Annie Jiang
CIS 3120
May 22, 2024

Obesity Classification Report

Obesity is a significant health issue affecting individuals worldwide, leading to various health complications such as diabetes, heart disease, and hypertension. Effective classification of obesity levels can aid in early diagnosis and targeted interventions.

In this project, I built a classification system to predict obesity levels based on various demographic, lifestyle, and physiological factors. To do so, I created and evaluated several machine learning models in Python including k-nearest neighbors, naive Bayes, support vector machines with different kernels, decision trees, random forests, and logistic regression. After determining the model with the best performance, I then deployed it to an online cloud provider via a web interface. In the following paragraphs, I will go into more detail about the methodologies used.

Firstly, I preprocessed the data. For context, the dataset used for this project was sourced from a publicly available repository, Kaggle, and consists of 2111 rows and 17 columns. The dataset includes various features such as demographic details (age, gender), lifestyle habits (smoking, alcohol consumption), and physiological measures (height, weight). After looking at how the data was formatted, I checked for missing values, and fortunately there were none. I also looked at the descriptive statistics and data types for each column, which helped me gain a better understanding of the variables present in the dataset.

Next, I created several data visualizations to explore relationships between different variables and identify any potential patterns or trends. I made a pairplot, shown in the appendix as Plot 1, which displays the relationships between the features in the dataset with each point colored by the obesity classification. We can extract several insights from this plot alone. For example, looking at the height vs. weight plot, there is clearly a positive linear relationship between height and weight. The obesity levels in the plot also form distinct clusters, with higher obesity levels corresponding to higher weights for similar heights. In addition, I created a correlation matrix using a subset of the numerical columns, as shown in Plot 2. Again, we can see that height and weight have a strong correlation. We can also see that the other features show weaker correlations with each other. This suggests that these features may provide non-redundant information when doing classification. Furthermore, when I created Plot 3, a bar plot showing gender distribution by obesity level, I noticed something interesting. It came to my attention that almost all of those who classified as Obesity Type III were female, and almost all of those who classified as Obesity Type II were male. This observation prompted a closer inspection of the data, and upon further analysis, only 1 observation of a male classified as Obesity Type III and 2 observations of females classified as Obesity Type II. This will have to be something that will be taken into consideration for future analysis.

After visualizing the data, I began to prepare it for classification. First, I encoded all the categorical columns so that the machine learning algorithms will know how to recognize them. I separated the 8 categorical features and encoded them using OneHotEncoder from the scikit-learn library. Next, I standardized the numerical features, which I have 8 of, using StandardScaler. After encoding and standardizing, I separated the label, NObeyesdad (obesity level), from the features to then split the data into training and test sets. I divided the data in a 70:30 ratio, so that 70% of the data would be used for training and the other 30% would be used to test model performance.

Now that the data was all ready, I fit the classification models to the training dataset and evaluated their performance using 5-fold cross validation. I used k-nearest neighbors with k = 5; Gaussian naive Bayes; linear, RBF, polynomial, and sigmoid SVMs with C = 1 and gamma = 1; decision tree; random forest; and logistic regression. Using 5-fold cross validation, I found that the model with the highest accuracy was linear SVM, with 93.97%. This may be because there was a clear separation of classes, similar to when we looked at the pair plot and saw evident clusters and relationships between features like height and weight. Next were decision tree and polynomial SVM, with accuracies of 91.13% and 90.25%, respectively. Decision trees are good at capturing non-linear relationships and interactions between features, which likely contributed to their strong performance. The polynomial SVM could also capture more complex relationships than the linear SVM, but there is the risk of it overfitting. The model with the worst accuracy was naive Bayes at 47.93%. This makes sense, since naive Bayes assumes that there is no correlation between the features, which is hardly the case in real life. Naive Bayes should mainly be used when computing speed is prioritized over accuracy.

Afterwards, I created classification reports for each model based on their predictions on the test data. While I used 5-fold cross validation to evaluate the models' performance, this evaluation was performed solely on the training data that I used to fit the models. Now, it was time to have the models make predictions and compare them to the actual results in the unseen test data. The final performance of the models would be judged based on metrics such as accuracy, precision, recall, and f1-score, which are derived from the models' confusion matrices. After fitting the models to the training data, I used the test data to generate predictions and created classification reports for each model. These reports provided a detailed breakdown of each model's performance, including the aforementioned metrics. Furthermore, I put together the weighted averages of each model from the classification reports and plotted them all on a single bar plot, shown in Plot 5. This way, we have a visualization for the comparison of performances across models. Looking at the plot, it is clear that the model with the highest weighted averages was the linear SVM. This model not only achieved high accuracy but also maintained high precision, recall, and F1-score, with all of them at 96%. This shows the model's robustness and reliability in classifying obesity levels.

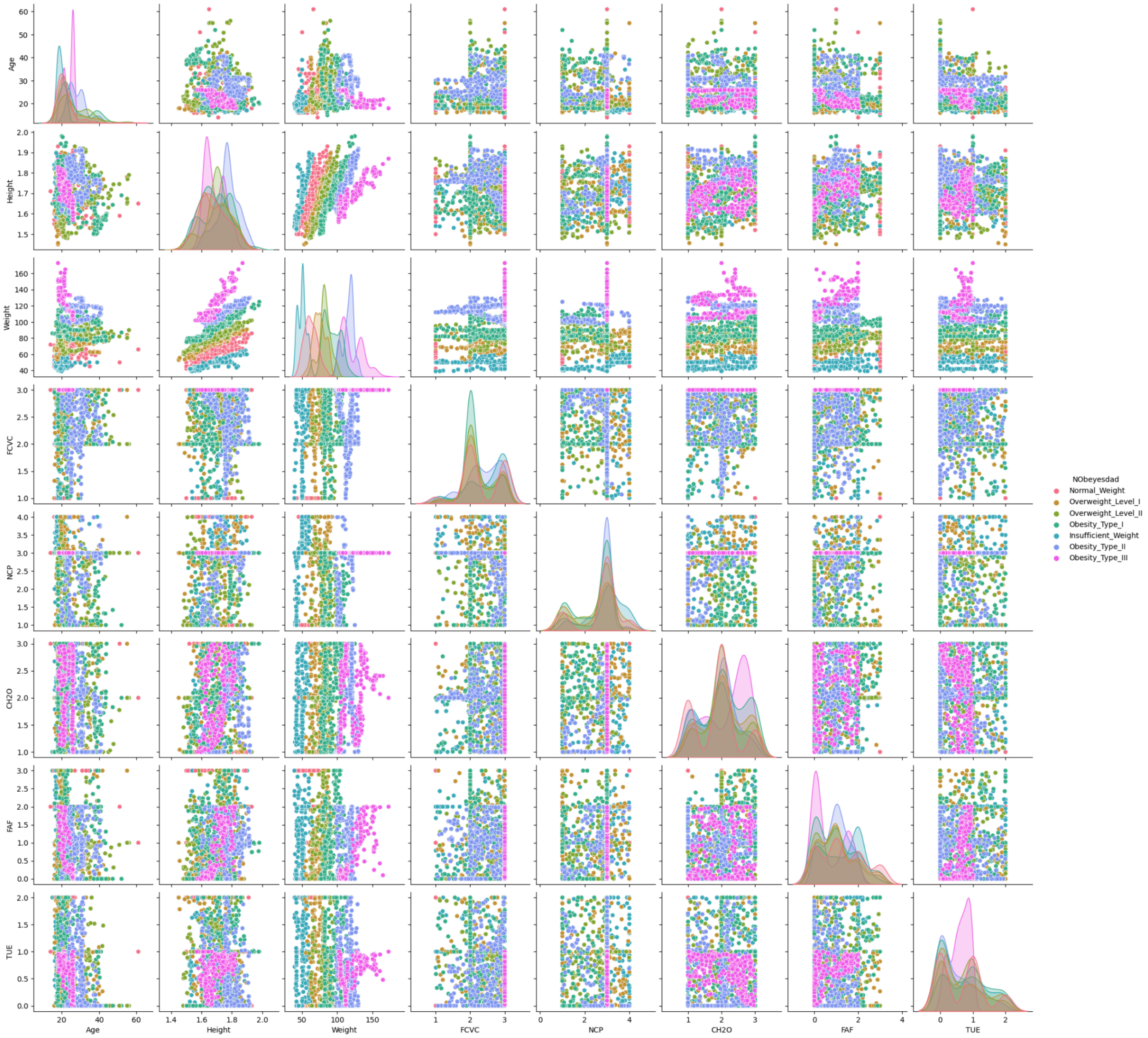
Having determined the optimal model, I was ready to deploy it to a web interface so that non-technical users would be able to make predictions on obesity levels with it. I used PythonAnywhere to build and host a website where I could deploy the model. PythonAnywhere provides an accessible platform for hosting Python applications, which makes it ideal for deploying machine learning models. In preparing the model for deployment, I saved the trained linear SVM model using Python's pickle library. After that, I developed a simple web application using Flask, a web framework for Python. I built a simple form where users can input information for the features used in the model, such as their age, gender, dietary habits, physical activity, etc. Upon submission, the model makes a prediction and displays the result along with the confidence.

This program can be helpful in several ways. First of all, it allows users to easily input relevant health and lifestyle information and receive immediate feedback on their obesity levels. This model can help with early detection of obesity, which lets users know if they need to make any lifestyle or dietary changes. Secondly, this program can be used as an educational tool. Users can play around with the model and learn more about the factors contributing to obesity. It can motivate individuals to adopt healthier behaviors by providing feedback on their health status.

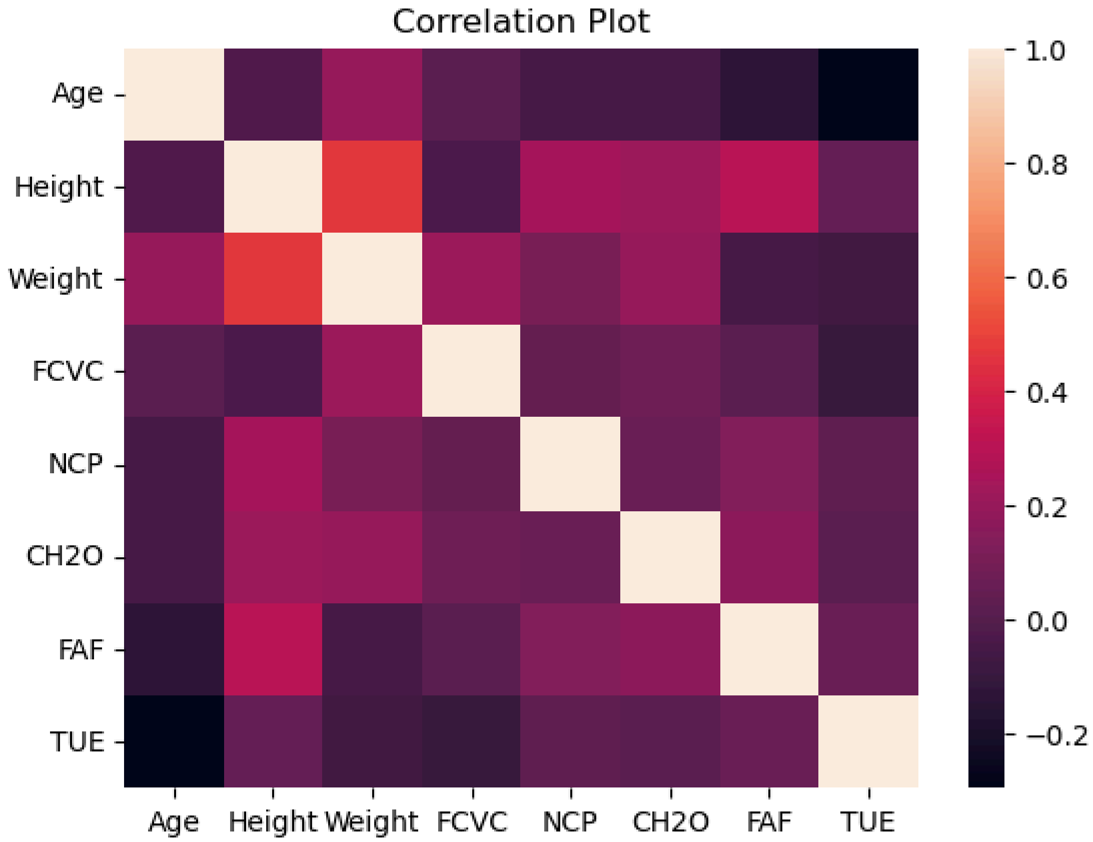
Overall, in this project, I determined the best machine learning model to classify obesity level based on various features, which is linear SVM. I deployed the model on a website, where users can use the model to make predictions based on their inputted data. This program has a lot of potential, since it allows users to receive immediate feedback on their obesity levels, and it can be used as an educational tool to encourage individuals to adopt healthier habits.

Appendix

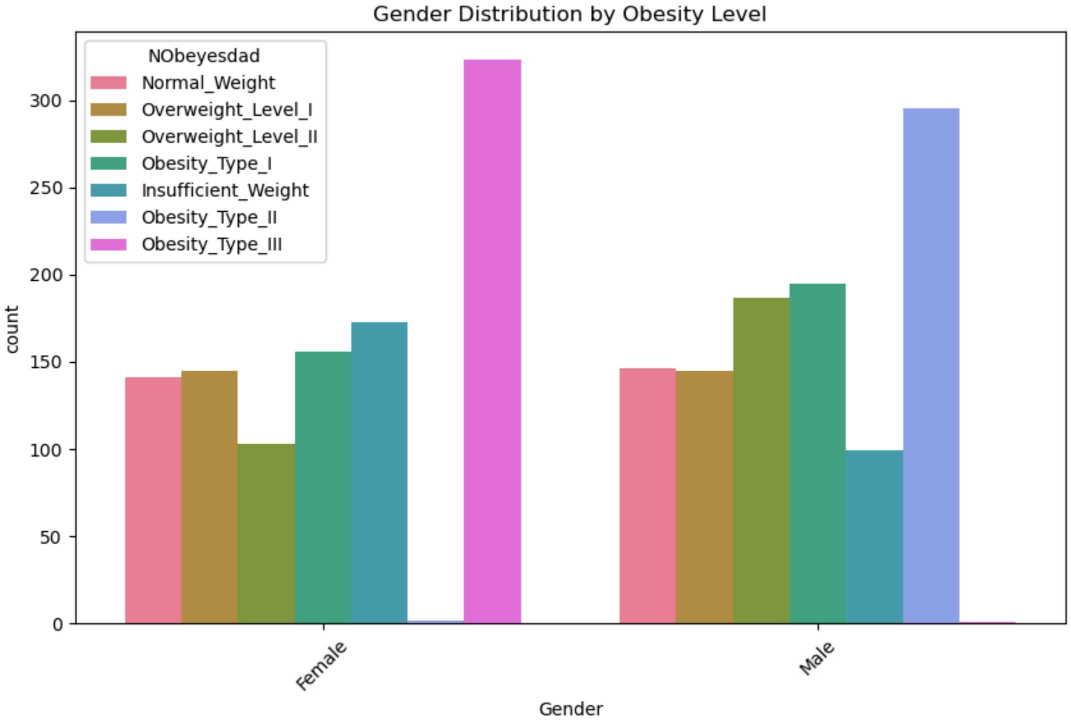
Plot 1 - Pairplot



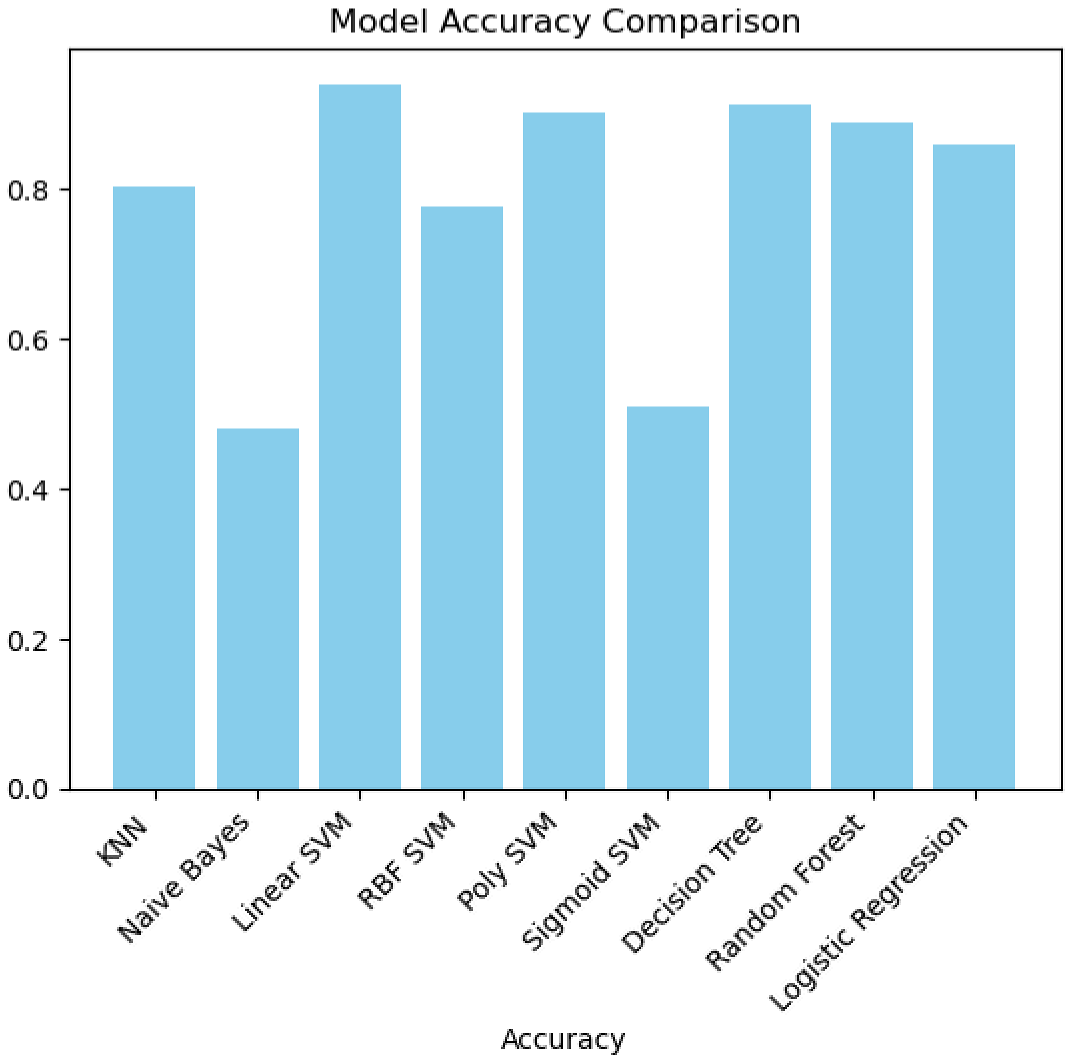
Plot 2 - Correlation Matrix



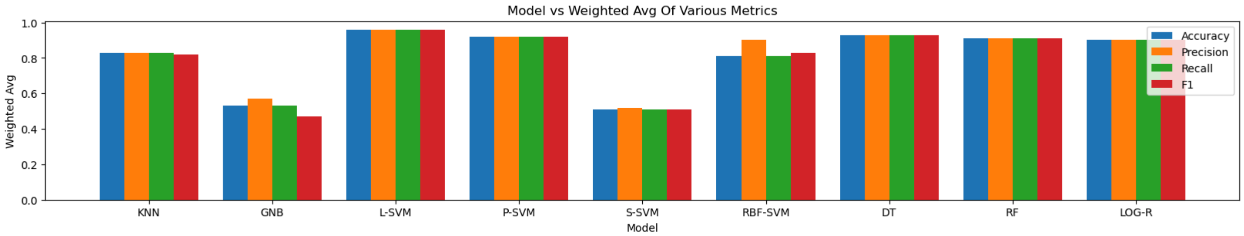
Plot 3 - Gender distribution by level



Plot 4 - Model Accuracy Comparison



Plot 5 - Classification Metrics Graph


Model vs Weighted Avg Of Various Metrics
Weighted Avg
KNN
GNB
L-SVM
P-SVM
S-SVM
RBF-SVM
DT
RF
LOG-R
Model
Accuracy
Precision
Recall
F1