Project Report

Big Data Analytics with Sparklyr

B239464

09-12-2024

Task 1.0: INTRODUCTION TO DATASET

1.1: Context

This dataset include data for the estimation of obesity levels in individuals from the countries of Mexico, Peru and Colombia, based on their eating habits and physical condition. The data contains 17 variables and 2111 observation, the observation are labeled with the class variable NObesity (Obesity categories). 77% of the data was generated synthetically using the Weka tool and the SMOTE filter, 23% of the data was collected directly from users through a web platform.

Download page link: **Obesity DataSet UCI ML**. This page contains features related to dataset.

1.2: Problem

Classification of BMI categories (NObesity) based on eating habits and physical condition using the dataset.

1.3: Loading Required Packages

```r
library(sparklyr) # Big data environment in R
library(dplyr)    # Data wrangling
library(ggplot2)  # Plotting graphs
library(knitr)    # Clean tables
library(broom)    # For tidy data
```

1.4: Create a spark connection

Create a spark connection that treat local machine as a cluster. Load up the downloaded dataset obesity_raw.csv into a Spark dataframe.

```r
sc = spark_connect(master = "local")
obesity_data = spark_read_csv(sc, 'obesity_raw.csv')
```

TASK 2.0: Data Cleaning

2.1: Variable Selection

The obesity_data has 17 variables but this report will be based on just 7 variables including the dependent variable NObeyesdad (BMI categories). The number of observations remains 2111.

This selection was based on domain knowledge and simplicity for health analyst and clinicians. These are 6 potential risk factors for predicting Obesity categories and are chosen based on **Eating habits** and **Physical Condition** as described in the data set. Below are the break down of selected variables.

Eating habits

- Frequent consumption of high caloric food (FAVC)

- Frequency of consumption of vegetables (FCVC)

Physical condition

- Weight
- Height
- Family history with overweight
- Physical activity frequency (FAF)

Finally the **Bmi categories** (NObeyesdad). This is the dependent or response variable.

2.2: Renaming Variables & Summarizing

Here the 7 variables are selected and renamed to lower case for easy manipulation and then a summary statistics is computed

```r
obs <- obesity_data %>%
  select(bmi_cat = NObeyesdad, height = Height, weight = Weight, faf = FAF,
         fcvc = FCVC, favc = FAVC, fho = family_history_with_overweight)

# Quick Summary of selected Variables.
sdf_describe(obs) %>%
  mutate(height = round(height, 2),
         weight = round(weight, 2),
         faf = round(faf, 2),
         fcvc = round(fcvc, 2)) %>%
  kable(format = "pipe", caption = "Summary Statistics") # from knitr package
```

Table 1: Summary Statistics

summary	bmi_cat	height	weight	faf	fcvc	favc	fho
count	2111	2111.00	2111.00	2111.00	2111.00	2111	2111
mean	NA	1.70	86.59	1.01	2.42	NA	NA
stddev	NA	0.09	26.19	0.85	0.53	NA	NA
min	Insufficient_Weight	1.45	39.00	0.00	1.00	no	no
max	Overweight_Level_II	1.98	173.00	3.00	3.00	yes	yes

Categorical Variables: `bmi_cat`, `favc`, `fho`

Numeric Variables: `Height`, `Weight`, `fcvc`, `faf` are all continuous Variables

A count of 2111 observation across each all variables. A higher standard deviation seen in `weight` & `faf`.

2.3: Check Missing Values

This is to check for any missing values in the current dataset.

```
obs %>%
  summarise_all(~sum(as.integer(is.na(.)))) %>%
  kable()
```

```
## Warning: Missing values are always removed in SQL aggregation functions.
## Use ‘na.rm = TRUE‘ to silence this warning
## This warning is displayed once every 8 hours.
```

bmi_cat	height	weight	faf	fcvc	favc	fho
0	0	0	0	0	0	0

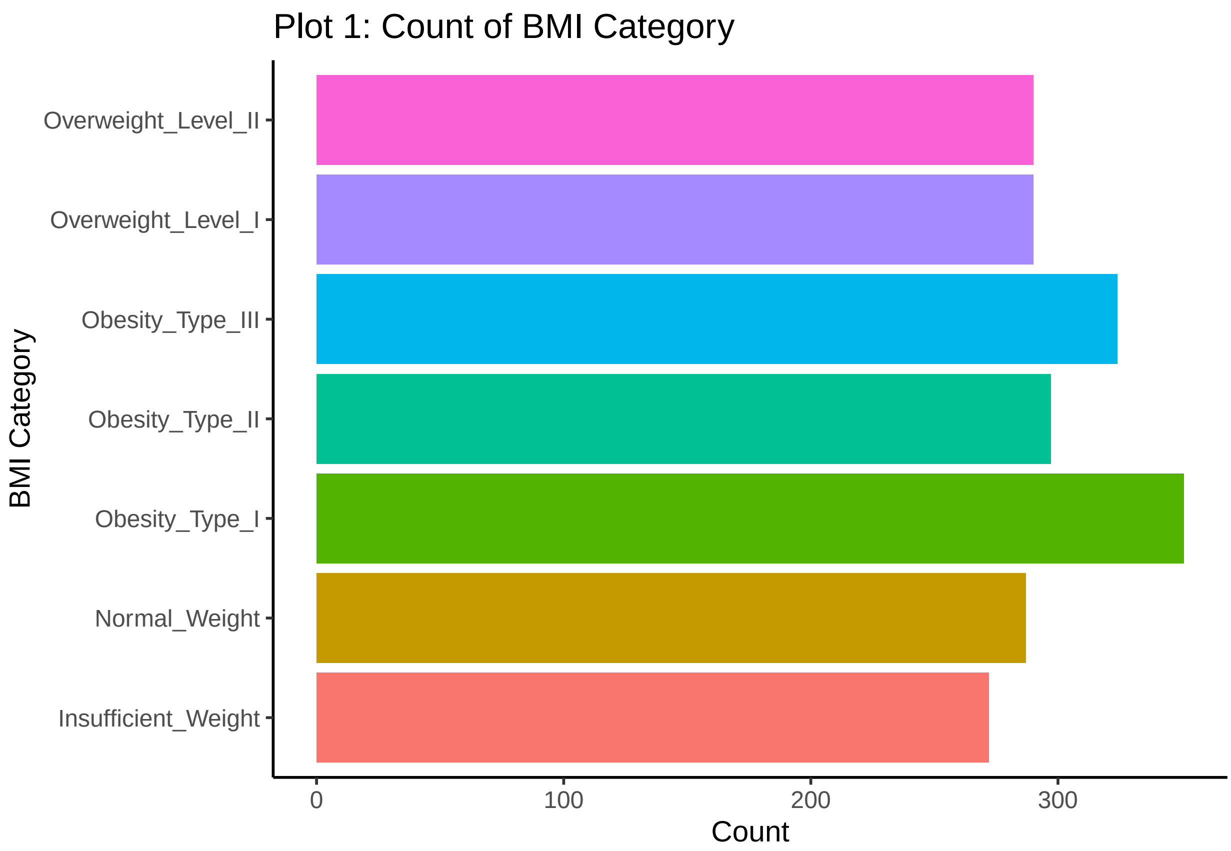
No NA values seen in all selected variables.

Task 3.0: Exploratoy Data Analysis

3.1: BMI category Count

```
obs_plot1 <- obs %>%
  group_by(bmi_cat) %>%
  summarise(n = count()) %>%
  collect()

ggplot(obs_plot1, aes(x = bmi_cat, y = n, fill = bmi_cat)) +
geom_col() + coord_flip() + theme_classic() +
theme(legend.position = 'none') +
labs(title = 'Plot 1: Count of BMI Category',
     x = 'BMI Category', y = 'Count')
```

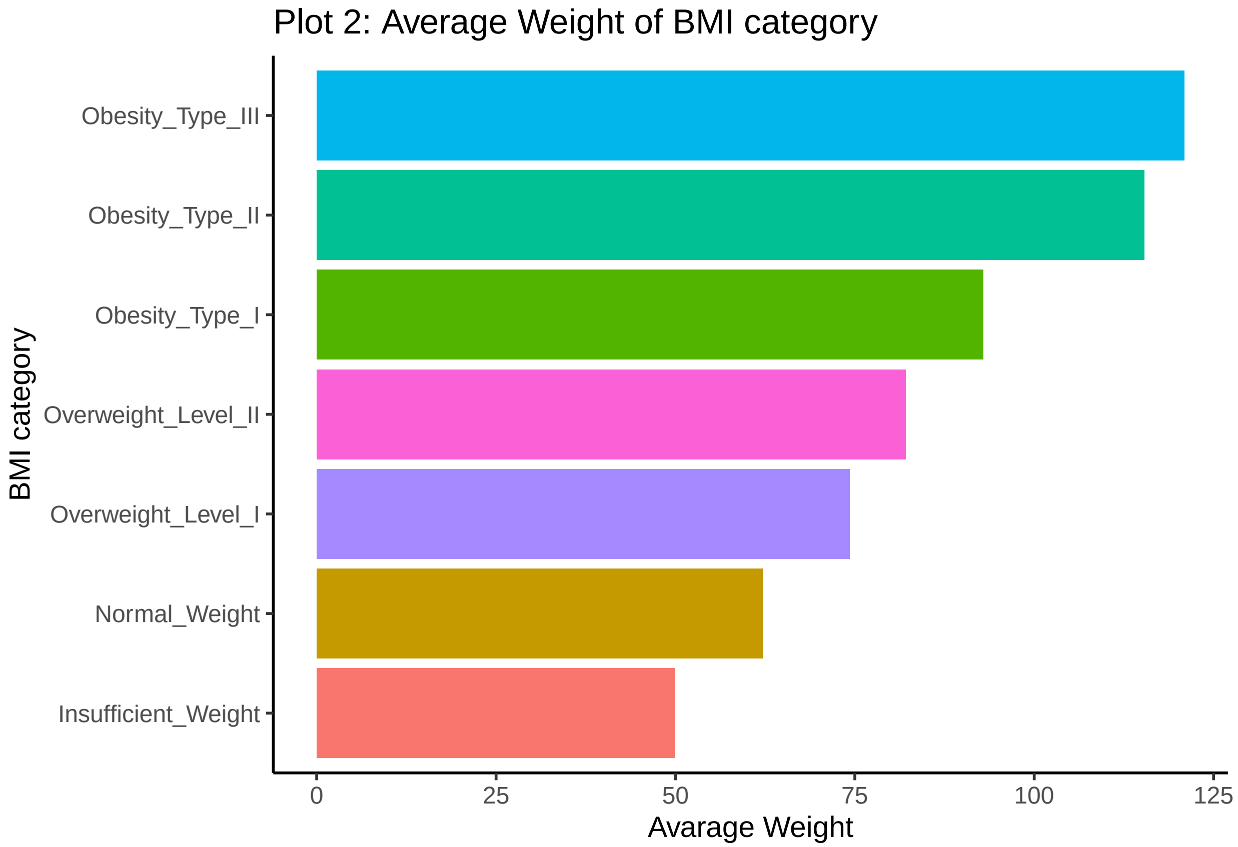


The plot shows that all levels are relatively equally distribution, with slightly higher counts for obesity type 1 and type 3.

3.2: BMI Category by Avarage Weight

```
obs_plot2 <- obs %>%
  group_by(bmi_cat) %>%
  summarise(avg_weight = mean(weight)) %>%
  collect()

ggplot(obs_plot2, aes(x = reorder(bmi_cat, avg_weight),
                      y = avg_weight, fill = bmi_cat)) +
geom_col() + coord_flip() + theme_classic() +
theme(legend.position = 'none') +
labs(title = 'Plot 2: Average Weight of BMI category',
     x = 'BMI category', y = 'Avarage Weight')
```



The bar plot shows the distribution of BMI categories by average weight. As seen above, the BMI category increases as the average weight increase. Weight could potentially be a major predictor to model BMI Category.

3.3: Pairwise correlation

The table computes a correlation matrix across all continuous variables in the selected dataset. It shows how strongly each pair of variables is related to each other.

```
obs %>%
  select(weight, height, faf, fcvc) %>%
  ml_corr() %>% kable(format = "pipe",
                      caption = "Pairwise Correlation")
```

Table 3: Pairwise Correlation

weight	height	faf	fcvc
1.0000000	0.4631361	-0.0514363	0.2161247
0.4631361	1.0000000	0.2947090	-0.0381211
-0.0514363	0.2947090	1.0000000	0.0199394
0.2161247	-0.0381211	0.0199394	1.0000000

A moderate positive correlation is seen between `weight` and `height` (0.46). There is also a positive relationship between `height` and physical activity `faf` (0.29), followed by `weight` and frequency of vegetable consumption `fcvc` (0.21).

No multicollinearity observed as all correlation values are less than 0.8 or greater than -0.8, which are far from absolute 1 or -1 correlation.

TASK 4.0: Model Justification

From the selected variables, the aim is to classify BMI categories based on selected eating habits and physical condition in the created *obs* data object. The model features:

Independent/Predictor Variables

- Frequent consumption of high caloric food (favc)
- Frequency of consumption of vegetables (fcvc)
- Family History with Overweight (fho)
- Physical activity frequency (faf)
- Weight
- Height

Dependent/Response Variables

- BMI category (bmi_cat)

However, a good fit to model such several levels of `bmi_cat` variable would be a **Random Forest Tree Model**. A better choice would be the gradient boosted trees model but sparklyr only agrees for binary classification using gradient boosted trees for now.

Why choose Random Forest Tree?

The principle of a Random Forest tree model (RFT) is based on an ensemble learning method that builds multiple decision trees and combines their outputs to improve predictive performance and reduce overfitting. Here's a breakdown of the key concepts:

- The final prediction is based on majority voting among the individual trees. The class with the most votes is selected as the predicted class.
- It is suitable for high levels categorical dependent variable modelling and handling both categorical and numerical predictors efficiently without needing to one hot encode.

TASK 5.0: Modelling Implementation

Every steps taken here is done as it would done in a big data settings.

5.1: Model Training

- Splitting the **obs** data in train and test data. 70% for training and the remaining 30% for testing data (like a unseen/new dataset).
- Setting a seed help ensure consistency so that every time the code is run, the same result is achieved (Reproducability).
- Tuning the number of trees in the model, helps reduce variance among trees and improves stability, which by default is 20 trees. The dataset is a little complex, so a good start of 50 trees with computational demand in check.
- Setting the max depth of the tree, controls complexity of the trees and prevents over fitting. It is unlimited but a good start for this dataset would be between 10 - 30.

```r
set.seed(369) # for reproducibility
train_data <- obs %>% sample_frac(.70) # 70% to train
test_data <- obs %>% anti_join(train_data) # remaining 30% to test

# Create a Random Forest model
rf_model <- ml_random_forest(train_data,
                             bmi_cat ~ height + weight +
                             faf + fcvc + favc + fho,
                             type = "classification",
                             num_trees = 50, # mitigates variance
                             max_depth = 10) # controls tree complexity

# Evalute model matrix on the training set: Accuracy
ml_evaluate(rf_model, train_data) %>% kable()
```

Accuracy
0.9952619

The model achieved a high accuracy of 99.5% on the training set, suggesting it has learned the patterns in the training data well. We can not fully conclude on this accuracy until it is used on a new dataset or the test data. All this is done to make sure there is no over fitting, which is when the training set is performing far better than the test set or unseen data.

5.2: Feature Analysis

This is done to checks the level of importance for each predictor variable (feature) on the model. That is the contribution rate of each of them to the model.

```r
# Feature Analysis
tidy(rf_model) %>% # from broom package
  mutate(importance = round(importance, 3)) %>% # round to 3 decimal places
  kable(format = "pipe", caption = "Feature Importance of RFT Model")
```

Table 5: Feature Importance of RFT Model

feature	importance
weight	0.517
height	0.223
fcvc	0.139
faf	0.064
fho_yes	0.033
favc_yes	0.024

`weight` has the largest contribution with approximately 52% on predicting BMI Category followed by `height` with 22%. The least important variable is frequent consumption of high caloric food `(favc)` with just 2%. The sum of the importance column makes up 1 or 100% of the contribution power in modelling Bmi category.

We can also see that Random forest handled the independent categorical variables efficiently and avoided the dummy trap.

5.3: Model Prediction

Here predictions are made on test set, making it look like a new set of data which the model has not seen or learnt before. Lets view the top rows of predictions

```
# Make predictions on the testing set
ml_predictions <- ml_predict(rf_model, test_data)

ml_predictions %>%
  select(bmi_cat, label, predicted_label, prediction) %>%
  head() %>% kable(format = "pipe",
                   caption = "Top Row Predictions & lables on Test Data")
```

Table 6: Top Row Predictions & lables on Test Data

bmi_cat	label	predicted_label	prediction
Insufficient_Weight	6	Insufficient_Weight	6
Insufficient_Weight	6	Insufficient_Weight	6
Insufficient_Weight	6	Insufficient_Weight	6
Insufficient_Weight	6	Insufficient_Weight	6
Insufficient_Weight	6	Insufficient_Weight	6
Insufficient_Weight	6	Insufficient_Weight	6

Wow, from the first few rows, the model seems to have perfectly classified the `bmi_cat` with is respective labels. A very few more rows may have been misclassified with deeper observation.

5.4: Model Evaluation

Evaluating the predictive power of the test set. To know how efficient it is at classifying BMI categories based on the predictor variables.

5.4.1: Model Accuracy

```
# Evaluate model accuracy on the predicted model from the test dataset
ml_multiclass_classification_evaluator(ml_predictions,
                                       metric_name = "accuracy")
```

```
## [1] 0.9564516
```

The final model matrix on the test set produces an accuracy of 96%.

5.4.2 Model F1 Score

```
# Evaluate model f1 score on the predicted model from the test dataset
ml_multiclass_classification_evaluator(ml_predictions,
                                       metric_name = "f1")
```

```
## [1] 0.9568114
```

The final model matrix on the test set produces a f1 score also indicating a 96% score.

Task 6.0: Model Interpretation

- *Accuracy*: 0.9564516

This indicates that the model correctly classified approximately *96%* of obesity levels instances in the test dataset. High accuracy suggests the model is performing well in general.

- *F1 Score*: 0.9568114

The F1 score is a measure of the model's accuracy that considers both precision and recall. A score of approximately *0.96* or *96%* indicates a good balance between precision and recall (sensitivity), which means the model is both reliable in its predictions and correctly identifies true positives while minimizing false positives instances for respective BMI category.

To reiterate, the result shows that the predictors are well suitable for considering as risk factor when trying to classify the BMI categories.

Task 7.0: Conclusion

7.1: Strengths of the Report Approach

1. *Clear Variable Selection:* The selection of variables is well-justified, focusing on key factors related to eating habits and physical conditions. This targeted approach simplifies the analysis and makes it more relevant for health analysts and clinicians.
2. *Robust Modelling Technique:* The choice of a Random Forest Model is appropriate given its ability to handle both categorical and numerical predictors efficiently. The model's high Accuracy and F1 score indicate strong predictive performance.
3. *Detailed Model Evaluation:* The report provides a thorough evaluation of the model's performance, including accuracy and F1 score, which helps in assessing the model's reliability and effectiveness.

7.2: Weaknesses of the Report Approach

1. *Synthetic Data Proportion:* The dataset includes 77% synthetically generated data, which might not fully capture the complexities and variations of real-world data. This could affect the generalisation of the model's predictions in real life settings.

2. *Model Complexity:* Random forest is like a complex technique where most of what happens could not be explained by basic statistical analogy e.g the standard deviation and confidence intervals. The model only evaluates Accuracy and F1 scores.

3. *Computational Demand:* Using such Machine learning techniques may require higher demand of computational resource especially when it comes to tuning features like number of trees, max depth etc. Running validation processes like cross validations could take more computational resources. However, the dataset was splitted into two, and low to moderated parameter tuning has been set in respect to the dataset used.

7.3: Broad Implications of the Analysis

1. *Public Health Insights:* The analysis provides valuable insights into the factors contributing to obesity, which can inform public health strategies and interventions aimed at reducing obesity rates in the studied regions.

2. *Policy Development:* Policymakers can use the findings to develop targeted policies that address specific risk factors, such as reducing weight, promoting healthier eating habits and increasing physical activity among populations.

3. *Clinical Applications:* Health professionals can leverage the model to identify individuals at risk of obesity and implement preventive measures or personalized treatment plans based on the identified risk factors as seen in the exploratory analysis and Feature Importance.

In summary, the report demonstrates a solid approach to classifying BMI categories using RFT, with clear strengths in variable selection, feature importance and model evaluation. Addressing the identified weaknesses and considering the broader implications can enhance the impact and applicability of the findings.