Obesity Risk Analysis and Prediction

TEAM:
Ricardo Gomez
Ricardo Luna
Elena Zamudio

CONTENTS

INTRO

STUDY CASE
DATA SET

PROJECT DEVELOPMENT

DATA PREPROCESSING
MODELING
MODEL EVALUATION
OPTIMIZATION

RESULTS

INTERPRETATION
VISUALIZATION

CONCLUSION

SUMMARY
KEY TAKE AWAYS

INTRODUCTION

STUDY CASE

There are 1 in 8 people in the world were living with obesity.

5 billion deaths linked to:

- Cardiovascular diseases
- Diabetes
- Cancers
- Neurological disorders
- Chronic respiratory diseases
- Digestive disorders.

A COMPLEX DISEASE...

Overweight and obesity result from an imbalance of energy intake (diet) and energy expenditure (physical activity).

Multifactorial condition: There are Over 200 factors can influence being overweight or obese some of them are:

- Body Characteristics (age, genetics, height, hormonal)
- Eating Habits
- Physical condition (exercise, sleep)
- Psychological and social factors
- Environment
- Etiological factors (diseases, immobilization, medications)

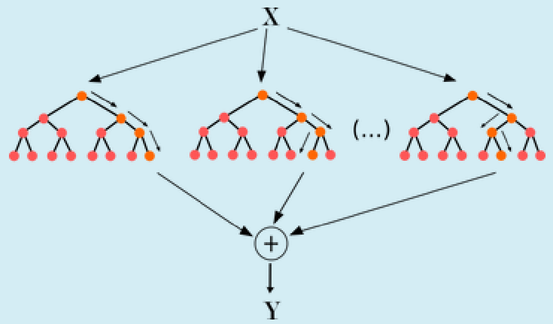
INTRODUCTION

STUDY CASE

APPLYING ML FOR OBESITY STUDY

- Analyzes Complexity: Captures interactions between genetics, diet, activity, and environment.
- Reveals Patterns: Identifies hidden predictors and correlations.
- Personalizes Interventions: Tailors treatments to individual profiles.
- Improves Predictions: Enhances accuracy for risk and outcomes.
- Informs Decisions: Supports effective policy and clinical strategies.

RANDOM FOREST CLASSIFIER & OBESITY



- Handles Various Data: Works with both categorical and continuous variables.
- Manages Missing Data: Reliable even with incomplete datasets.
- Captures Complexity: Identifies interactions between diet, activity, and genetics.
- High Accuracy: Provides reliable classification of obesity risk.
- Identifies Key Features: Ranks important predictors of obesity.
- Prevents Overfitting: Ensures generalizable, robust predictions.

INTRODUCTION

DATA SET

Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from Colombia, Peru and Mexico

*Data collected by online survey

kaggle

Description

- 2019
- 2,111 records
- 16 feature variables:
- 1 response: Obesity Level Deducted
    - Underweight
    - Normal weight
    - Overweight I
    - Overweight Type II
    - Obesity Type I
    - Obesity Type II
    - Obesity Type III

Demographics

1. Gender
2. **Age**
3. **Height**
4. **Weight**
5. Family History with Overweight

Eating Habits

1. FAVC: Frequent Consumption of High Caloric Food
2. **FCVC: Frequency of Consumption of Vegetables (per week)**
3. **NCP: Number of meals**
4. CAEC: Consumption of food between meals
5. Smoke or not

11. CH2O: Consumption of water daily
12. SCC: Calories consumption monitoring
13. CALC: Consumption of alcohol

Physical Condition

13. FAF: Physical activity frequency (times/week)

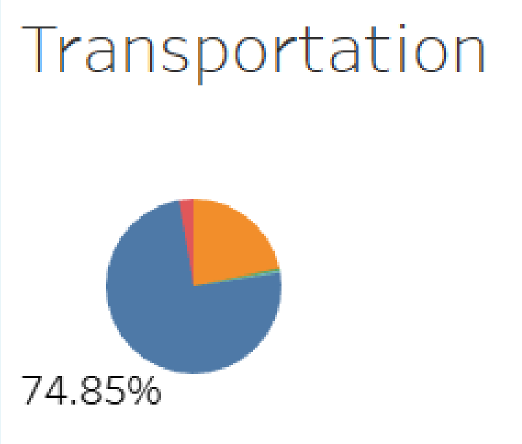
14. TUE: Time using technology devices (hr/day)

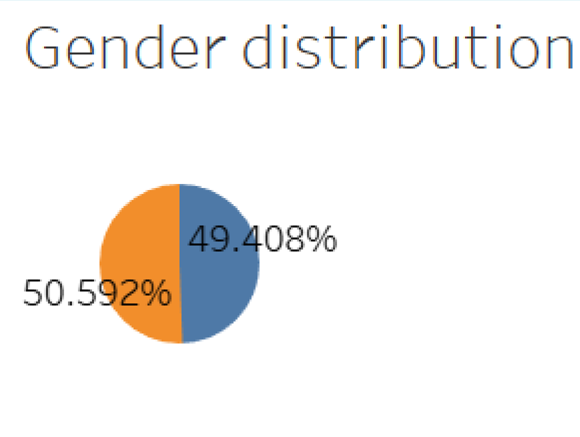
16. MTRANS: Transportation used

Continuous Categorical Boolean

PROJECT DEVELOPMENT

DATA PREPROCESSING





CHECK AND CLEAN DATA

No missing values

Gender 50/50.

Height around 1.6 and 1.8 m.

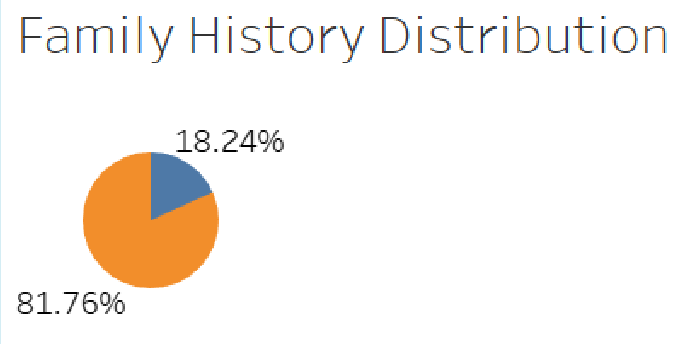
Most use public transportation.

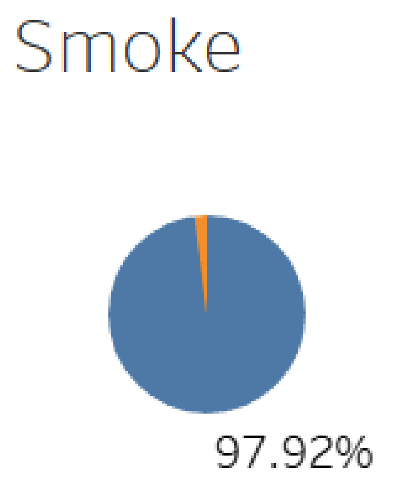
Most non smoking.

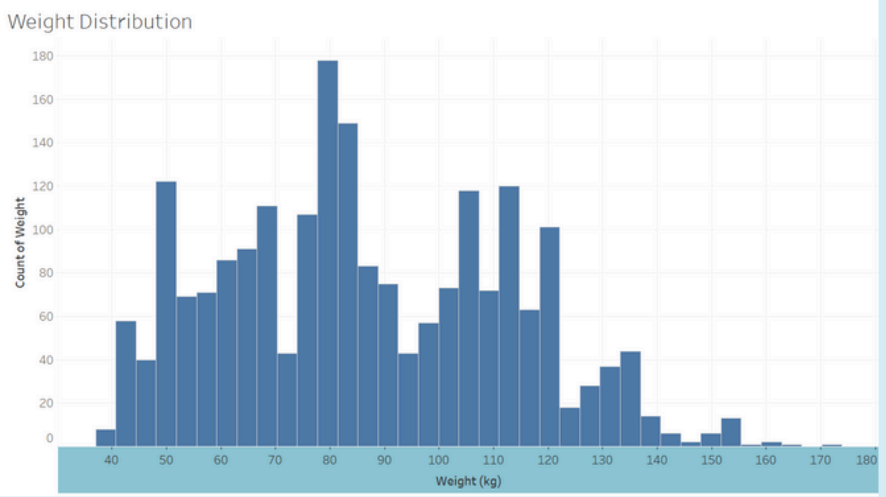
Most with family history of overweight.

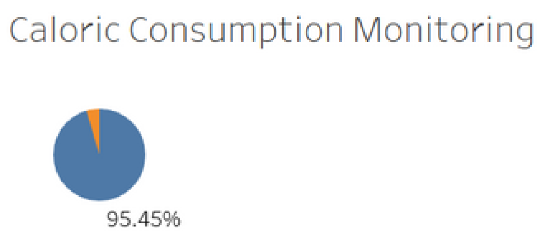
Most don’t monitor calories.

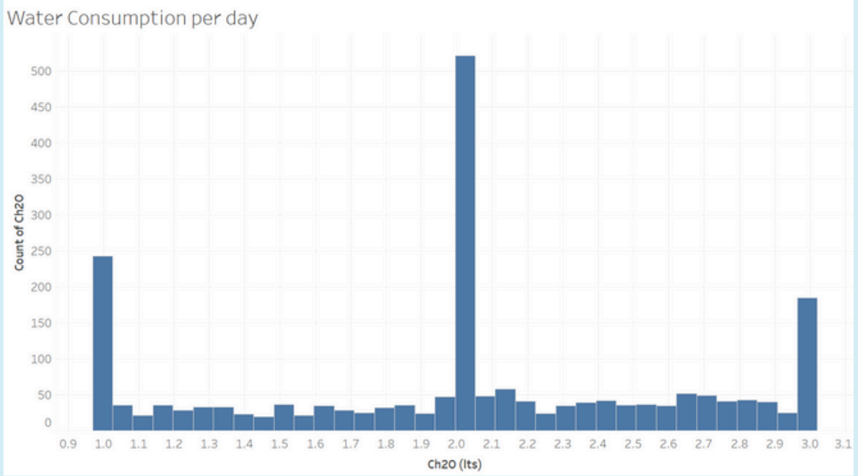
Similar quantity of subject by BMI Classification

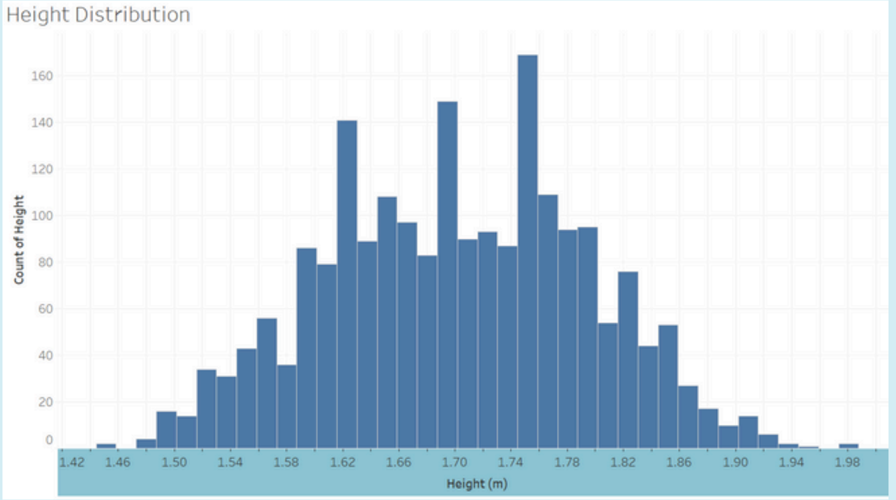


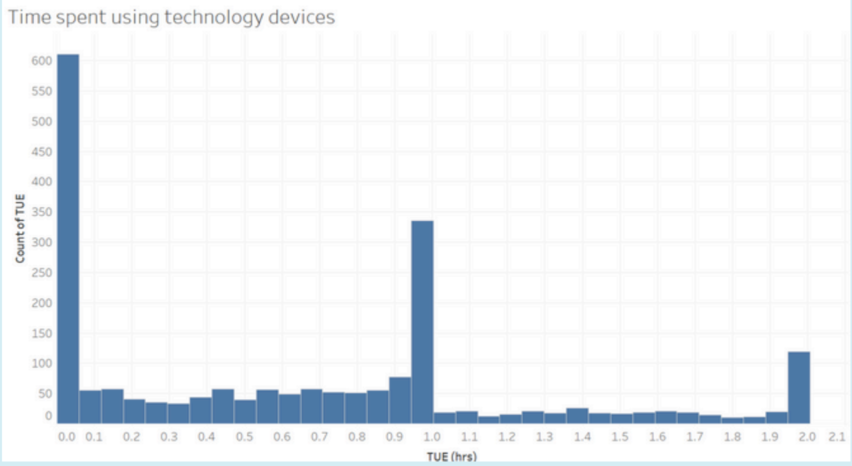


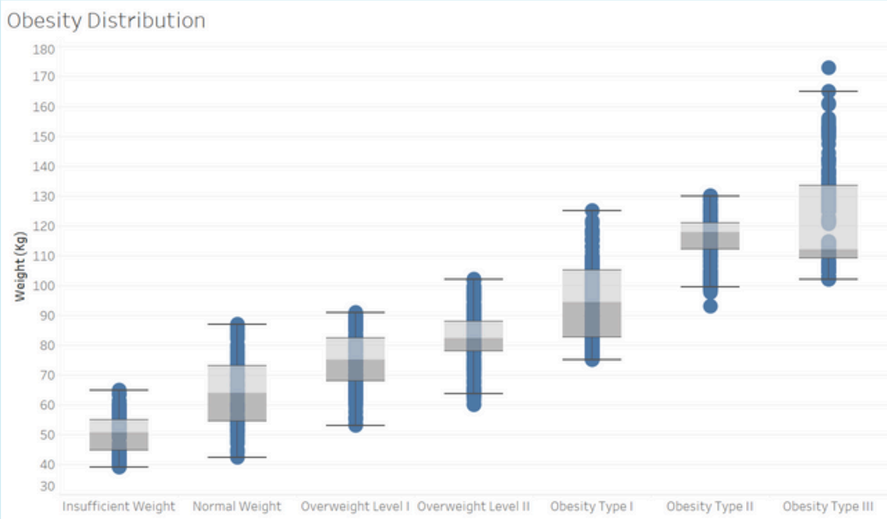












PROJECT DEVELOPMENT

DATA PREPROCESSING

Loading and Preprocessing Loans Encoded Data

```python
import findspark
findspark.init()
from pyspark.sql import SparkSession
import os
import pandas as pd

# Set environment variables
os.environ["JAVA_HOME"] = "C:/Program Files/Java/jdk-1.8"
os.environ["SPARK_HOME"] = "C:/Spark/spark-3.5.3-bin-hadoop3"

# Initialize Spark session
spark = SparkSession.builder.appName("ObesityData").getOrCreate()

# Load the new dataset
df_obesity = spark.read.csv("ObesityDataSet.csv", header=True, inferSchema=True)

# Show the first few rows of the PySpark DataFrame
df_obesity.show()
```

```python
# Show the first few rows of the PySpark DataFrame
df_obesity.show()

# Collect the data from the PySpark DataFrame into a list of rows
data = df_obesity.collect()

# Convert the data into a list of dictionaries (each dictionary corresponds to a row)
data_dict = [row.asDict() for row in data]

# Convert the list of dictionaries into a Pandas DataFrame
df_obesity_pandas = pd.DataFrame(data_dict)

df_obesity_pandas.head()
```

```
+------+----+------+------+------------------------------+----+----+---+----------+-----+----+---+---+---+----------+--------
------------+-------------------+
|Gender| Age|Height|Weight|family_history_with_overweight|FAVC|FCVC|NCP|      CAEC|SMOKE|CH2O|SCC|FAF|TUE|      CALC|
MTRANS|          NObeyesdad|
+------+----+------+------+------------------------------+----+----+---+----------+-----+----+---+---+---+----------+--------
------------+-------------------+
|Female|21.0|  1.62|  64.0|                           yes|  no| 2.0|3.0| Sometimes|   no| 2.0| no|0.0|1.0|        no|Public_T
```

```python
# Register the DataFrame as a temporary view
df_obesity.createOrReplaceTempView("obesity_data")

# Count total number of rows in the DataFrame with Spark SQL
query = """
SELECT COUNT(*) AS total_rows
FROM obesity_data
"""

# Execute the query and show the result
spark.sql(query).show()
```

```
+----------+
|total_rows|
+----------+
|      2111|
+----------+
```

PROJECT DEVELOPMENT

DATA PREPROCESSING

SCALING NUMERICAL DATA

```python
# Scale the numerical features
obesity_data_scaled = StandardScaler().fit_transform(X[["Age","Height","Weight","FCVC","NCP","CH2O","FAF","TUE"]])

# Create a DataFrame with the scaled data
df_obesity_transformed = pd.DataFrame(
    obesity_data_scaled, columns=["Age","Height","Weight","FCVC","NCP","CH2O","FAF","TUE"]
)

# Show the scaled data
df_obesity_transformed.head()
```

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE
0	-0.522124	-0.875589	-0.862558	-0.785019	0.404153	-0.013073	-1.188039	0.561997
1	-0.522124	-1.947599	-1.168077	1.088342	0.404153	1.618759	2.339750	-1.080625
2	-0.206889	1.054029	-0.366090	-0.785019	0.404153	-0.013073	1.163820	0.561997
3	0.423582	1.054029	0.015808	1.088342	0.404153	-0.013073	1.163820	-1.080625
4	-0.364507	0.839627	0.122740	-0.785019	-2.167023	-0.013073	-1.188039	-1.080625

+ CONCATENATE

GOT DUMMIES FOR CATEGORICAL AND BOOLEAN

```python
#One-hot encode categorical data
X = pd.get_dummies(X)
X.head()
```

	Age	Height	Weight	FCVC	NCP	CH2O	FAF	TUE	Gender_Female	Gender_Male	...	SCC_yes	CALC_Always	CALC_Frequently	CAL
0	21.0	1.62	64.0	2.0	3.0	2.0	0.0	1.0	True	False	...	False	False	False	
1	21.0	1.52	56.0	3.0	3.0	3.0	3.0	0.0	True	False	...	True	False	False	
2	23.0	1.80	77.0	2.0	3.0	2.0	2.0	1.0	False	True	...	False	False	True	
3	27.0	1.80	87.0	3.0	3.0	2.0	2.0	0.0	False	True	...	False	False	True	
4	22.0	1.78	89.8	2.0	1.0	2.0	0.0	0.0	False	True	...	False	False	False	

5 rows × 31 columns

PROCESSED DATAFRAME

PROJECT DEVELOPMENT

MODELING

SPLIT DATAFRAME INTO TRAINING AND TESTING SETS

```
# Splitting into Train and Test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=78)
```

Fitting the Random Forest Model

CREATE A RANDOM FOREST MODEL WITH 500 ESTIMATORS

```
# Create a random forest classifier
rf_model = RandomForestClassifier(n_estimators=500, random_state=78)
```

```
# Fitting the model
rf_model = rf_model.fit(X_train, y_train)
```

Making Predictions Using the Random Forest Model

GENERATE PREDICTIONS TO TEST MODEL

```
# Making predictions using the testing data
predictions = rf_model.predict(X_test)
```

RESULTS

MODEL EVALUATION

Random Forest model

- Performance Metrics:

Accuracy: 93.56%

- Strongest Class:

Obesity Type III (100% accuracy)
Average F1-Score: 94%

- Confusion Matrix:

Correct predictions dominate diagonal. Misclassifications are minimal, showing high reliability.

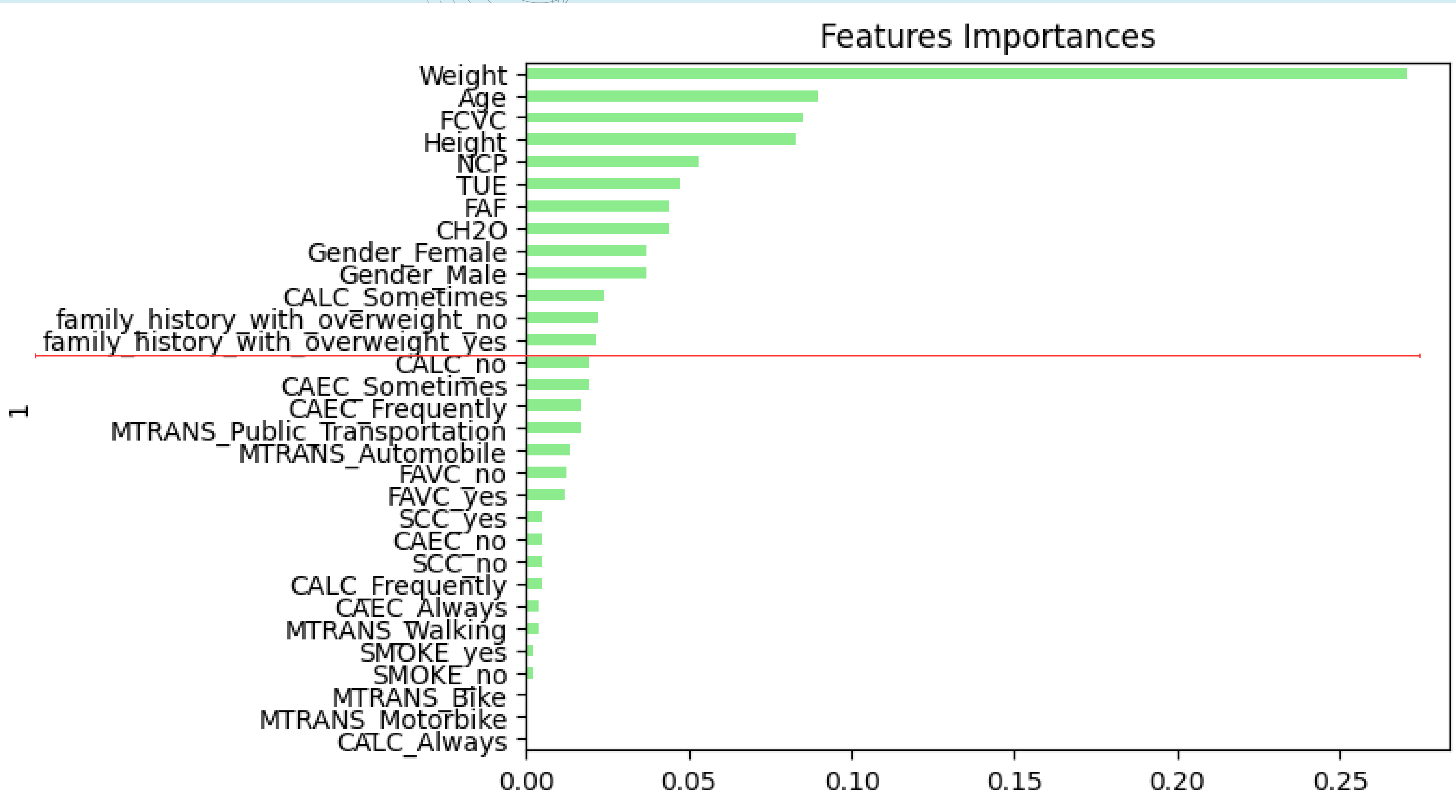
	Predicted 0	Predicted 1	Predicted 2	Predicted 3	Predicted 4	Predicted 5	Predicted 6
Actual 0	67	3	0	0	0	0	0
Actual 1	3	64	0	0	0	1	1
Actual 2	0	2	92	0	0	0	4
Actual 3	0	0	0	69	0	0	0
Actual 4	0	0	0	0	72	0	0
Actual 5	0	7	1	0	0	61	4
Actual 6	0	2	1	1	0	4	69

```
Accuracy Score : 0.9356060606060606
Classification Report
                     precision    recall  f1-score   support

Insufficient_Weight       0.96      0.96      0.96        70
      Normal_Weight       0.82      0.93      0.87        69
     Obesity_Type_I       0.98      0.94      0.96        98
    Obesity_Type_II       0.99      1.00      0.99        69
   Obesity_Type_III       1.00      1.00      1.00        72
 Overweight_Level_I       0.92      0.84      0.88        73
Overweight_Level_II       0.88      0.90      0.89        77

           accuracy                           0.94       528
          macro avg       0.94      0.94      0.94       528
       weighted avg       0.94      0.94      0.94       528
```

RESULTS

INTERPRETATION

Random Forest model



Number of meals is relevant

Liters of water per day is relevant

Time spent with electronic devices is relevant

Physical Activity Frequency is relevant

Height is relevant

Weight is relevant

Age is relevant

Most use public transportation.
= not relevant

Most non smoking.
= not relevant

Most with family history of overweight.
= relevant

Most don’t monitor calories.
= not relevant

Alcohol Consuption
= partialy not relevant

PROJECT DEVELOPTMENT

OPTIMIZATION

Random Forest model

- 2 Iterations for optimization:
- Different quantity of estimators
  - 1,000 first optimization
  - 500 second optimization
- Non relevant variables removed:
  - First optimization SCC, FAVC
  - Second optimization SMOKE
- Accuracy Results:
  - 94.12% (+0.5% increase)
  - 93.76% (-0.3% decrease)

First Optimization

Accuracy Score : 0.9412878787878788

Classification Report

	precision	recall	f1-score	support
Insufficient_Weight	0.97	0.96	0.96	70
Normal_Weight	0.84	0.88	0.86	69
Obesity_Type_I	0.98	0.96	0.97	98
Obesity_Type_II	0.99	1.00	0.99	69
Obesity_Type_III	1.00	1.00	1.00	72
Overweight_Level_I	0.88	0.88	0.88	73
Overweight_Level_II	0.93	0.91	0.92	77
accuracy			0.94	528
macro avg	0.94	0.94	0.94	528
weighted avg	0.94	0.94	0.94	528

Second Optimization

Accuracy Score : 0.9375

Classification Report

	precision	recall	f1-score	support
Insufficient_Weight	0.93	0.97	0.95	70
Normal_Weight	0.86	0.86	0.86	69
Obesity_Type_I	0.98	0.94	0.96	98
Obesity_Type_II	0.97	1.00	0.99	69
Obesity_Type_III	1.00	1.00	1.00	72
Overweight_Level_I	0.90	0.86	0.88	73
Overweight_Level_II	0.91	0.94	0.92	77
accuracy			0.94	528
macro avg	0.94	0.94	0.94	528
weighted avg	0.94	0.94	0.94	528

RESULTS

SUMMARY

Key Components

- **Data Preprocessing:**
  1. Handled missing values.
  2. Converted categorical data into numerical formats for compatibility with machine learning algorithms.

- **Exploratory Data Analysis (EDA):**
  1. Understand the dataset, identify patterns, and uncover relationships between features influencing obesity risk.
  2. Conducted statistical analysis and visualizations using Python libraries like matplotlib and seaborn.

- **Machine Learning:**
  1. Implemented a Random Forest classifier to predict obesity levels.
  2. Achieved 93.56% accuracy, with strong precision (~92%), recall (~91%), and F1-score (~91.5%).
  3. Precision, Recall, and F1-Score: All above 91%, indicating robust and reliable classification.

```
* Accuracy: ~93%
* Precision: ~92%
* Recall: ~91%
* F1-Score: ~91.5
```

CONCLUSION

CONCLUSION

This project demonstrated the effective use of machine learning to predict obesity levels, **achieving a high accuracy of 94.12%** through a Random Forest classifier. The analysis identified weight, physical activity, meal frequency, and dietary habits as the most significant factors influencing obesity risk.

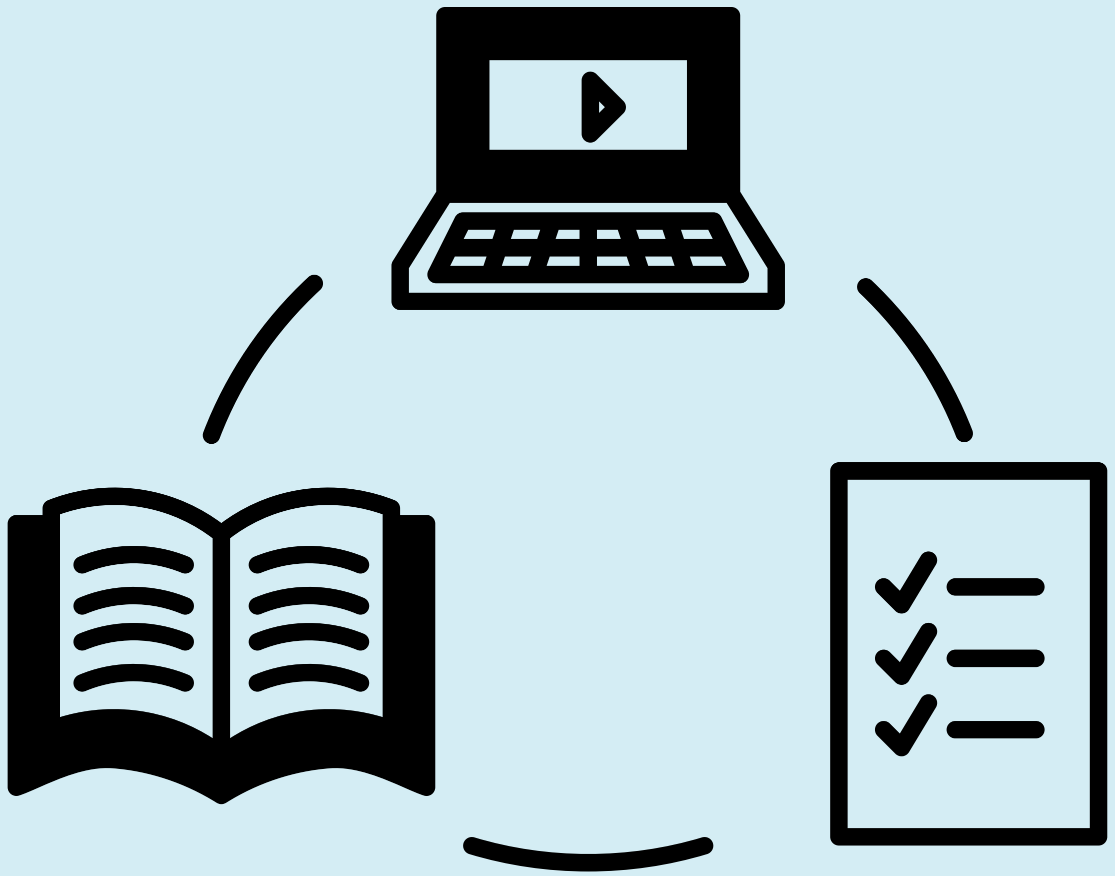
Exploratory Data Analysis revealed actionable insights, such as the importance of increased vegetable intake and regular exercise in reducing obesity risk. Visualizations created with Tableau enhanced understanding and presented key findings in an accessible format.

Overall, this project highlights the critical role of data analysis in addressing health challenges and provides valuable insights for promoting healthier lifestyles.

CONCLUSION

KEY TAKE AWAYS

Random Forest model



- Surprised about the high accuracy of the model.
- Data set could be a little skewed in certain variables like most non-smoking, most public transportation, most with relatives with history of overweight.
- Optimization process is complex.
- Surprised about how certain actions do not affect overweight like counting calories, alcohol consumption.
- Time spent with techonological devices do affect in being overweight!