Estimation of Obesity levels based on eating habits and physical conditions of people

Manash Bhele
Softwarica College
MSc Data Science and Computational Intelligence
Kathmandu, Nepal

***Abstract*—This paper looks at how eating habits and physical conditions of people from Mexico, Peru and Colombia with ages between 14 and 61 can be used to estimate obesity levels.The project uses data from UCI ML Repository. Various classification algorithms were implemented using Python Programming language and its packages and many performance metrics were applied to visualize and compare their performances.**

***Index Terms*—Obesity, Eating habits, Physical conditions, Estimation, Classification, Random Forest Classifier, Light GBM, KNeighbors, Accuracy, ROC, AUC, F1 score**

I. Introduction

The WHO defines obesity and overweight as abnormal or excessive fat accumulation that may impair health. According to the data from WHO 39% of adults aged 18 years and older were overweight and 13% were obese in 2016. [1] Obesity is considered to be a concerning issue and has also been listed as one of the top 5 reasons for global death. Many studies have exemplified that obesity is a complex health issue which arises from many factors including genetics as well as unhealthy eating habits. [2] A meta-analysis performed to investigate whether a measure like childhood obesity using BMI could predict future obesity in adolescence and adulthood, brought to light the fact that, obese children and adolescents were 5 times more likely to be obese in adulthood. [3] The physical appearance and health conditions of a person depends on many factors. Foremost, factors like their daily eating habits have been found to determine their physical and mental well being and quality of life [4].

Many researchers also agree upon the fact that obesity is a disease that individuals become afflicted with, due to their lifestyle choices such as low rates of physical activeness and binge-eating. Physical activity frequency has also been linked to whether a person will be obese or not. Studies have found that with age, physical activity decreases and preventing such decline in activity might help in reducing obesity. [5]

Various studies have also shown that Machine Learning Techniques can be used to predict signs of early childhood obesity. [6] With the increase in the usage of technology in everyday life and advancements in various sectors of health sciences with the use of technology, being able to predict an individual's future health statistics is not a no-go.

II. Data set Description

Physical activities frequency and eating habits have been found to be highly related to the BMI of individuals. [7] This project focuses on whether the physical conditions and eating habits of a person can be used to estimate if they are obese or not.

The data set used in this project comes from an online survey carried out by Fabio Mendoza Palechor and Alexis de la Hoz Manotas from Universidad de la Costa, CUC, Colombia. The data set has been made accessible through an open access article. [8]

The process of data collection was carried out making using of a web platform with a survey where anonymous users answered the questions. The initial dataset had 485 records with use of which, the data was labeled into various obesity categories by calculating the mass body index using the equation:

$$\text{Mass Body Index} = \frac{\text{Weight}}{\text{Height * Height}}$$

The values of BMI were then categorized as:

- Underweight Less than 18.5
- Normal 18.5 to 24.9
- Overweight 25.0 to 29.9
- Obesity I 30.0 to 34.9
- Obesity II 35.0 to 39.9
- Obesity III Higher than 40

Due to an unbalance in the categories, SMOTE was used to introduce up to 77 percent of synthetic data which in turn balanced the dataset. The final balanced dataset used here has 2111 records and 17 attributes. The attributes have been generally divided into two categories, namely 'Attributes related with Eating Habits', 'Attributes related with Physical Conditions' and other remaining. The attributes related with eating habits are:

1) Frequent consumption of high caloric food (FAVC)
2) Frequency of consumption of vegetables (FCVC)
3) Number of main meals (NCP)
4) Consumption of food between meals (CAEC)
5) Consumption of water daily (CH20)
6) Consumption of alcohol (CALC).

The attributes related with the physical condition are:

1) Calories consumption monitoring (SCC)
2) Physical activity frequency (FAF)
3) Time using technology devices (TUE)
4) Transportation used (MTRANS)

other remaining variables are:

1) Gender
2) Age
3) Height
4) Weight.
5) Smoke (SMOKE)
6) Family History with Overweight (family_history_with_overweight)

Table I displays the overview of the data set outlining the features and its types.

No.	Feature	Feature Type
1	Gender	Categorical
2	Age	Numerical
3	Height	Numerical
4	Weight	Numerical
5	family_history_with_overweight	Categorical
6	FAVC	Categorical
7	FCVC	Numerical
8	NCP	Numerical
9	CAEC	Categorical
10	SMOKE	Categorical
11	CH2O	Numerical
12	SCC	Categorical
13	FAF	Numerical
14	TUE	Numerical
15	CALC	Categorical
16	MTRANS	Categorical
17	NObeyesdad	Categorical

TABLE I
DATA SET FEATURES

III. Methodology

A. KNeighbors Classifier

The KNeighbors classifier comes from the kNN algorithm. In this algorithm the class label of a data point is determined by the majority class label of its k nearest neighbors. [9] The KNeighbors classifier is one of the most commonly used classification techniques where the value of k is an integer which is specified by the user and is highly dependent on the nature of data. [10] Various distance metrics are used in order to measure the distance between the query point and the other k data points, with Euclidean Distance being the most used metric. [11]

B. Random Forest Classifier

Random Forest is a supervised machine learning algorithm which can be used in regression problems as well as problems related to classification of categorical target variables. It is a type of algorithm, generally termed as an 'Ensemble' method. Ensemble methods combines multiple models instead of using a single model which in-turn helps in improving the accuracy of the model. Random Forest combines prediction from smaller models where each of the smaller models in the random forest ensemble is a decision tree. [12]

Random Forest uses an ensemble method called Bagging also known as Bootstrap Aggregation. In this method, a random sample is chosen from the entire data set and a model is trained independently for every subset. The data for the samples are chosen with replacement. When all the models are trained, the results are combined and final output is based on majority voting. [13] If the parameter 'bootstrap' is set to True(default) then the subset sample size is defined by the parameter max_samples, otherwise the whole data set is used to build each tree. [14]

C. LGBM Classifier

LGBM or Light gradient-boosting machine, as the name suggests is a gradient boosting framework for machine learning and is based on decision tree algorithms. The way how boosting algorithms work is, models are built sequentially and these models try to reduce the errors of the previous model. [15] One of the major differences between LGBM and other gradient boosting algorithms lies in the way the trees are constructed. Most GBM implementations follow a Level-wise tree growth whereas LGBM implements a leaf-wise tree growth strategy. The loss in leaf-wise strategy tends to be lower in comparison to that of level-wise strategy but it might cause some over-fitting issues on a smaller data set. [16]

IV. Experimental Setup

This section covers the steps of data pre-processing, dealing with categorical features and encoding it, analysis of the numerical features and feature analysis. The process of applying the classification algorithms has also been discussed.

The experiment uses a Supervised machine learning model. Figure 1 provides a general idea on how the model works for this experiment.

For this project, Python programming language has been used with many other packages like Numpy, Pandas and Scikit-Learn to state some. All the coding has been done on Jupyter Notebook.

Firstly, the packages important for the project was imported. The data set was in '.csv' format. To load the data into the environment, the Pandas library was imported and the data was stored on a pandas dataframe. For creating meaningful

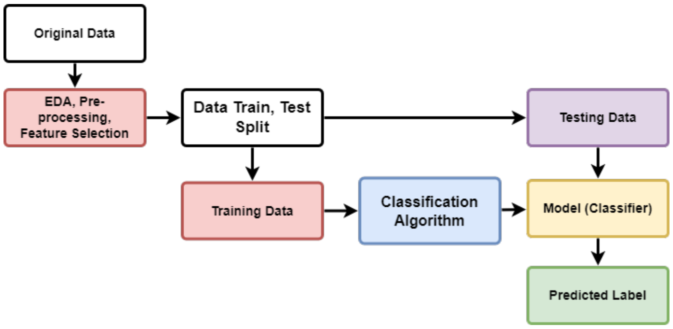


Fig. 1. Basic Working of Machine Learning Algorithm

plots and visualizations, the Matplotlib and Seaborn library was used.

The data set when accessed was already clean with no missing values, so the data set was fine to use for further processes. As already shown in Table I, there were many categorical as well as numerical data. Distribution plots for various numerical features were created in order to see if they followed a Normal Distribution. The boxplot of the numerical features was also plotted to see if they had any extreme values.

For further processing and to use the data to create a machine learning model, the features need to be converted to numerical values. So the features which had categorical and text values were converted to binary form using the get_dummies() function from the Pandas library. This process is known as one hot encoding. At this point, the target variable, 'NObeyesdad' was not converted or encoded and was left as it is.

	Gender
1	Male
2	Female
3	Female
4	Male

→

	Gender_male	Gender_female
1	1	0
2	0	1
3	0	1
4	1	0

The table above shows a simple example on how one hot encoding works using the get_dummies() function.

After completing these steps, feature scaling was carried out. All the features except the target label column was then scaled using the MinMaxScaler() function. The MinMaxScaler scales the data to a range which is specified by the feature_range parameter of the function. The feature_range was set to be the default range value of (0,1).

The label variable 'NObeyesdad' was then encoded using the Label Encoder function from the sklearn library. The label encoder function encodes the categorical values to numerical values ranging from 0 to number of classes - 1. In the case of this experiment, the values range from 0 to 6 as there were 7 classes.

Next, the data was split into training and testing sets using the train_test_split method from the sklearn library. The data was split into 70% of training data and remaining 30% to testing data. After that, the machine learning algorithms, namely KNeighborsClassifier, RandomForestClassifier and LGBMClassifier were imported to carry out the model training phase. Performance metrics like accuracy_score, classification_report, confusion_matrix were also imported from the sklearn library.

The model fitting and training phase was then started, the results of which, is discussed in the sections below. The feature importance was also measured to check for what features played a vital role.

V. Results

All the machine learning algorithms were trained using their default parameters. The models were trained on the X_train ad y_train split of the data set. Then the model.predict() function was used to predict the labels of the X_test split.

The main metric used to differentiate and find which model performed the best was accuracy_score. The following table shows the accuracy scores of the different models.

No.	Classification Algorithm	Accuracy
1	KNeighbors	76.97%
2	Random Forest	92.59%
3	LGBM	95.90%

The accuracy score represents the number of correctly classified data instances over the total number of data instances. [17] A high accuracy does not always mean that the model is good. When a dataset is unbalanced, relying only on accuracy as a performance metric can cause one to make ambiguous interpretations. In order to deal with that, other metrics such as precision, recall and f1-score should also be looked at.

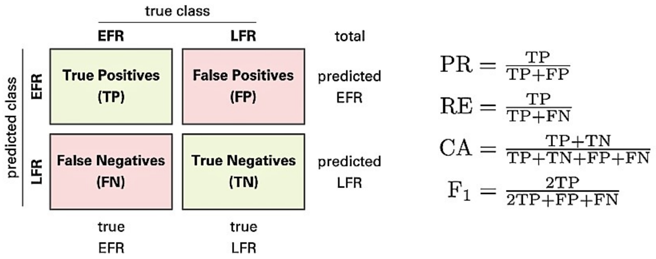

true class
EFR
LFR
total
predicted class
EFR
LFR
True Positives (TP)
False Positives (FP)
predicted EFR
False Negatives (FN)
True Negatives (TN)
predicted LFR
true EFR
true LFR
$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$
$\mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$
$\mathrm{CA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$
$\mathrm{F}_1 = \frac{\mathrm{2TP}}{\mathrm{2TP+FP+FN}}$

Fig. 2. Confusion Matrix [18]

The figure 2 is a figure of a confusion matrix, where PR signifies Precision, RE signifies Recall, CA signifies accuracy and F1 signifies the f-1 score. The respective formulas of these metrics are also provided in the figure.

The Precision value for a good classifier is ideally 1. For precision to become high, the FP (false positive) values must be less or for ideal conditions, 0. Similary, for a good classifier, the Recall value is also ideally 1. Looking at the forumla for Recall from the figure above, for a high recall, the FN (false negative) should be low, or ideally 0. Overall, for a classifier to be said a good one, ideally the values of Precision and Recall should be 1, meaning the false positives and false negatives are 0.

F1-score is such metric which takes into consideration both precision and recall. The f1-score becomes when precision and recall both are high.

$$\text{F1 score} = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}$$

F1 score is the harmonic mean of precision and recall and is a better measure than accuracy. [19]

The tables below show the precision, recall and f1-scores for all the classifier algorithms.

	precision	recall	f1-score	support
0	0.70	0.83	0.76	92
1	0.62	0.48	0.54	77
2	0.75	0.77	0.76	114
3	0.88	0.96	0.92	85
4	0.98	0.99	0.98	92
5	0.72	0.65	0.68	89
6	0.69	0.66	0.67	85
accuracy			0.77	634
macro avg	0.76	0.76	0.76	634
weighted avg	0.76	0.77	0.76	634

TABLE II
PRECISION, RECALL AND F1-SCORE OF KNEIGHBORS

	precision	recall	f1-score	support
0	0.95	0.96	0.95	92
1	0.72	0.92	0.81	77
2	0.99	0.91	0.95	114
3	1.00	1.00	1.00	85
4	1.00	1.00	1.00	92
5	0.96	0.78	0.86	89
6	0.88	0.92	0.90	85
accuracy			0.93	634
macro avg	0.93	0.93	0.92	634
weighted avg	0.93	0.93	0.93	634

TABLE III
PRECISION, RECALL AND F1-SCORE OF RANDOM FOREST

	precision	recall	f1-score	support
0	0.98	0.98	0.98	92
1	0.88	0.94	0.91	77
2	0.97	0.97	0.97	114
3	0.98	1.00	0.99	85
4	1.00	1.00	1.00	92
5	0.94	0.88	0.91	89
6	0.95	0.94	0.95	85
accuracy			0.96	634
macro avg	0.96	0.96	0.96	634
weighted avg	0.96	0.96	0.96	634

TABLE IV
PRECISION, RECALL AND F1-SCORE OF LGBM

The following table sums up the precision, recall and f1-score values for the classifiers.

No.	Classification Algorithm	Precision	Recall	F1 Score
1.	KNeighbors	0.76	0.77	0.76
2.	Random Forest	0.93	0.93	0.93
3.	LGBM Algorithm	0.96	0.96	0.96

TABLE V
PRECISION, RECALL AND F1-SCORE

Upon taking a look at the classification reports from Table II, III and IV it can be seen clearly that LGBM has the best performance out of all. The KNeighbors classifier was the one with the least f1-score among all.

Another metric which is used for measuring the performance of machine learning models is the ROC-AUC curve. ROC (Receiver Operating Characteristic) curve is a graph showing the performance of a classification model at all thresholds. AUC (Area under the ROC curve) measures the area underneath the entire ROC curve. The value of AUC ranges from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0. [20]

The AUC score of the machine learning models was also calculated. The AUC scores of the models are displayed in the table below:

No.	Classification Algorithm	AUC score
1	KNeighbors	0.93
2	Random Forest	0.99
3	LGBM	0.99

TABLE VI
AUC SCORES OF THE MODELS

LGBM and Random Forest Classifier both have a high AUC score of 0.99 which shows that these models are good and highly efficient.

The algorithms that use Decision Trees can also be used to find something called feature importance. The feature importance measure helps to understand, among all the features, which features are actually more helpful than the others. As Random Forest and LGBM both make use of Decision Trees, the feature importance was also performed. Based on Random Forest Classifier, the top 5 most important features were:

1) Weight
2) Height
3) Age
4) FCVC
5) NCP

Similarly, for LGBM Classifier, the top 5 most important features were:

1) Weight
2) Height
3) Age
4) TUE
5) CH2O

The figure showing the feature importance has been added to the appendix.

VI. Discussion and Conclusions

Overall, the experiment provided good results. Based on features comprised of the values of various eating habits and physical conditions of individuals, it can be concluded that, their obesity levels can be estimated. Both the decision tree based algorithms, Random Forest and LGBM Classifier performed the best with f1-scores of 0.93 and 0.99 respectively.

The majority of the data in the data set was synthetically generated and the size of the data set was slightly small. Say that a data is genuine and has a higher number of instances, the results might vary. It might not be totally true that only the features used on this data set are solely responsible for estimating a person's obesity levels. So, having a more diverse and huge data set might be more informative and conclusive. Also, the data set here was collected only from individuals residing in Peru, Colombia and Mexico. So it might be a bit biased as well, as eating habits and physical conditioning can vary all over the world.

References

[1] W. H. Organization *et al.*, "Obesity and overweight," 2017.

[2] E. P. Williams, M. Mesidor, K. Winters, P. M. Dubbert, and S. B. Wyatt, "Overweight and obesity: prevalence, consequences, and causes of a growing public health problem," *Current obesity reports*, vol. 4, pp. 363–370, 2015.

[3] M. Simmonds, A. Llewellyn, C. G. Owen, and N. Woolacott, "Predicting adult obesity from childhood obesity: a systematic review and meta-analysis," *Obesity reviews*, vol. 17, no. 2, pp. 95–107, 2016.

[4] J. Firth, J. E. Gangwisch, A. Borsini, R. E. Wootton, and E. A. Mayer, "Food and mood: how do diet and nutrition affect mental wellbeing?" *bmj*, vol. 369, 2020.

[5] S. Y. Kimm, N. W. Glynn, E. Obarzanek, A. M. Kriska, S. R. Daniels, B. A. Barton, and K. Liu, "Relation between the changes in physical activity and body-mass index during adolescence: a multicentre longitudinal study," *The Lancet*, vol. 366, no. 9482, pp. 301–307, 2005.

[6] T. M. Dugan, S. Mukhopadhyay, A. Carroll, and S. Downs, "Machine learning techniques for prediction of early childhood obesity," *Applied clinical informatics*, vol. 6, no. 03, pp. 506–520, 2015.

[7] S. R. Shaheed, J. A. Malik, and S. Z. N. Hafsa, "Moderating role of physical activity for the psychological determinants of eating behaviors affecting bmi among young adolescents," *Foundation University Journal of Psychology*, vol. 6, no. 1, 2022.

[8] F. M. Palechor and A. de la Hoz Manotas, "Dataset for estimation of obesity levels based on eating habits and physical condition in individuals from colombia, peru and mexico," *Data in brief*, vol. 25, p. 104344, 2019.

[9] "K-nearest neighbors (knn) classification with scikit-learn — datacamp," https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn, (Accessed on 05/05/2023).

[10] "1.6. nearest neighbors — scikit-learn 1.2.2 documentation," https://scikit-learn.org/stable/modules/neighbors.html#classification, (Accessed on 05/05/2023).

[11] What is the k-nearest neighbors algorithm? — ibm. https://www.ibm.com/topics/knn. (Accessed on 05/05/2023).

[12] "Random forest classification with scikit-learn — datacamp," https://www.datacamp.com/tutorial/random-forests-classifier-python, (Accessed on 05/06/2023).

[13] "Random forest algorithms - comprehensive guide with examples," https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/, (Accessed on 05/06/2023).

[14] "sklearn.ensemble.randomforestclassifier — scikit-learn 1.2.2 documentation," https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, (Accessed on 05/06/2023).

[15] "Gradient boosting algorithm: A complete guide for beginners," https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/, (Accessed on 05/06/2023).

[16] "What is lightgbm, how to implement it? how to fine tune the parameters? — by pushkar mandot — medium," https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc, (Accessed on 05/06/2023).

[17] "Confusion matrix, accuracy, precision, recall, f1 score — by harikrishnan n b — analytics vidhya — medium," https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd, (Accessed on 05/06/2023).

[18] "Confusion-matrix-exemplified-cm-with-the-formulas-of-precision-pr-recall-re.png (850×328)," https://www.researchgate.net/figure/Confusion-matrix-Exemplified-CM-with-the-formulas-of-precision-PR-recall-RE_fig1_330174519, (Accessed on 05/06/2023).

[19] "Confusion matrix, accuracy, precision, recall, f1 score — by harikrishnan n b analytics vidhya medium," https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd, (Accessed on 05/06/2023).

[20] "Classification: Roc curve and auc machine learning google developers," https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc, (Accessed on 05/06/2023).

VII. Appendix

Github Link : *https://github.com/manashb21/ObesityEstimation*

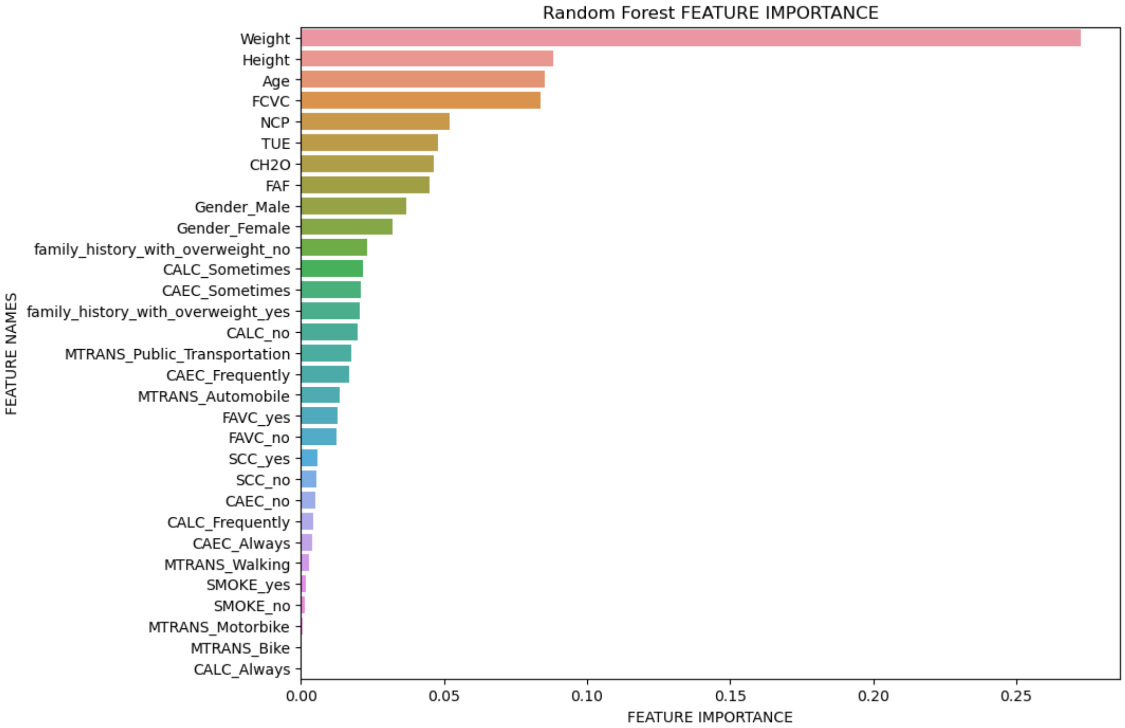


Fig. 3. Random Forest - Feature Importance

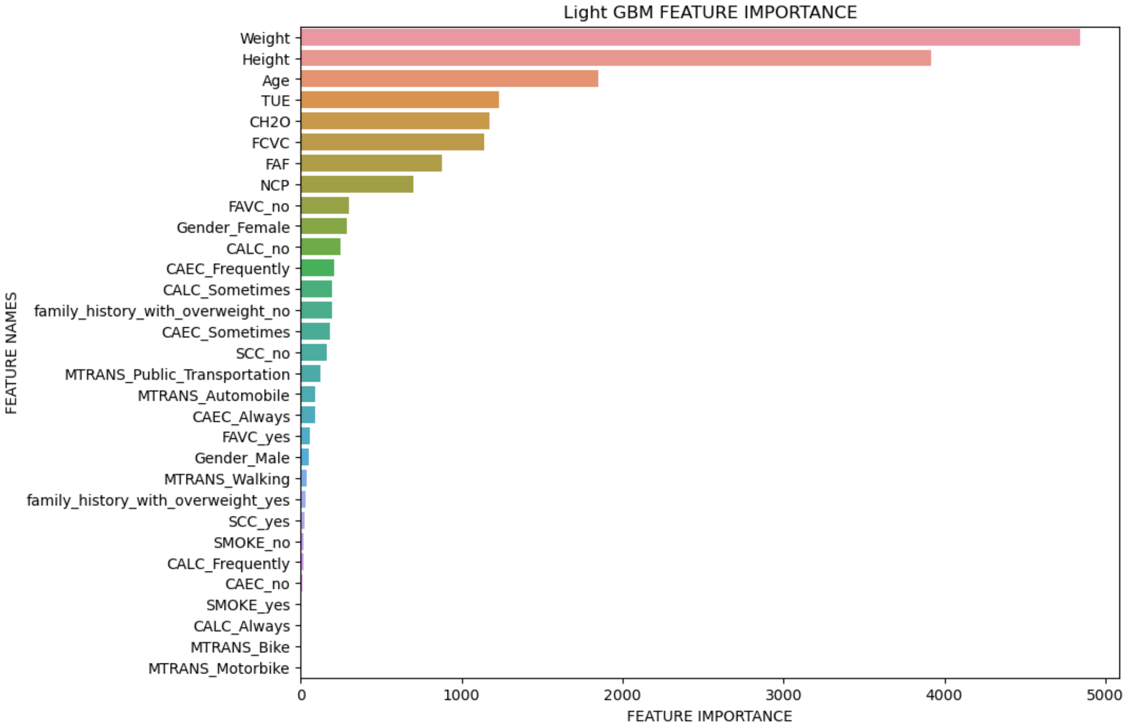


Fig. 4. LGBM - Feature Importance

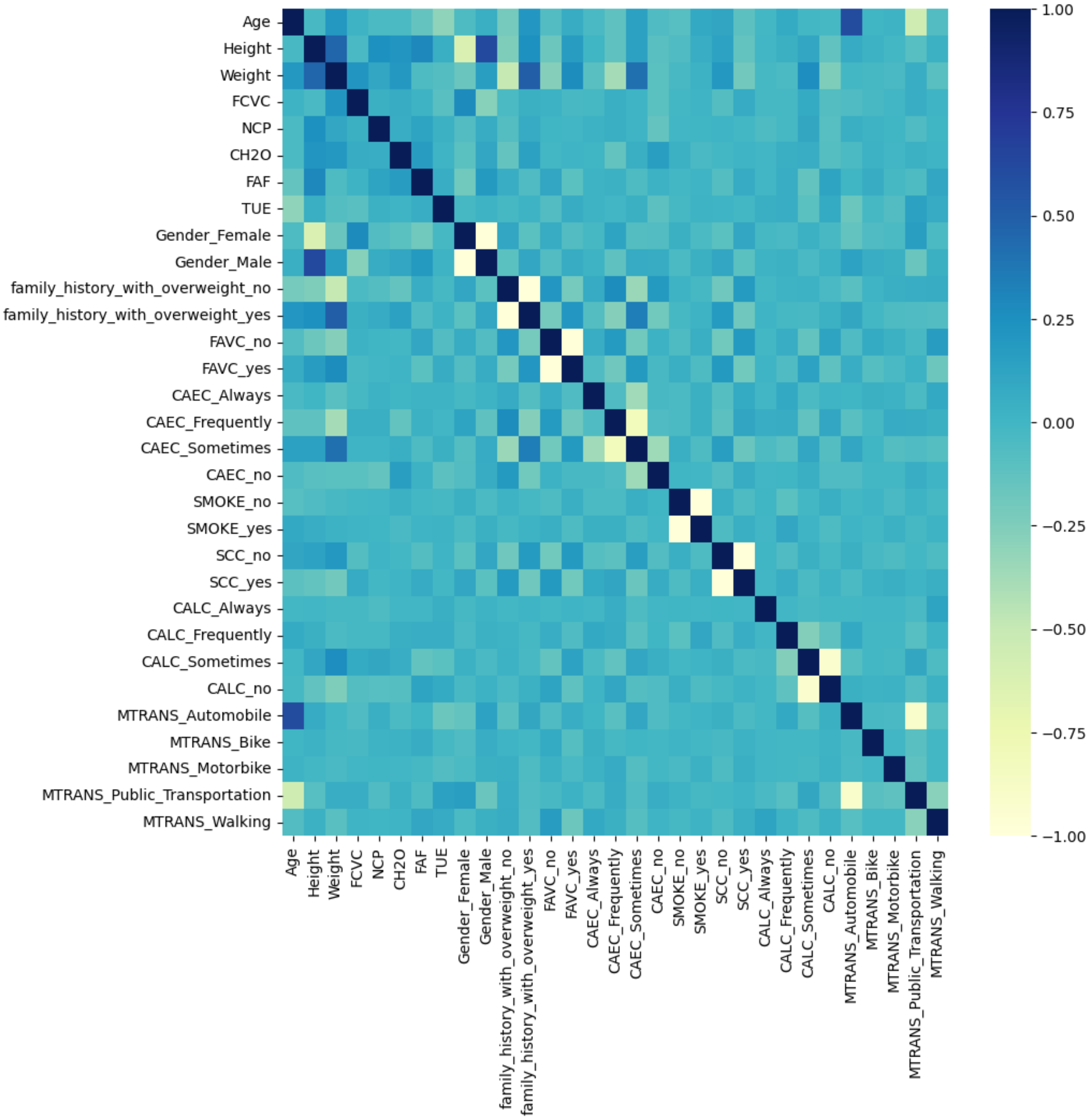


Fig. 5. Correlation Heatmap of Features

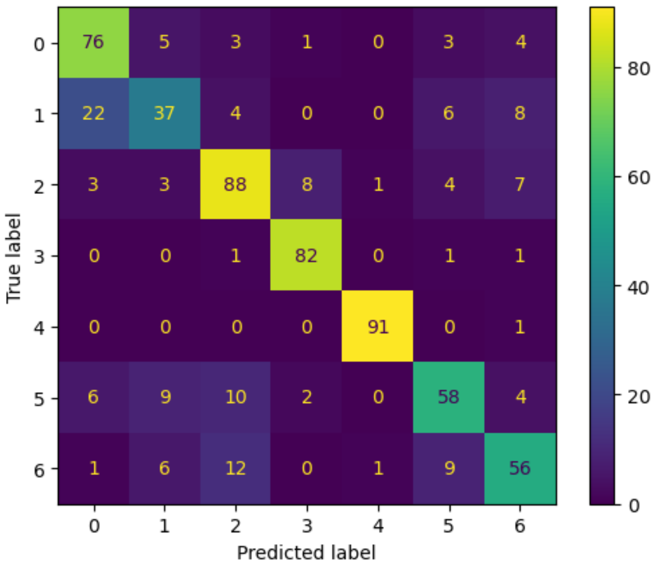


Fig. 6. Confusion Matrix - KNeighbors

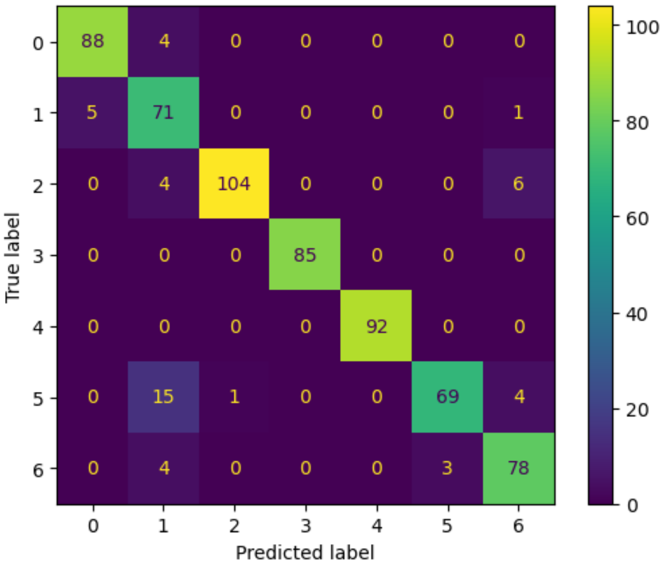
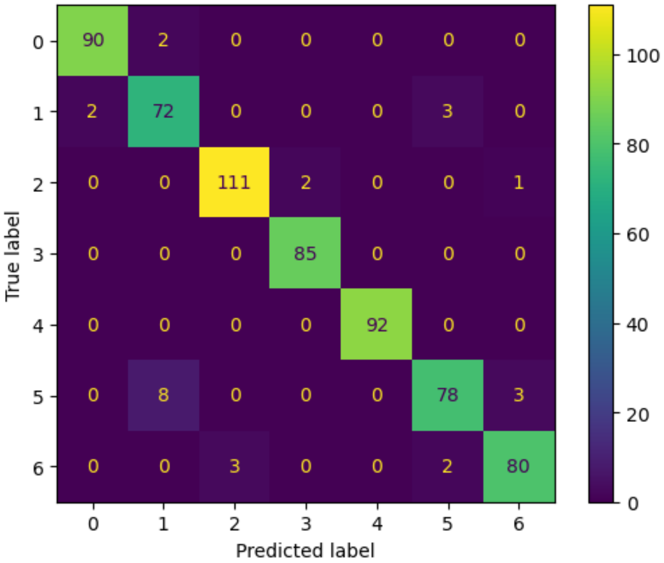


Fig. 7. Confusion Matrix - Random Forest



	0	1	2	3	4	5	6
0	90	2	0	0	0	0	0
1	2	72	0	0	0	3	0
2	0	0	111	2	0	0	1
3	0	0	0	85	0	0	0
4	0	0	0	0	92	0	0
5	0	8	0	0	0	78	3
6	0	0	3	0	0	2	80

Fig. 8. Confusion Matrix - LGBM