Obesity Level Classification using Feedforward Neural Networks

STAT 479 Spring 2024

Brian Luu, Cory Suzuki, Meha Patel, Yiming Wang

Overview

- Introduction and Motivation
- About the Data
- Model Building and Tuning
- Results
- Conclusion
- Questions

About Obesity

- Excessive accumulation of body fat
- Obesity estimates in 2022 [1]:
  - 1 in 8 people worldwide live with obesity
  - 43% of adults are overweight, 16% are obese
- Detrimental effects on quality-of-life
  - Physical Health
  - Mental Health
- Goal: Can obesity level(s) be predicted and classified based on known and existing habits and factors?

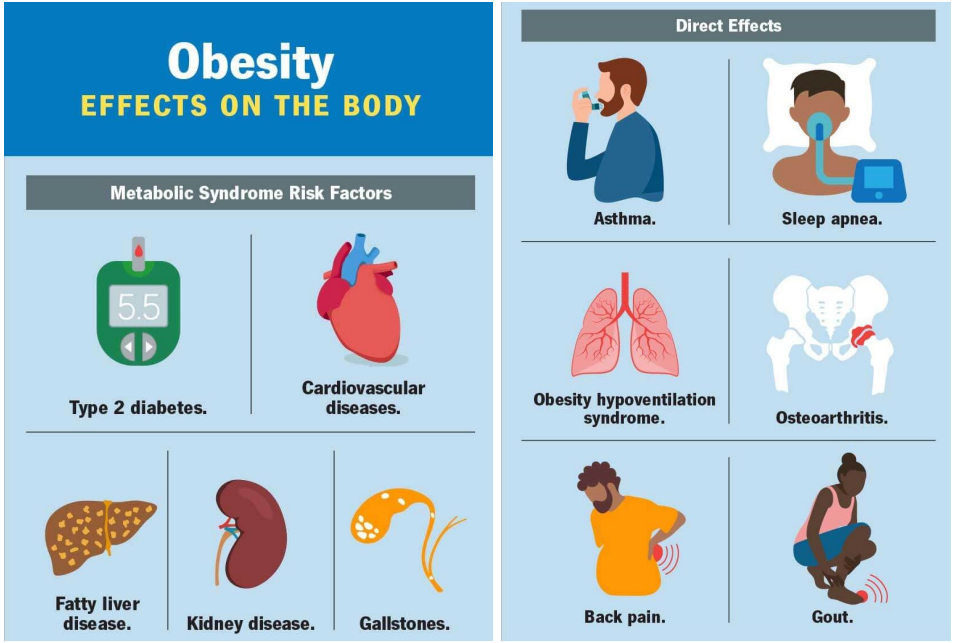


Figure 1. How Obesity may affect the body.
Retrieved from Cleveland Clinic.
https://my.clevelandclinic.org/health/diseases/11209-weight-control-and-obesity

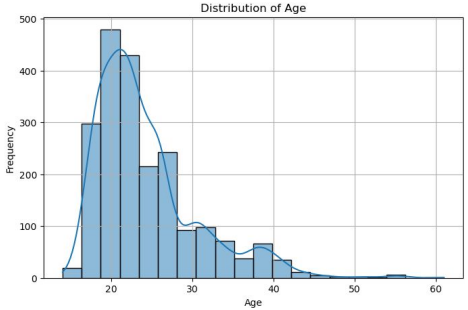
[1] Retrieved from World Health Organization.
https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

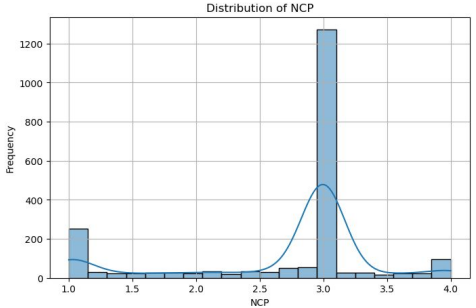
Variables

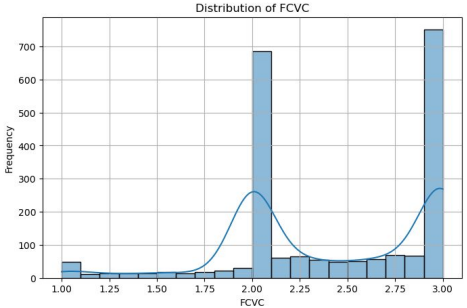
Target: Obesity levels

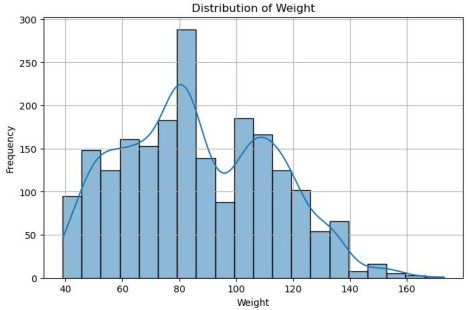
Predictions based on several variables relating to individual characteristics and habits

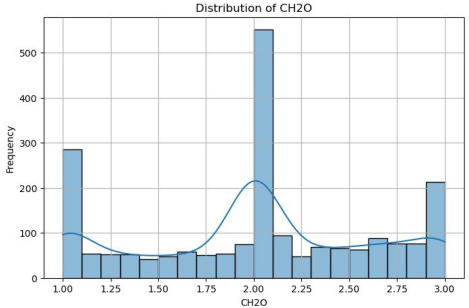
Category	Questions (Code)	Type
Characteristics	• Gender • Age • Height • Weight • Obesity (Family history)	• Categorical • Continuous • Continuous • Continuous • Binary
Dietary Habits	• High caloric foods (FAVC) • Vegetables (FCVC) • Main meals (NCP) • Snacking (CAEC) • Water (CH2O) • Alcohol (CALC)	• Binary • Integer (Cate.) • Continuous • Categorical • Continuous • Categorical
Health Habits	• Smoking (SMOKE) • Physical activity (FAF) • Calorie Monitoring (SCC) • Electronic device usage (TUE) • Transportation usage (MTRANS)	• Binary • Continuous • Binary • Integer (Cate.) • Categorical
Response	Obesity Level	• Insufficient Weight • Normal Weight • Overweight Level I • Overweight Level II • Obesity Type I • Obesity Type II • Obesity Type III

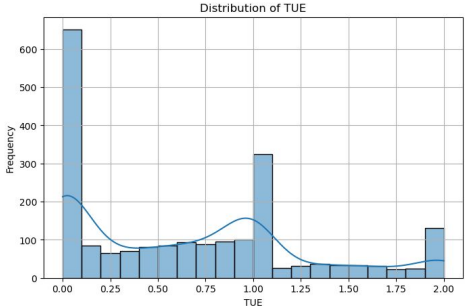


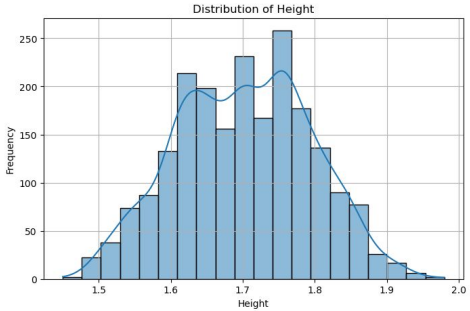


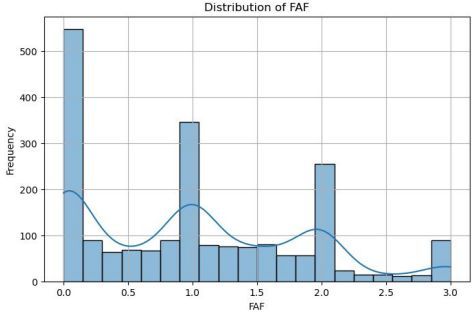




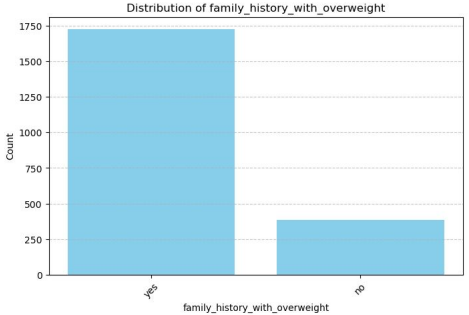


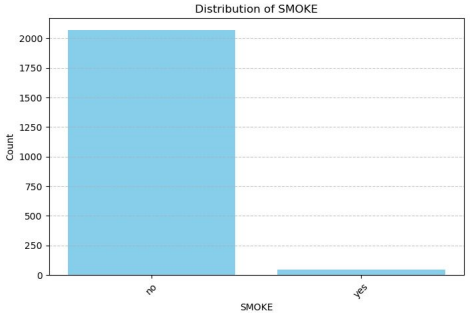


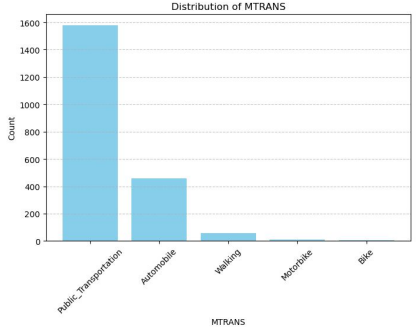


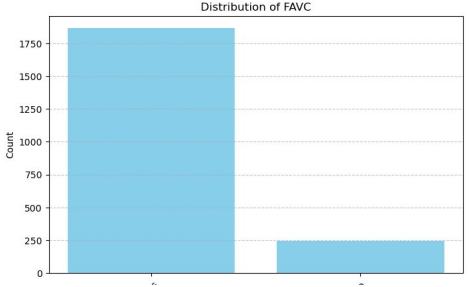


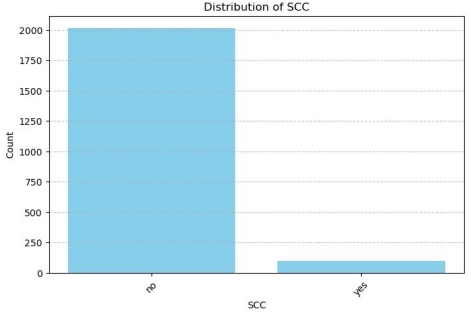
Distribution of Continuous Variables

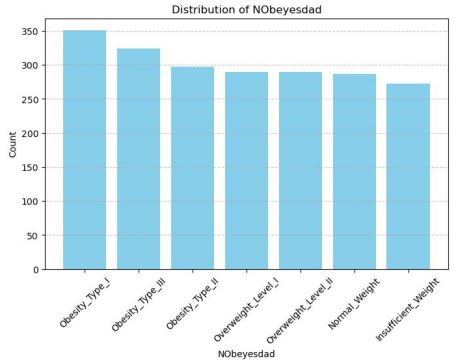


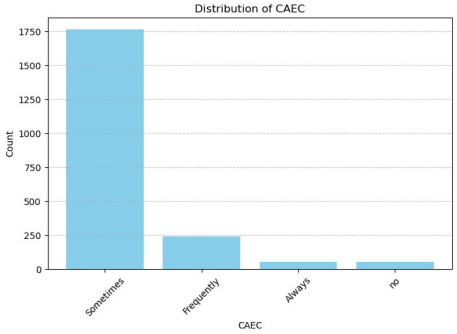


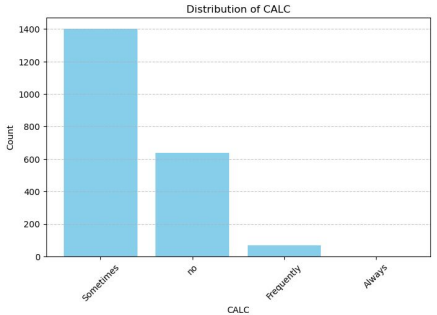












Distribution of Categorical Variables and Target Variable

Pre-processing

Type	Variable	Transformation	Result
Continuous	Age, Height, Weight, NCP, CH2O, FAF	Standardization	$N(\mu = 0, \sigma^2 = 1)$
Integer	FCVC, TUE	One-hot encoding	Prevent ordinality
Categorical	Gender, CAEC, CALC, MTRANS	One-hot encoding	Binary values

After cleaning and standardization, we apply a 60-20-20 split to our data:
Training data: (1266 observations, 31 features)
Validation data: (422 observations, 31 features)
Test data: (423 observations, 31 features)

Model (Prototype)

Input Layers:

Input shape: (**1266** observations, **31** features)

Output Layers:

Output levels: 7

Activation = **“softmax”**

Hidden Layers:

Number of hidden layers: **3**

Number of neurons: **128 -> 64 -> 32**

Activation function of hidden layers: **“selu”**

Kernel_regularizer = **l2(0.001)**

Dropout rate = **0.5**

Model (Prototype) Compile

Optimizer: Nadam

Learning Rate: 0.001

Loss: sparse_categorical_crossentropy

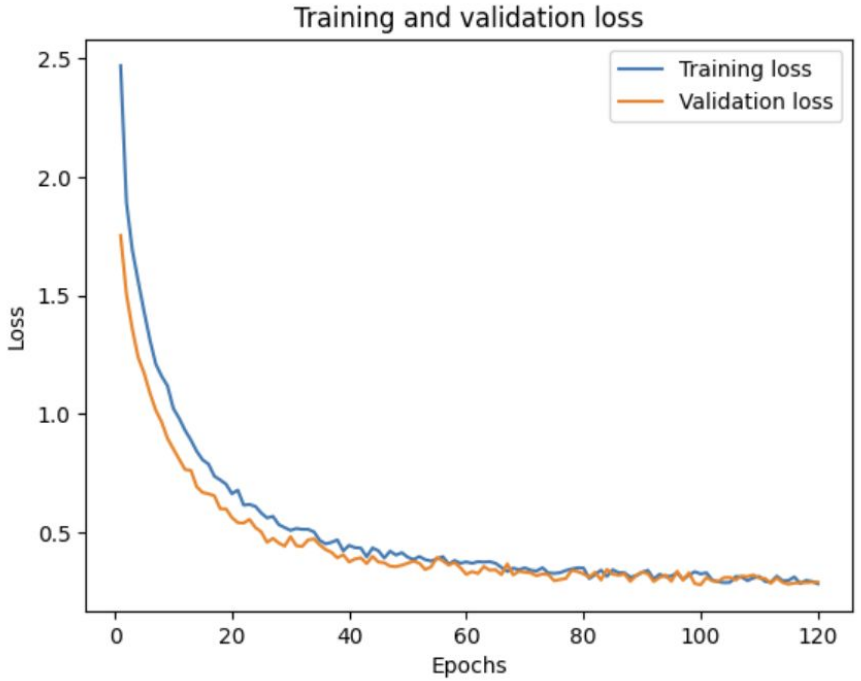
Model Fit

Epochs: 120 & Batch Size: 32

Model (Prototype)Performance

Test loss: 0.2441

Test accuracy: 0.9456



Hyperparameter Tuning

Skeleton Model Structure:

- **Hidden Layers**:
  - Dynamically selects 2 or 3 hidden layers (n_hidden).
  - Layer-specific neuron choices: [128, 64, 32, 16].
- **Learning Rate**: Choices: [0.001, 0.005, 0.01].
- **Regularization**: L2 regularization with choices: [0.001, 0.005, 0.01].
- **Dropout Rate**: [0.3, 0.4, 0.5] for each layer.
- **Optimizer**: Options: Adam, Nadam, Adamax.

Random Search:

- Use kt.RandomSearch for hyperparameter optimization.
- **Objective**: Maximize validation accuracy.
- **Search Space**: Randomly explores combinations of hyperparameters over 25 trials.
- **Training Details**:
  - Epochs: 120
  - Batch Size: 32

Best Model with Highest Validation Accuracy

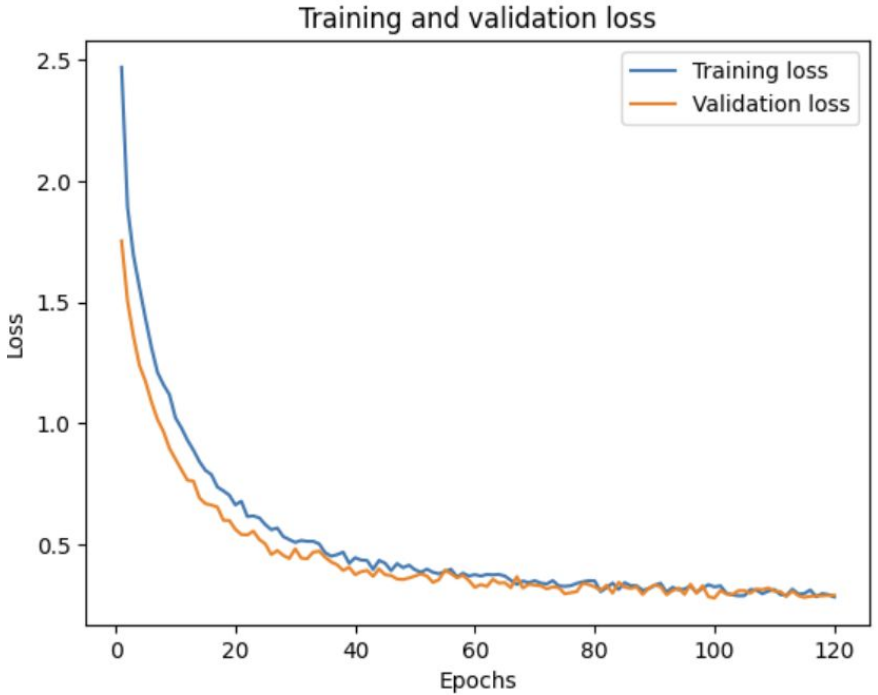
```
Best val_accuracy So Far: 0.9668246507644653
```

```
Hyperparameters:
n_hidden: 3
n_neurons_layer_1: 64
n_neurons_layer_2: 64
n_neurons_layer_3: 64
learning_rate: 0.005
optimizer: Adamax
l2_regularization: 0.005
dropout_rate: 0.3
Score: 0.9668246507644653
```

Construct the BEST Model

Test loss: 0.2532

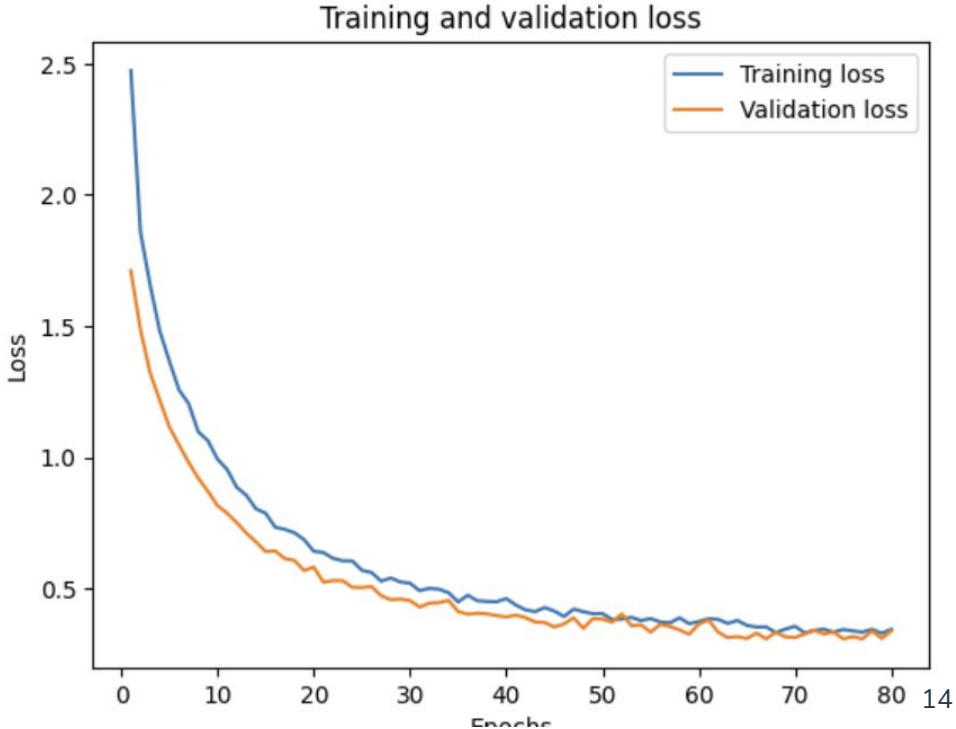
Test accuracy: 0.9669



Reduce Epochs

Test loss 0.2890

Test accuracy 0.9551



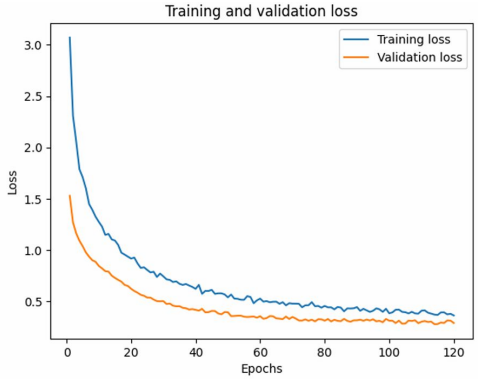
Conclusion/Discussion

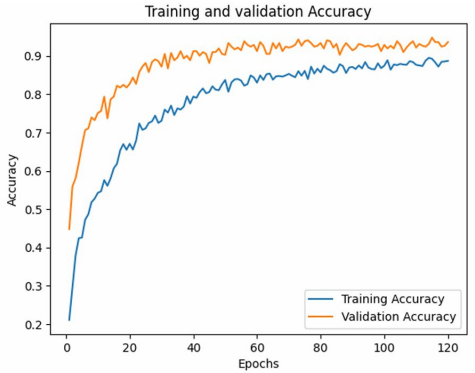
Initial Model (120 epochs):
- Test Loss: 0.2441
- Test Accuracy: 0.9456

Tuned Model
- 120 epochs
  - Test Loss: 0.2532
  - Test Accuracy: 0.9669
- 80 epochs
  - Test Loss: 0.2890
  - Test Accuracy: 0.9551

General Notes:
- Lower epochs -> Prevent overfitting
- Highest accuracy in tuned model with 120 epochs

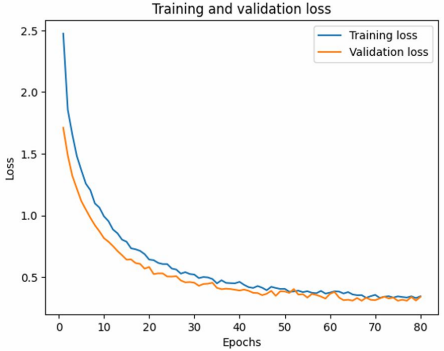
Prototype





Best 120 epochs

Best 80 epochs



Thank you!

Q&A session to follow

Questions & Answers

- Why do we split training, validation, and test datasets before standardizing continuous features and encoding categorical features?
- Is a 60-20-20 split okay?
- Why do we change the feature type from float64 to int for feature FCVC and TUE?
- Why do we standardize continuous features and encode categorical features?
- Why do we use fit_transform for X_train and only transform for X_test and X_val?

Questions & Answers

- Why do we use SELU activation for input and hidden layers?
- What is kernel_regularizer=l2? Why do we need it?
- What is dropout rate? Why do we need it?
- Why do we use softmax for the output?
- What is learning rate? Why do we need it?
- What is sparse_categorical_crossentropy?
- What is batch size? Why do we set it to 32?

Questions & Answers

- What is skeleton_model for hyperparameter tuning?
- What is Keras Tuner for best hyperparameter?
- Why do we use 16, 32, 64, 128 for the number of neurons?
- What is the difference between Adam, Nadam, and Adamax?
- Why is the dropout range 0.3 to 0.5?

Questions & Answers

- Why is the learning rate range between 0.001 to 0.01?
- Why is the L2 regularization range from 0.001 to 0.01?
- Why choose the hyperparameters that provide the best val_accuracy?
- Why is my validation loss curve always below the training loss curve, and why is my validation accuracy curve always above the training accuracy curve?
- Why don't we encode binary features as 0 and 1?