:.2f}')
print(f'Kurtosis of % INACTIVE: {kurtosis_inactive:.2f}')
```

Skewness of % OBESE: -2.75
Kurtosis of % OBESE: 12.93
Skewness of % INACTIVE: 0.43
Kurtosis of % INACTIVE: 1.61

```
# Feature Engineering
df['Interaction'] = df['% OBESE'] * df['% INACTIVE']
df['% OBESE_squared'] = df['% OBESE'] ** 2
df['% INACTIVE_squared'] = df['% INACTIVE'] ** 2

# Define dependent variable and independent variables
X = df[['% OBESE', '% INACTIVE', 'Interaction', '% OBESE_squared', '% INACTIVE_squared']]  # Add new features
y = df['% DIABETIC']  # Replace with the actual column name

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Preprocessing for numerical features (scaling)
numerical_transformer = Pipeline(steps=[
    ('scaler', StandardScaler())
])

# Create and fit the Ridge Regression model in a pipeline
ridge_model = Pipeline(steps=[
    ('preprocessor', numerical_transformer),
    ('model', Ridge(alpha=0.1))  # You can adjust the alpha parameter for regularization strength
])

# Perform cross-validation (5-fold cross-validation)
cv_scores = cross_val_score(ridge_model, X, y, cv=5, scoring='neg_mean_squared_error')
mse_cv = -cv_scores.mean()  # Calculate the mean MSE across folds
print(f"Mean Squared Error (Cross-Validation): {mse_cv:.2f}")

# Fit the model on the training data
ridge_model.fit(X_train, y_train)

# Make predictions on the testing set
y_pred = ridge_model.predict(X_test)

# Calculate the Mean Squared Error (MSE) on the testing set
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error (Testing): {mse:.2f}")
```

```
# Calculate the R-squared value
r2 = r2_score(y_test, y_pred)
print(f"R-squared (Testing): {r2:.2f}")
```

Mean Squared Error (Cross-Validation): 0.37
Mean Squared Error (Testing): 0.40
R-squared (Testing): 0.39

Link for Google collab for code reusability:
https://colab.research.google.com/drive/1K_TLlh7cFzXSj5p65u5iFlW5VtpPyODO#scrollTo=Gy4jKFNIn29e

References:

1. Reference: “An Introduction to Statistical Learning with Applications in Python”, Chapter 4.
2. Applied Logistic Regression, Hosmer & Lemeshow.