ML Project Document

Team 7

Name	Section	Bench No.
Essam Wisam	2	2
Mohamed Saad	2	15
Mariem Mohamed	2	22
Mariem Naser	2	23

Supervised by

Dr. Dina El Reedy

Eng. Mohamed Shawky

May 2023

Table of Contents

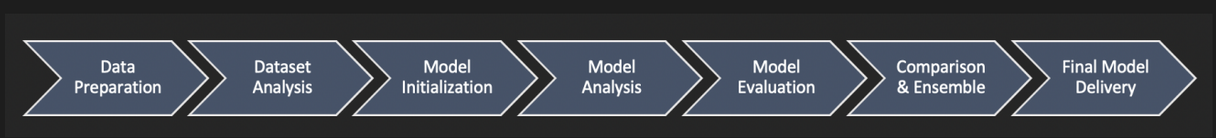
Introduction

Problem Definition

The problem comprises tackling the problem of body level classification. Given features regarding an individual's health such as their height, weight, family history, age, and eleven others; the objective is to predict the body level of the individual (out of four possible levels).

Project Pipeline

Our solution to the aforementioned problem considers the following pipeline



Where the description of each of each stage is as follows

- Data Preparation
  - Flexible data preparation for dataset analysis and all the models
- Dataset Analysis
  - Various analyses are carried out here on the dataset to study the underlying target function and shed light on suitable models to initialize
- Model Initialization
  - Initializing various models on the dataset
- Model Analysis
  - Tuning & analyzing the performance of different models
- Model Evaluation
  - Evaluating the final forms of different models to prepare for comparison
- Comparison & Ensemble
  - Comparing different models and forming an ensemble that surpasses their performance
- Final Model Delivery
  - Deploying the final ensemble model

The following section shows the underlying folder structure. Note that:

- Notebooks are purely for demonstration and tuning; meanwhile, programming logic is purely in the .py files
- Saved and Quests folder to save models and log experiments.

Folder Structure

```
.
├── DataFiles
│   ├── dataset.csv
│   ├── train.csv
│   └── val.csv
├── DataPreparation
│   ├── CovarianceAnalysis.py
│   ├── DataPreparation.ipynb
│   └── DataPreparation.py
├── HandleClassImbalance
│   ├── HandleClassImbalance.ipynb
│   ├── HandleClassImbalance.py
├── ModelBaselines
│   └── Baseline.ipynb
├── Model Pipelines
│   ├── AdaBoost
│   │   └── Adaboost.ipynb
│   ├── Bagging
│   │   ├── Analysis.ipynb
│   │   └── SVMBagging.ipynb
│   ├── LogisticRegression
│   │   ├── Analysis.ipynb
│   │   └── LogisticRegression.ipynb
│   ├── Perceptron
│   │   ├── Analysis.ipynb
│   │   └── Perceptron.ipynb
│   ├── RandomForest
│   │   ├── Analysis.ipynb
│   │   └── RandomForest.ipynb
│   ├── SVM
│   │   ├── Analysis.ipynb
│   │   └── SVM.ipynb
│   ├── StackingEnsemble
│   │   └── StackingEnsemble.ipynb
│   ├── VotingEnsemble
│   │   └── VotingEnsemble.ipynb
│   ├── ModelAnalysis.py
│   ├── ModelVisualization.py
├── ModelScoring
│   └── Pipeline.py
├── References
│   └── ML Project Document.pdf
├── Saved
├── Quests
├── README.md
└── utils.py
```

Data Preparation

Data preparation involves reading the data and putting in a suitable form. Options suitable for this stage beyond reading the data

- To read specific splits of the data (by default train)
- To read only columns of numerical or categorical types (or both)
- Label, one-hot or frequency encoding for categorical features
- To standardize the data

This module was used to ingest the data for all subsequent models and analysis.

Dataset Analysis

In light of guiding model initiation and studying the target function we have performed the following analyses

Features Involved

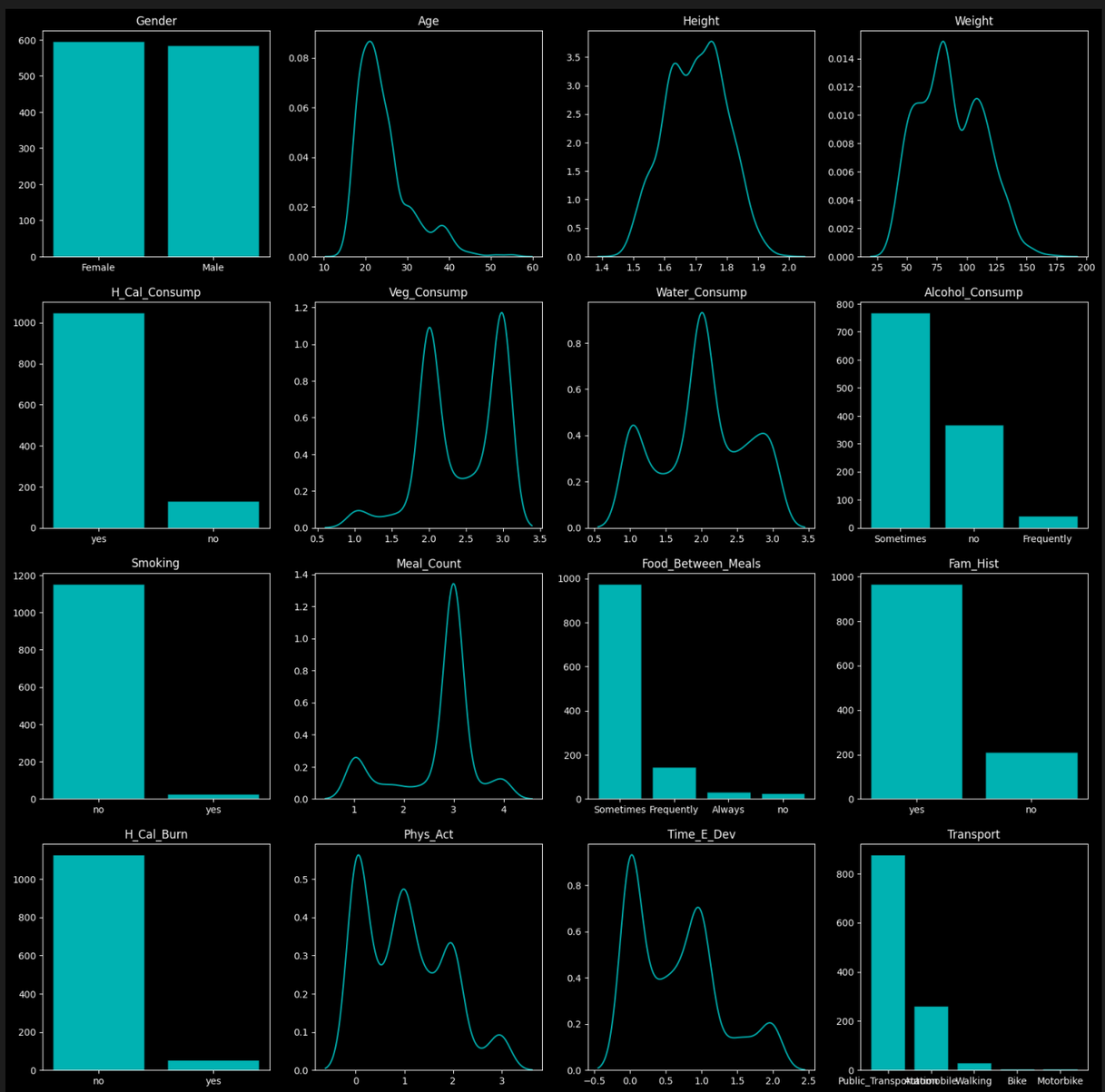
Gender	Age	Height	Weight	H_Cal_Consump	Veg_Consump	Water_Consump	Alcohol_Consump

Smoking	Meal_Count	Food_ Between_ Meals	Fam_ Hist	H_Cal_Burn	Phys_Act	Time_E_Dev	Transport

Basic Counts

Number of samples	Number of features	Number of classes
1180	16	4

Feature Distributions



Number of unique values of each feature

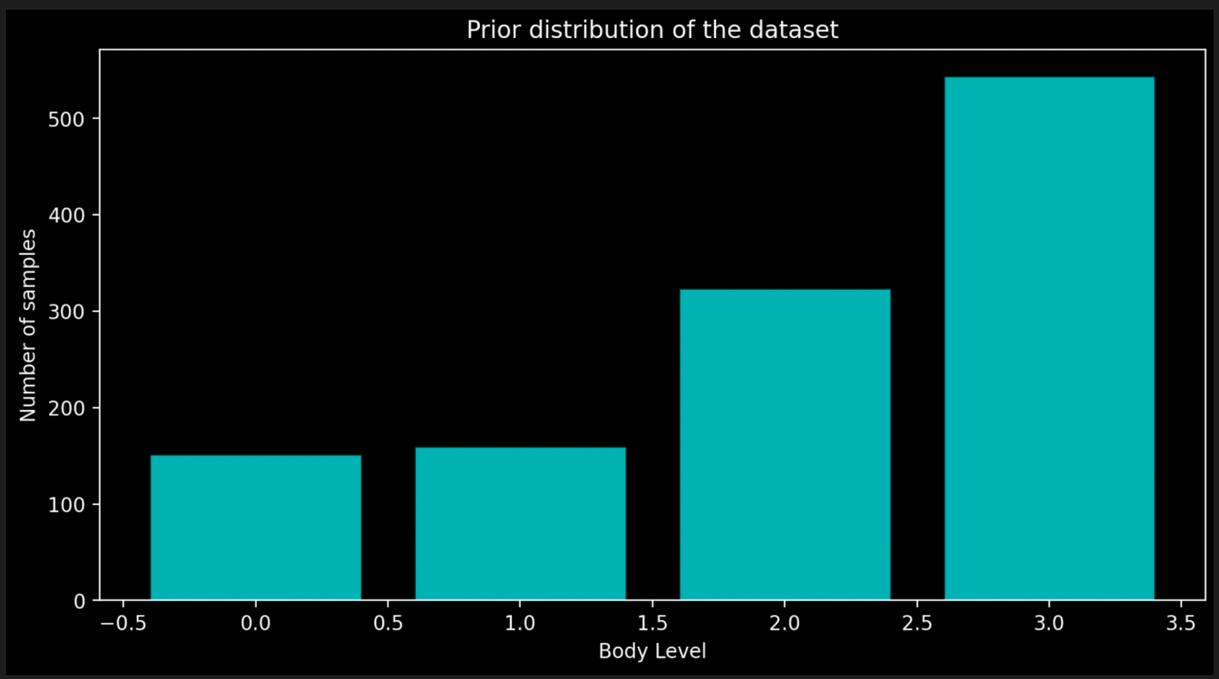
Gender	H_Cal_Consump	Smoking	Fam_Hist	H_Cal_Burn	Alcohol_Consump	Food_Between_Meals	Transport
2	2	2	2	2	3	4	5

Age	Height	Weight	Veg_Consump	Water_Consump	Meal_Count	Phys_Act	Time_E_Dev
numeric	numeric	numeric	numeric	numerical	numeric	numeric	numeric

Insights

- ✦ Most categorical features besides gender suffer high imbalance. Especially, H_Cal_Cons, H_Cal_Burn, Smoking which isn't good news for their usefulness
- ✦ Most continuous distributions are multimodal
- ✦ There don't seem to be any outliers

Prior Distributions



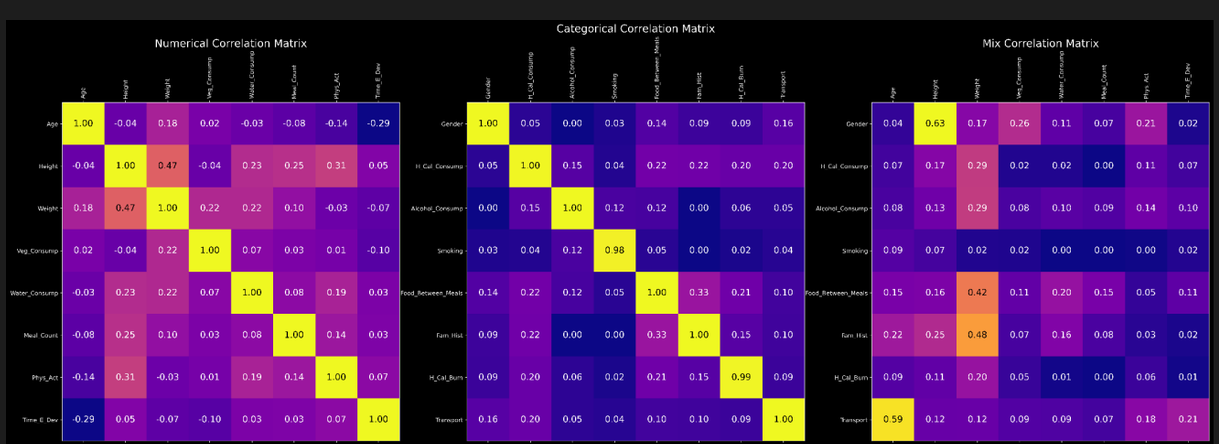
Number of samples in each class

Class 0	Class 1	Class 2	Class 3
152	160	324	544

Insights

✦ There is a clear imbalance in the number of classes. Many models can be sensitive to such imbalance.

Analyzing Feature Correlations



Insights

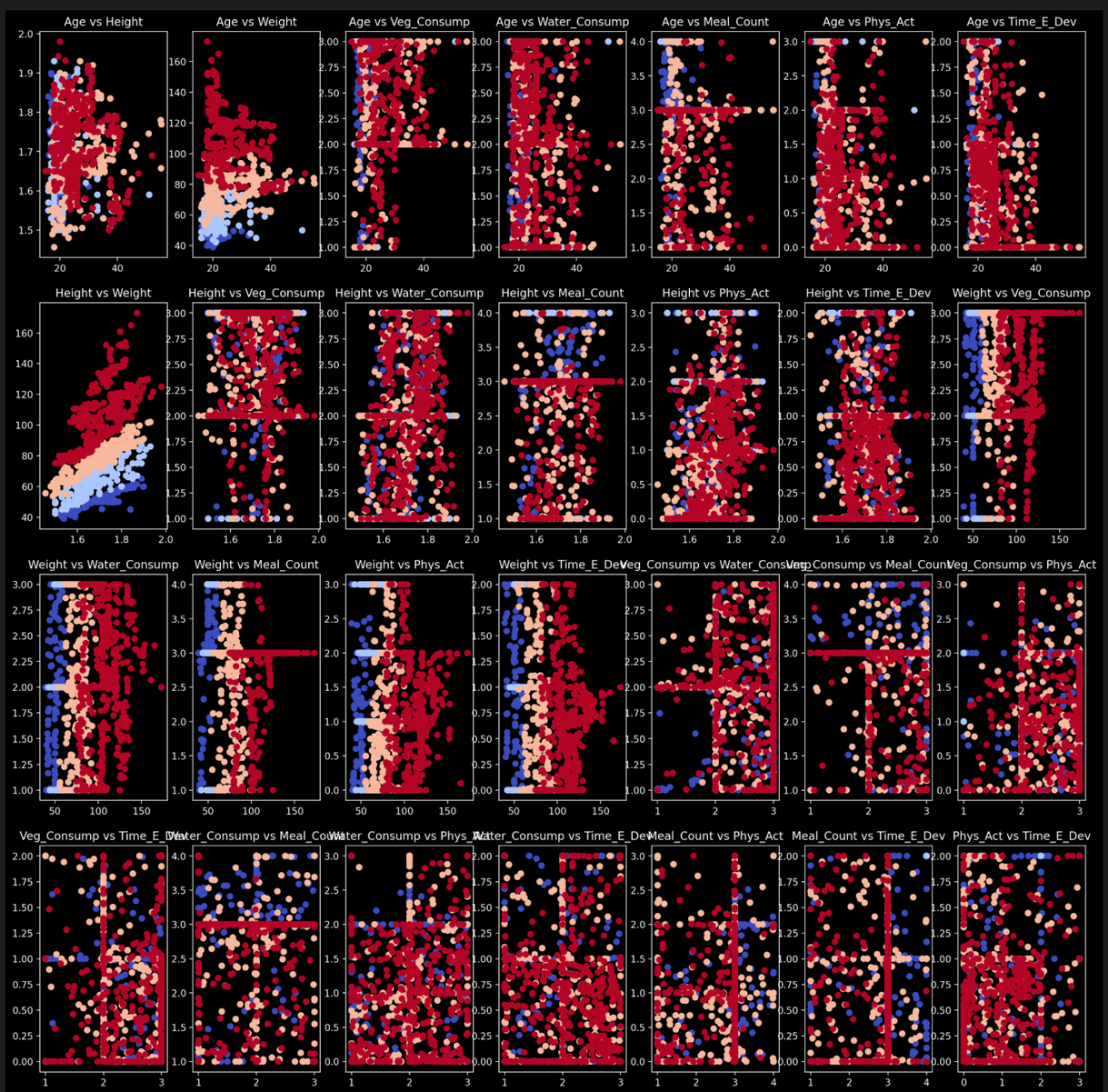
✦ No serious correlations between numerical features but taller people tend to be heavier and tend to do more exercise

✦ Categorical features are mostly uncorrelated

✦ There is a correlation between Age and the transport used. Also between weight and family history and food between meals

It may be noted that the correlation ratio (by Pearson) was used to associate numerical and categorical features. Meanwhile, categorical features were associated together using the Chi-Square test.

Separability of Numerical Features

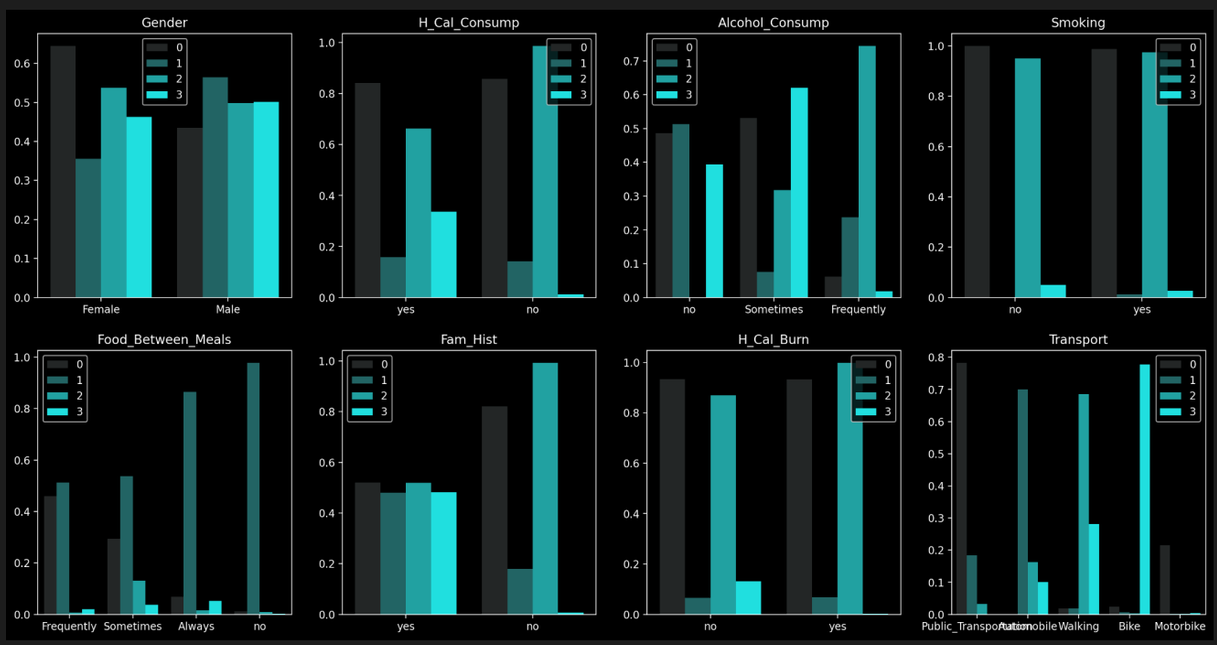


Insights

- ✦ Most continuous feature pairs don't linearly separate the data
- ✦ Weight seems to have the nicest interaction with other features and a spectacular interaction with height
- ✦ Linear models should stand a chance

Note that we will still consider feature selection for each model; we'll let the model make the decision for itself regarding the features to keep.

Separability of Categorical Features



Insights

✦ Categorical variables such as gender and smoking don't seem too relevant to the target variable

✦ Distributions strongly vary for Food_Between_Meals and Transport between the two classes

✦ Although it may seem that we can draw decision rules such as 'if Food_Between_Meals > a then class = 1', the support of the group is too small for it to be statistically significant as shown above.

Generalization & Validation Set Size

We worked on a module to help us decide an appropriate validation set size. Given two of ϵ, η, P where ϵ is the allowed generalization error, η is the ratio of the validation set and P is the probability of violating the generalization error, this would return the third parameter after applying Hoeffding's inequality.

Example invocation with $\eta = 0.2$ and $\epsilon = 0.06$

Hoeffding's Inequality states:

$$P[|E_{out}(g) - E_{test}(g)| \leq \epsilon] \geq 1 - 2e^{-2N_{test}\epsilon^2}$$

If we use validation set of size $0.2N_{train} = 295$ then with $\epsilon = 0.06$ we have

$$P[|E_{out}(g) - E_{test}(g)| \leq 0.06] \geq 0.761$$

In other words, with probability at least 0.761, the generalization error of our model will be at most 0.06 given a validation set of size 295.

Insights

✦ Dataset sadly doesn't offer serious generalization guarantees due to its size

Models Considered

Baselines

We considered two trivial baselines and another nontrivial baselines so that we can set the bar regarding the bias of further models we consider. The results in terms of training set weighted F1s are as follows

Most Frequent Baseline	Uniform Random Baseline	Gaussian Naive Bayes
0.29	0.285	0.749

Initiated Models

We initiated and analyzed the following models, the first three of which we have covered in the course.

- Support Vector Machines
- Logistic Regression
- Perceptron
- Random Forest
- Adaptive Boosting

All models were moderately successful, except for adaptive boosting. In this, we used the models as implemented by the Sci-Kit learn library.

We later formed ensembles out of these models as will be seen later.

Model Analysis

General Structure

The analysis we carried out on any model was divided into four components at no particular order which are as follows

- Model Greetings
  - Initiating an instance of the model and viewing its hyperparameters
  - Studying the hyperparameters and their importance

- Model Analysis
  - Testing Model Assumptions (if any)
    - For instance, log-linearity for logistic regression

  - VC Dimension Check for Generalization
    - In particular, using the number of parameters in the model to check that it indeed holds that $N \geq 10d_{vc}$

  - Bias Variance Analysis
    - Comparing the model's training error to the best achieved error to study bias
    - Comparing the model's validation error to its training error to study variance

  - Learning Curve
    - Plotting the training error and validation error over different sizes of the dataset to see if the model would benefit from more data to reduce variance
    - Also, to indicate the level of the bias of the model (where the errors converge)

- Hyperparameter Analysis
  - Regularization & Overfitting
    - In this, a validation curve was used to evaluate the model's training and testing error over the range of given hyperparameters
    - From this, we would study a suitable range for the hyperparameter

values to search inside and mark the point after which the model overfits
        - This always involved a regularization hyperparameter if it existed

    - Hyperparameter Search
        - Random Search was used to find the optimal hyperparameters from a given grid over the important hyperparameters for a model
        - They were saved to be seen by the rest of the analysis or the main model file

    - Hyperparameter Logging
        - We used a Python library to log many of our experiments (i.e., any hyperparameter setting whether set by hand or by hyperparameter search and the corresponding metrics for the model)

- Feature Analysis
    - Feature Importance
        - As most models assigned feature importances whether by feature weight or other, we consider studying the importance of each feature given for the model

    - Recursive Feature Elimination
        - As suggested by one of Vapnik's papers
        - This removes the least important feature until a minimum number of features is reached or the metric is no longer improving

- Class Imbalance Analysis
    - Analyzing Different Methods
        - In this, we considered SMOTE and its variants (borderline, SMOTEN, SMOTENC)
    - Analyzing Different Hyperparameters
        - We considered the k hyperparameter and different resampling ratios or class-weight ratios (no resampling)
        - Thereby, we studied different solutions to the inherent class imbalance problem

Perceptron

- ## Model Greetings

  The hyperparameter of interest in this model are the learning rate η and the maximum number of iterations max_iter. Perceptron still converges even if data is not linearly separable due to the second hyperparameter.

- ## Model Analysis

  - ### VC Dimension Check for Generalization

    By estimating the VC dimension of the model, we have $d_{vc} = 37$. Since, $N = 1477$, it holds that

    $$N \geq 10 d_{vc}$$

    Hence, model is expected to have no issues with generalization.

  - ### Bias Variance Analysis

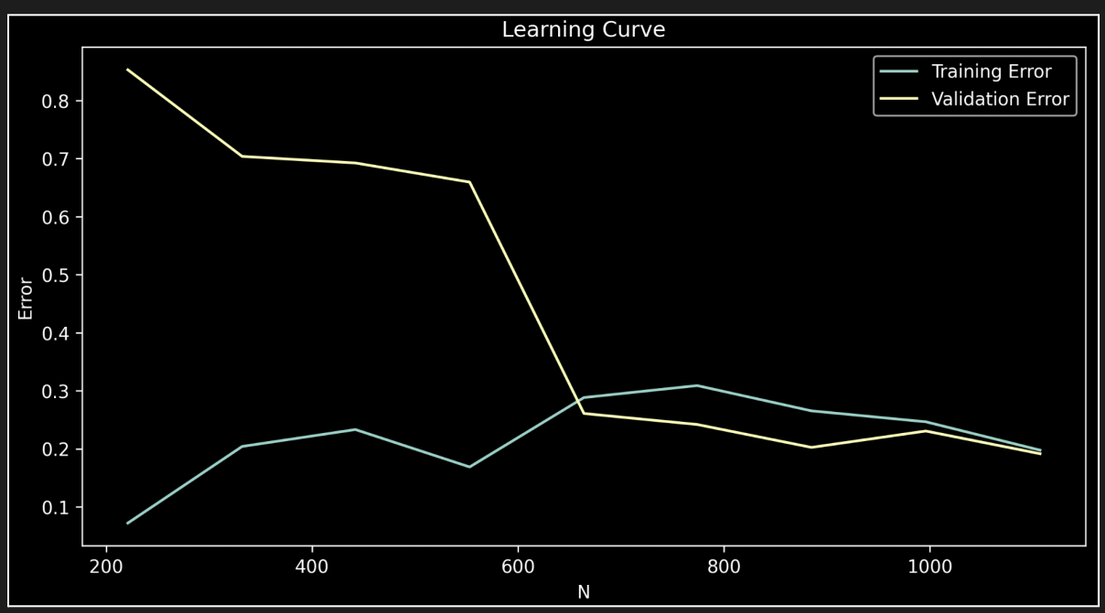
Train WF1	Val WF1	Aviodable Bias	Variance
0.724	0.784	0.276	-0.06

Insights

✦ Variance is too low, seems to be no overfitting.

✦ Bias is slightly high.

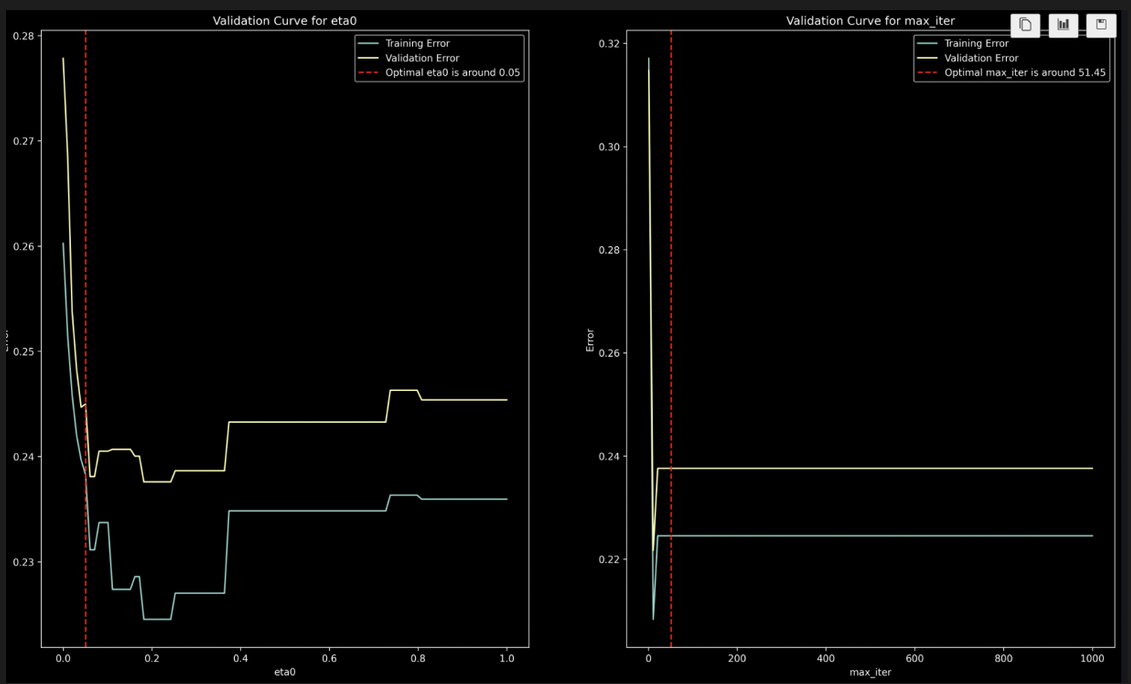
  - ### Learning Curve



Insights

✦ Model seems to have just enough data for E_{in} and E_{out} to converge

- Hyperparameter Analysis
  - Regularization & Overfitting



Insights

✦ Learning rate seems to have an ability to control the performance given a number of iterations (high learning rate can overfit)

✦ Model seems to converge with the default tolerance in as low as 100 iterations

  - Hyperparameter Search

Results from Random Search:

Optimal Configuration

max_iter	eta0	accuracy
334.0	0.24242	0.76301

Insights

✦ May not strongly rely on this due validation set contamination.

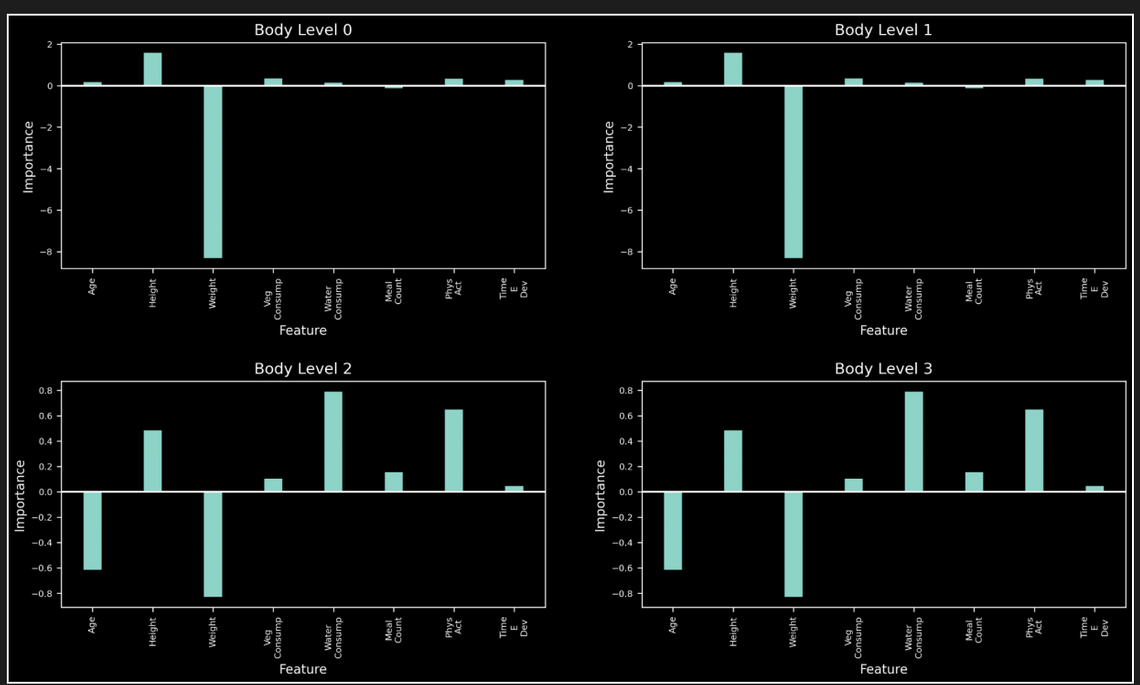
✦ Bias remains higher than usual

  - Hyperparameter Logging

Check the log at the end of Perceptron.ipynb

- Feature Analysis

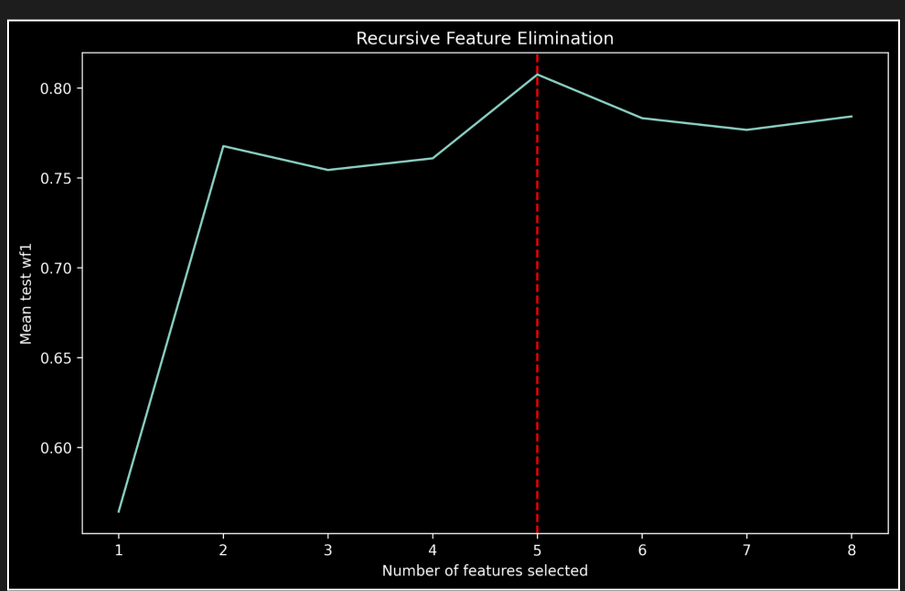
  - Feature Importance



Insights

✦ Weight, Height and Veg_Consump seem to be most relevant to discriminate the classes (esp. the first two)

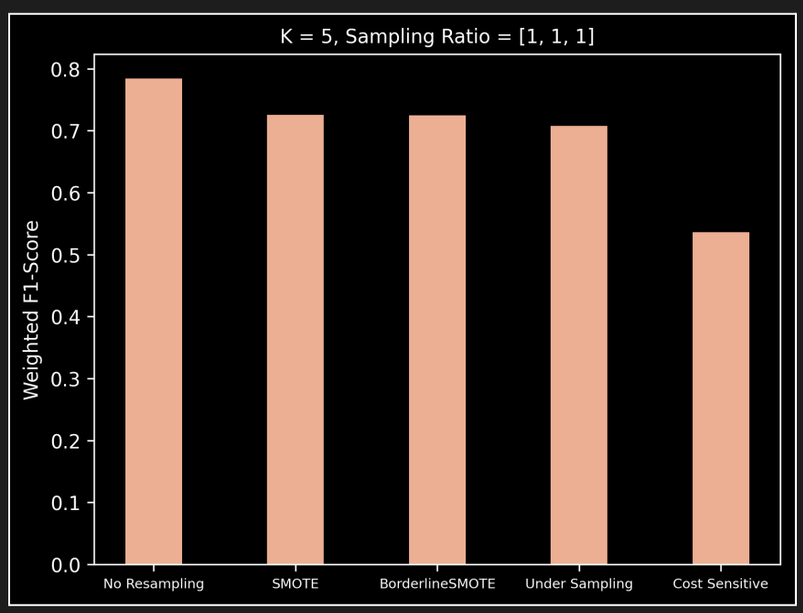
  - Recursive Feature Elimination



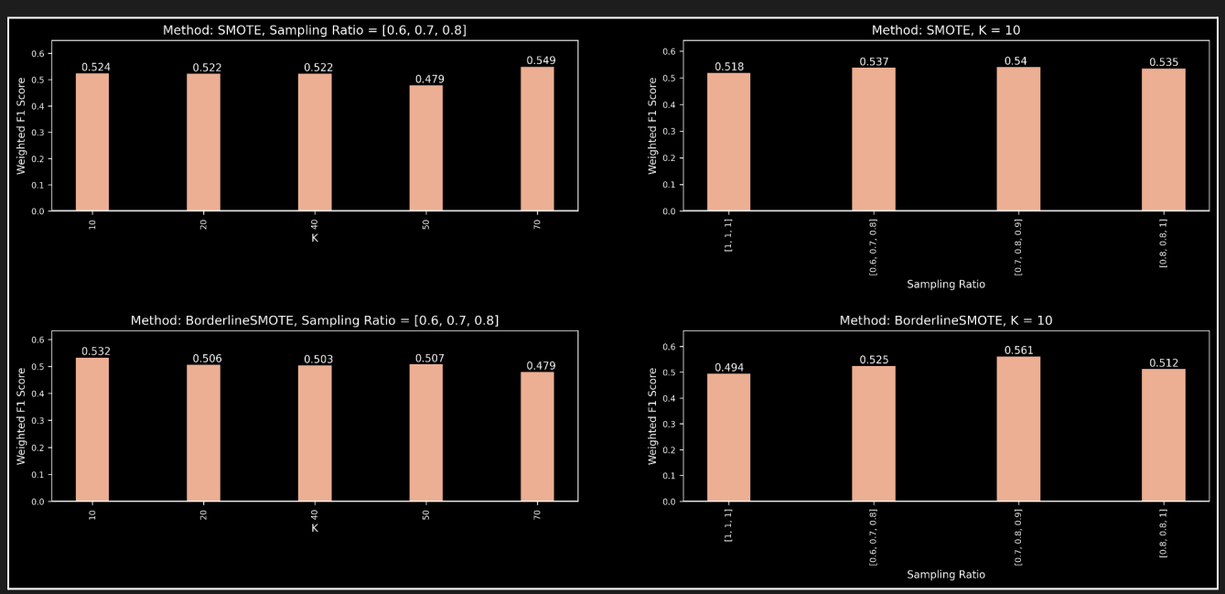
Insights

✦ Features to keep are Weight, Height, Physical Activity, Time_Dev, Water_Consump which matches expectations

- Class Imbalance Analysis
  - Analyzing Different Methods



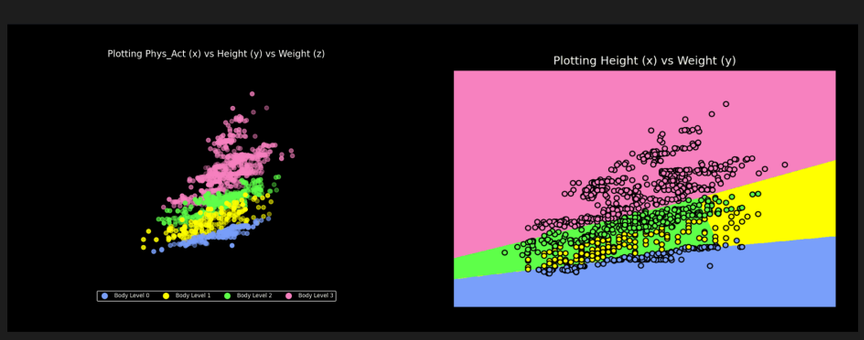
  - Analyzing Different Hyperparameters



Insights

- ✦ Resampling does not seem to help the model (sometimes making it worse as seen)
- ✦ Overall, class imbalance hyperparameters have no strong effect

- Visualization (Bonus)



Insights

- ✦ Poor separability regarding the 2nd and the 3rd class

SVM

- ## Model Greetings

  The hyperparameters of interest in this model are C , kernel and gamma.

- ## Model Analysis

  - ### VC Dimension Check for Generalization

    By estimating the VC dimension of the model, we have $d_{vc} = 55$. Since, $N = 1477$, it holds that

    $$N \geq 10 d_{vc}$$

    Hence, model is expected to have no issues with generalization.

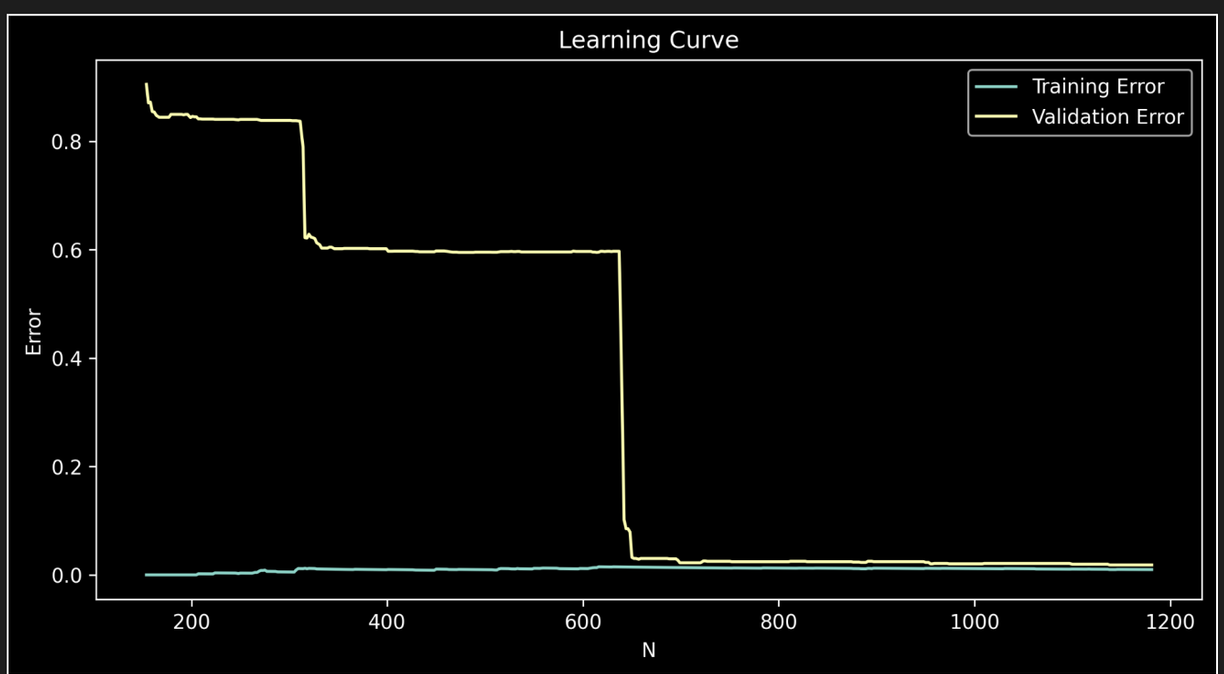
  - ### Bias Variance Analysis

Train WF1	Val WF1	Aviodable Bias	Variance
0.993	0.979	0.007	0.014

Insights

- ✦ Accuracy and WF1 are the same, so the model is not biased or overfitted.
- ✦ Bias is small enough but variance can may be improved a little.

  - ### Learning Curve

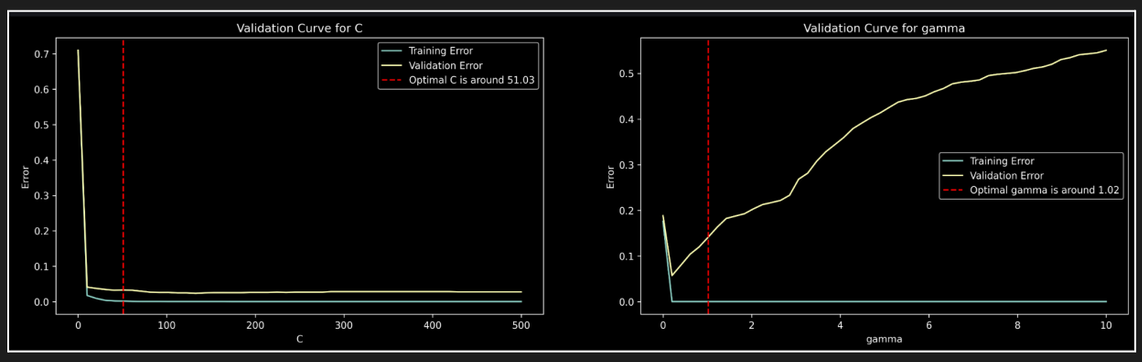


Insights

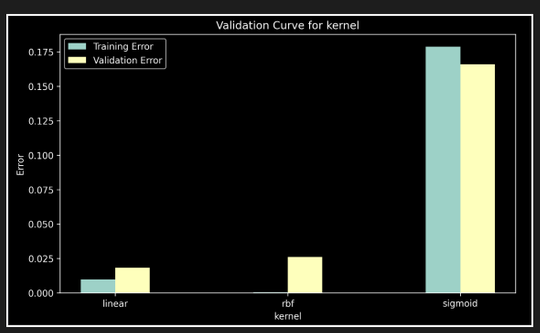
- ✦ Increasing N, make Eout decreasing until converge with Ein.

- Hyperparameter Analysis
  - Regularization & Overfitting

    RBF Kernel



- ✦ Gamma stimulates strong overfitting for the model
- ✦ Increasing C (less regularization) does not seem to encourage overfitting
- ✦ A linear kernel seems the best as shown



  - Hyperparameter Search

    I. Results from Random Search for scoring as the WF1:

Optimal Configuration

C	gamma	kernel	WF1
1186.86	0.55107	linear	0.9856

II. Results from Random Search for scoring as #support_vectors:

Optimal Configuration

C	gamma	kernel	SV
1938	0.25106	linear	38

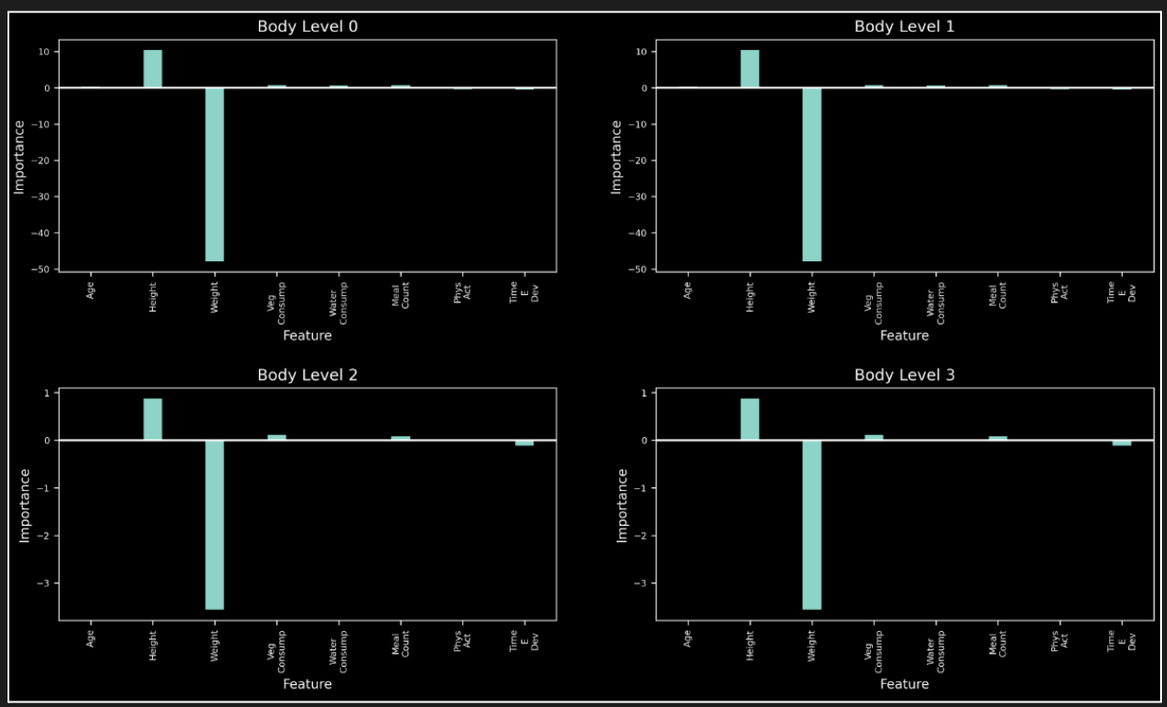
Insights

- ✦ Using sv as the metric result in less WF1.
- ✦ Using WF1 parameters' search result in the best performance of SVM.

  - Hyperparameter Logging

    Check the log at the end of SVM.ipynb

- Feature Analysis
  - Feature Importance

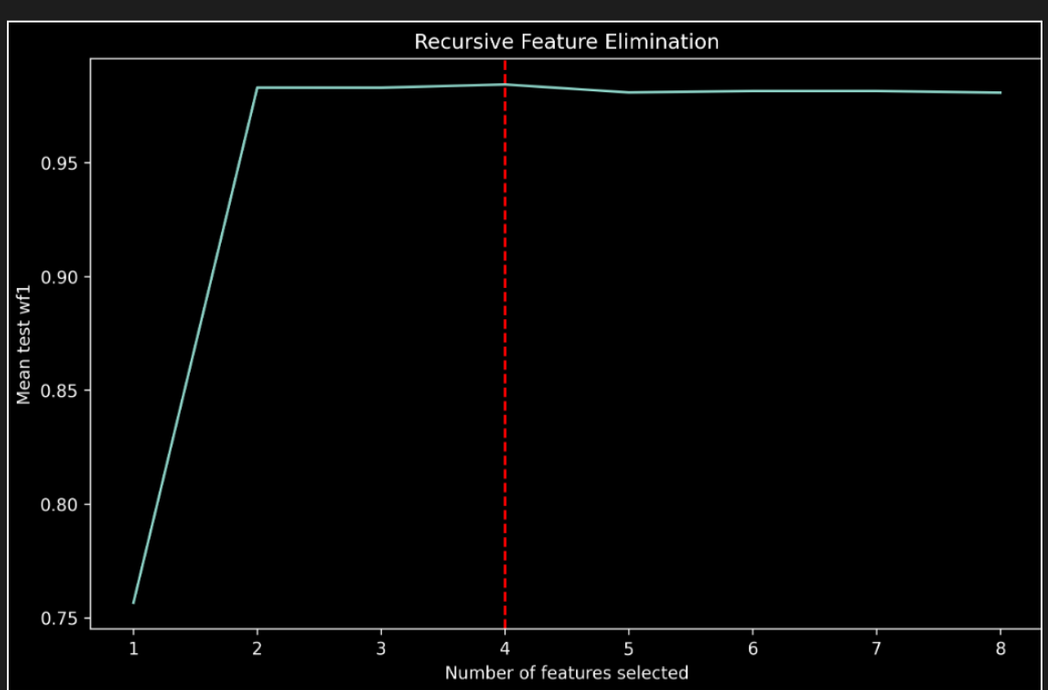


Insights

✦ Weight and height has the highest impact on the model.

✦ Veg Consumption, Meal Count and Time E Dev has less importance for the model.

  - Recursive Feature Elimination

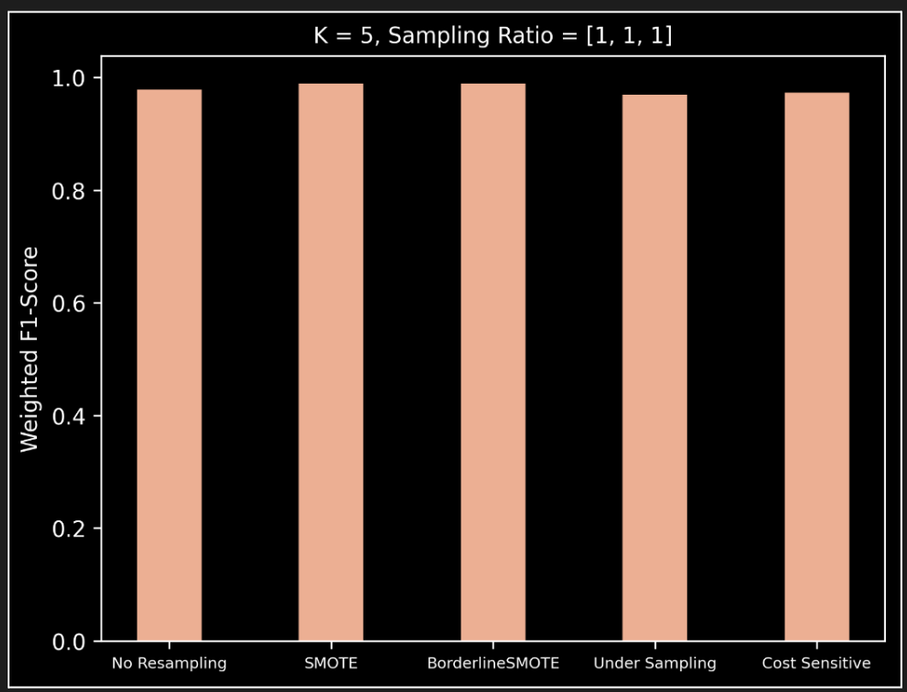


Insights

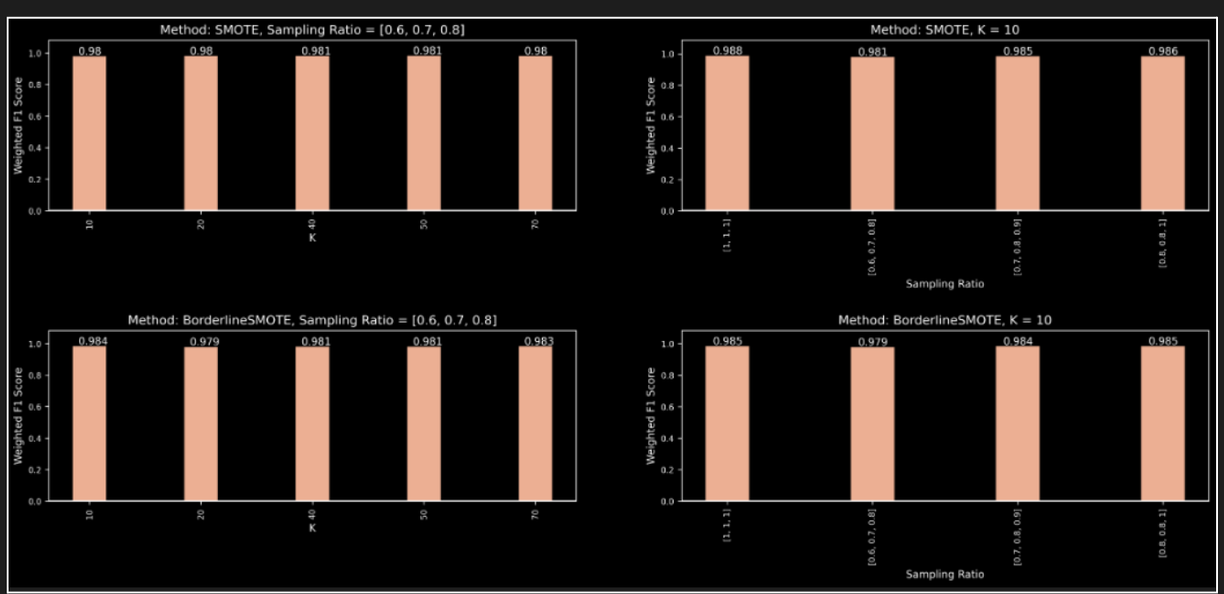
✦ Although Time_E_Dev has some importance, but elimination of it increase the accuracy.

✦ except for Time_E_Dev, Results do match expectations.

- Class Imbalance Analysis
- Analyzing Different Methods



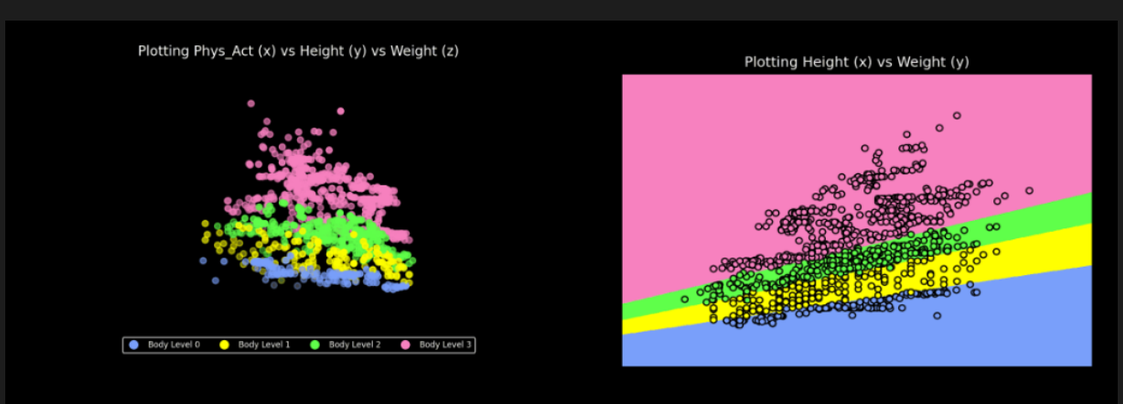
- Analyzing Different Hyperparameters



Insights

✦ Class imbalance seems to add no large benefit to the model

- Visualization (Bonus)



Logistic Regression

- ## Model Greetings

Hyperparameters

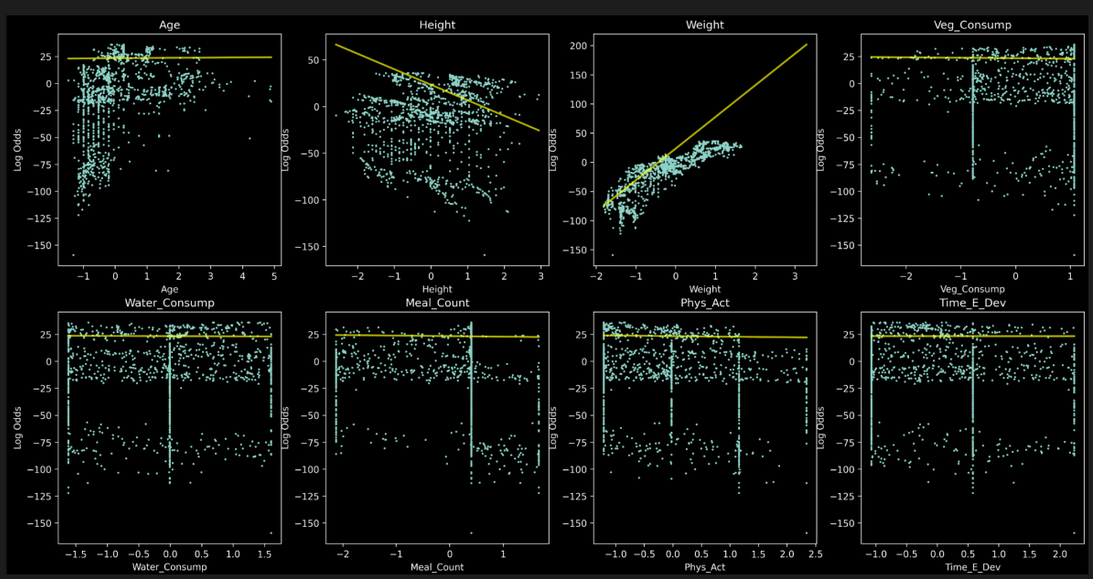
C	class_weight	dual	fit_intercept	intercept_scaling	l1_ratio	max_iter	multi_class	n_jobs	penalty	random_state	solver	tol
140.745	balanced	False	True	1	None	100	multinomial	None	l2	None	newton-cg	0.0001

Insights

- ✦ Interesting hyperparameters are C (Less Regularization), Solver, Multi-Class Strategy
- ✦ Not so interested in L1 as feature selection will be done later
- ✦ Solver should not be so important as the logistic is convex
- ✦ Likewise, small dataset means not to worry much about max iteration or tolerance

- ## Model Analysis

Testing Model Assumptions



Insights

- ✦ Almost all features fail the log-linearity test except for the third
- ✦ It will turn out that this is almost all we need, the rest won't be important

- ### VC Dimension Check for Generalization

By estimating the VC dimension of the model, we have $d_{vc} = 37$. Since, $N = 1477$, it holds that

$$N \geq 10 d_{vc}$$

Hence, model is expected to have no issues with generalization.

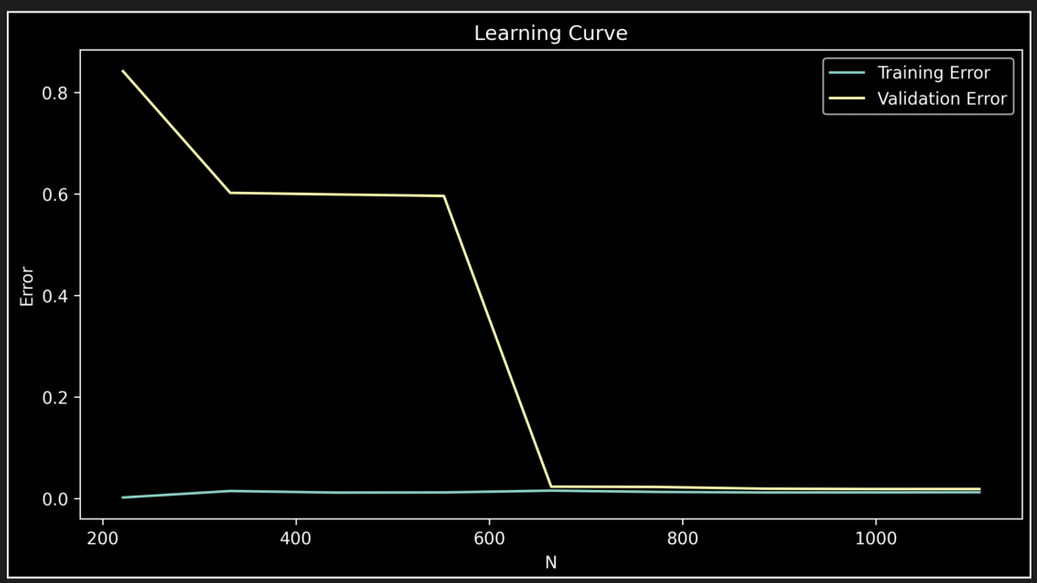
- ### Bias Variance Analysis

Train WF1	Val WF1	Avoidable Bias	Variance
0.986	0.981	0.014	0.005

Insights

- ✦ Accuracy and WF1 are largely agreeing
- ✦ Bias is fair but variance can may be improved a little

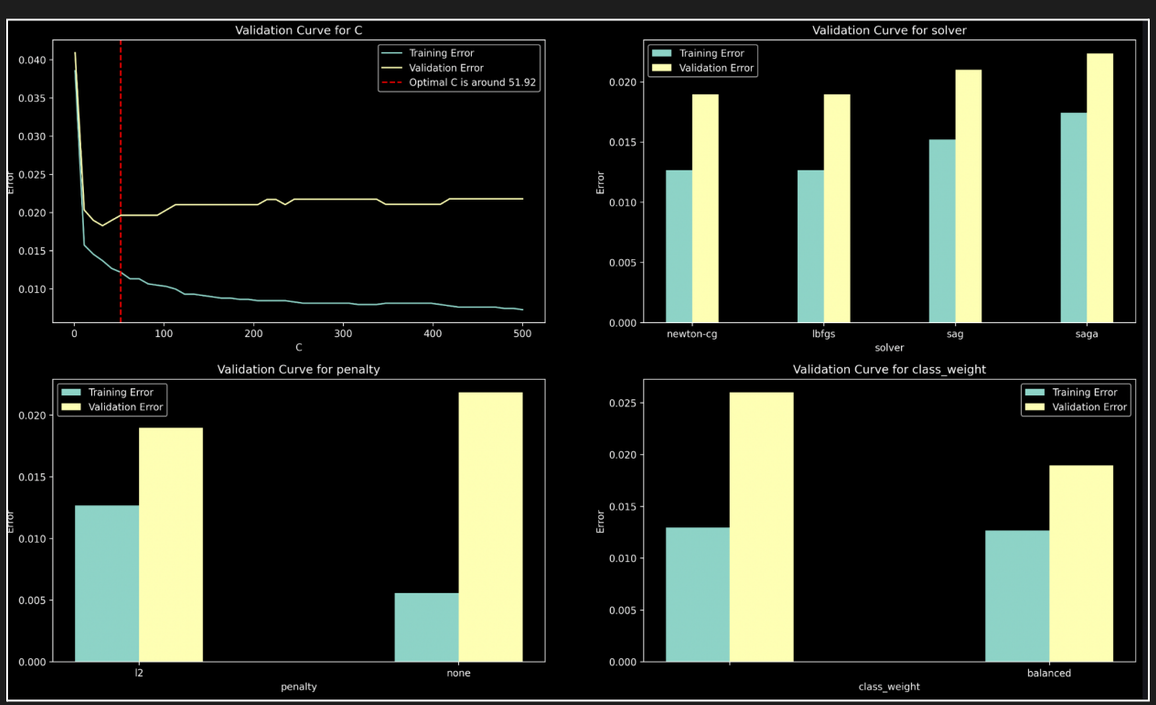
- Learning Curve



Insights

- ✦ As can be seen, bias is naturally low (compared to some other models)
- ✦ Data seems sufficient to bring variance to a stably low value

- Hyperparameter Analysis
  - Regularization & Overfitting



Insights

- ✦ Increasing model complexity via C does not result in extreme overfitting; model is inherently simple
- ✦ As expected different solvers do not affect the model much
- ✦ No regularizing (very large C) does not seem as good as some regularizing
- ✦ Class weighting does improve the model performance

- Hyperparameter Search

  Results from Random Search:

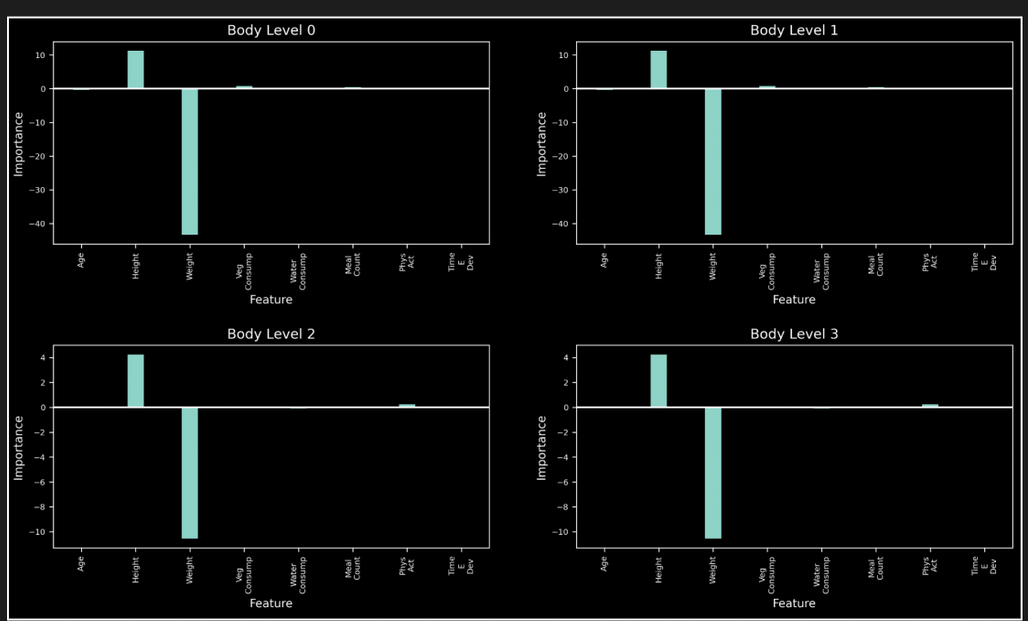
  Optimal Configuration

C	class_weight	multi_class	penalty	solver	WeightedF1
40.074	balanced	multinomial	l2	newton-cg	0.98104

- Hyperparameter Logging

  Check LogisticRegression.ipynb

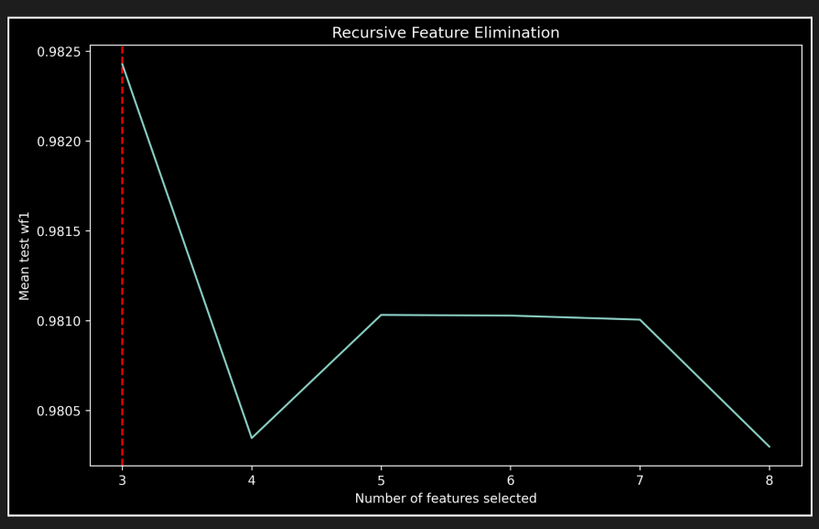
- Feature Analysis
  - Feature Importance



Insights

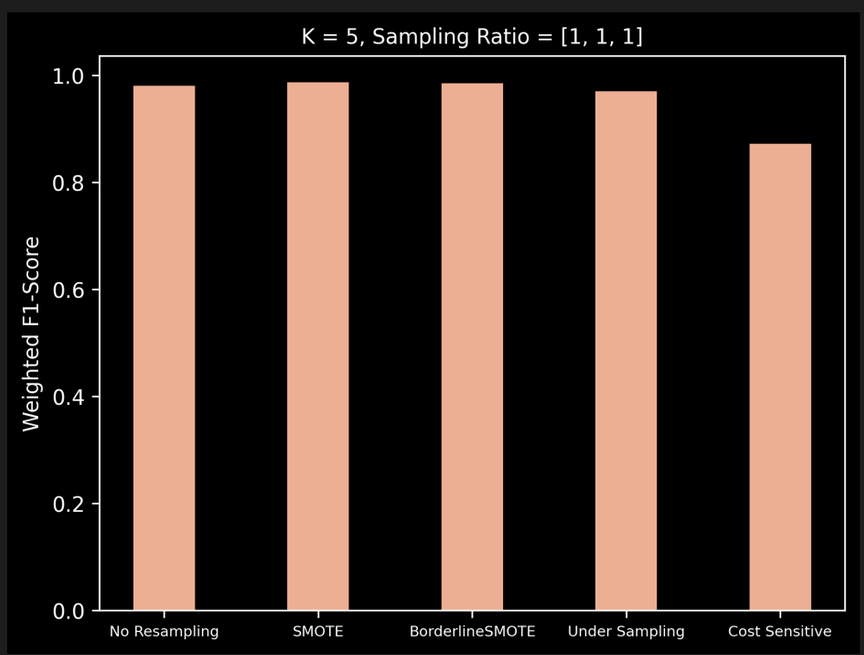
✦ Only weight and height seem to be most relevant to discriminate the classes

- Recursive Feature Elimination

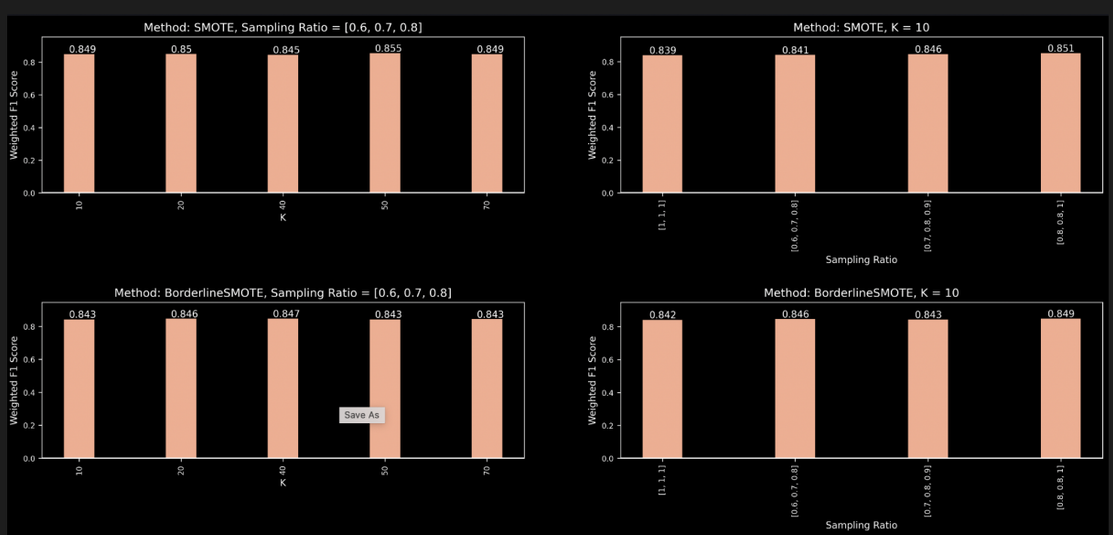


The three features to keep are Weight, Height and Veg_Consump.

- Class Imbalance Analysis
  - Analyzing Different Methods



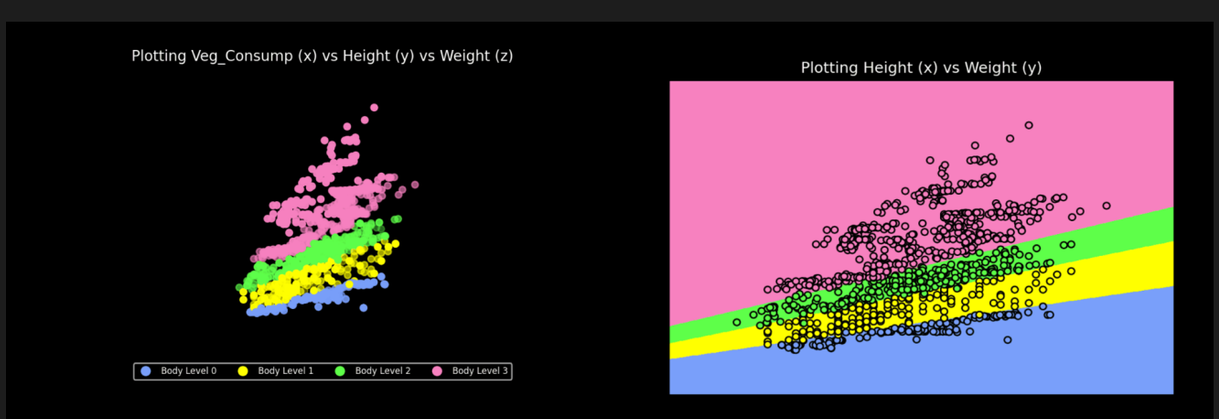
  - Analyzing Different Hyperparameters



Insights

✦ Class imbalance seems to add no large benefit to the model

- Visualization (Bonus)



Random Forest

- ## Model Greetings

  ### Hyperparameters

bootstrap	max depth	max features	min samples split	criterion	n estimators	min samples leaf
True	12	log2	3	entropy	100	1

Insights

✦ The most interesting hyperparameters are n estimators, max_depth, min_samples_split and n estimators.

- ## Model Analysis

  - ### VC Dimension Check for Generalization

By estimating the VC dimension of the model, we have $d_{vc} = 103700$. Since, $N = 1477$, here it holds that that

$$N < 10d_{vc}$$

Hence, generalization is not guaranteed and its advised to reduce the model complexity.

Insights

✦ The number of parameters here are not a good estimate of the effective number of parameters (VC dimension) as the Random Forest uses ensembles of decision trees.

  - ### Bias Variance Analysis

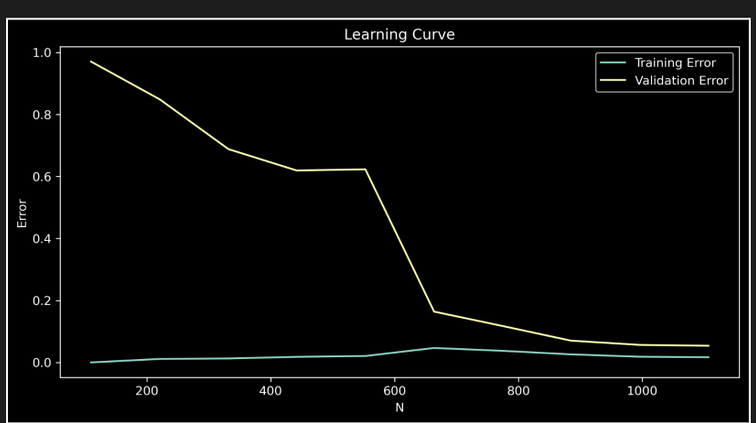
BV Analysis Using WF1

Train WF1	Val WF1	Aviodable Bias	Variance
1.0	0.973	0.0	0.027

Insights

✦ There is a zero bias but the variance is slightly high which suggests overfitting

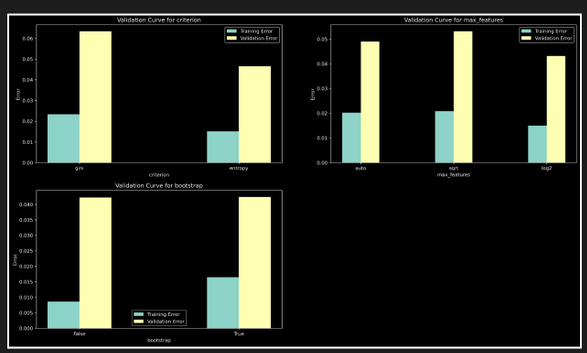
  - ### Learning Curve



Insights

✦ It is as expected the training error is getting slightly worse as the number of training samples increases and the validation error is getting better as the number of training samples increases.

- Hyperparameter Analysis
  - Regularization & Overfitting



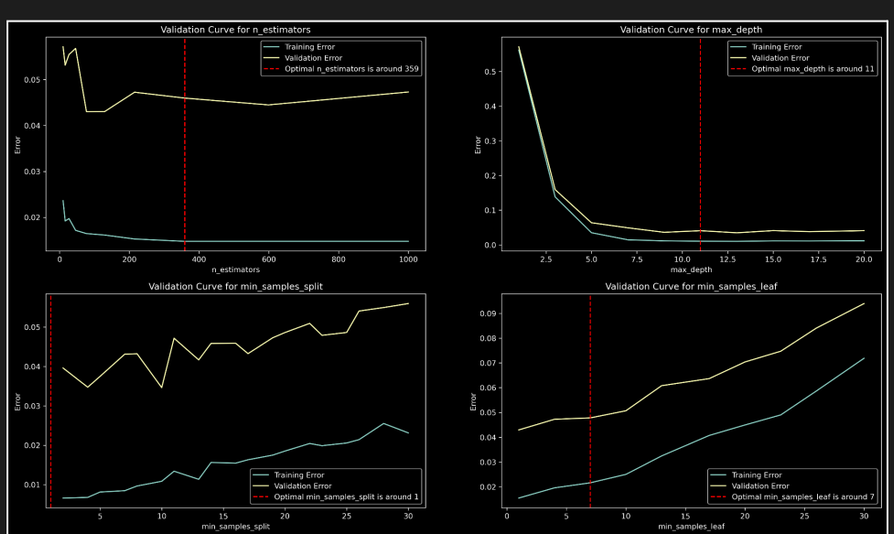
Insights

✦ criterion: entropy seems to be the best as it has the best validation error and better generalization.

✦ max_features: log2 seems to be the best as it has the best validation error and better generalization.

✦ bootstrap: True seems to be the best as it has the best validation error and better generalization.

Insights



✦ n_estimators: The main ensemble parameter. Generally the more the better (but more computations are needed), but in practice the performance settles after some point.

✦ max_depth: It acts as a regularization parameter. The larger the depth the more complex the model thus less regularization and vice versa.

✦ min_samples_split: It acts as a regularization parameter. The larger the number the less complex the model thus more regularization and vice versa.

✦ min_samples_leaf: It acts as a regularization parameter. The larger the number the less complex the model thus more regularization and vice versa, but the validation error is behaving slightly strange.

- Hyperparameter Search

Optimal Configuration

min_samples_split	min_samples_leaf	max_depth	accuracy
3	1	12	0.97156

Insights

✦ It gives a sense of the hyperparameters that are important and what values they should take.

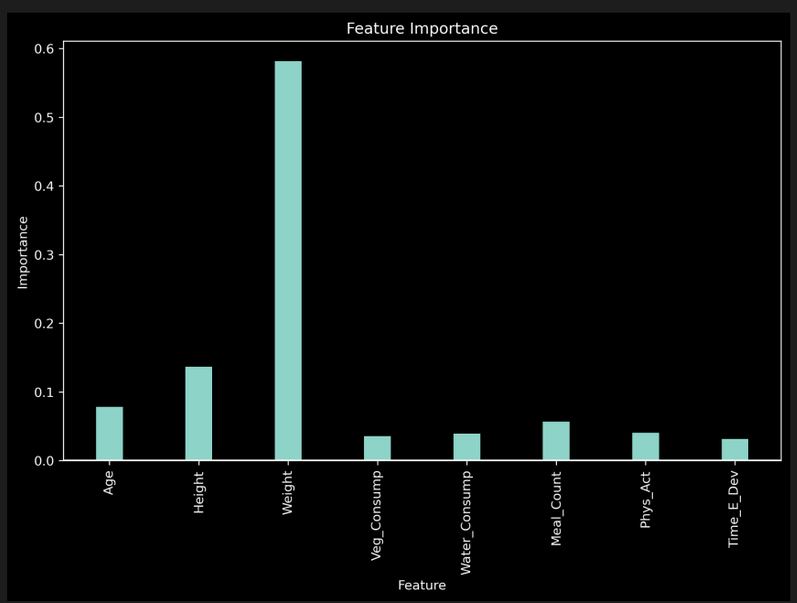
✦ Not should be taken for granted as the best as the validation is contaminated with any observations.

✦ The constant settings on which the optimization is ran are based on the validation curves.

- Hyperparameter Logging

Check RandomForest.ipynb

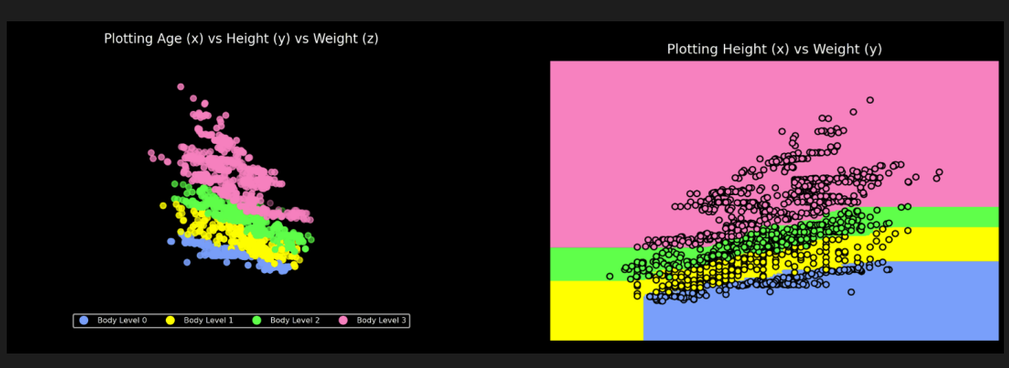
- Feature Analysis
  - Feature Importance



Insights

✦ Only weight, height and slightly the age seems to be most relevant to discriminate the classes.

- Visualization (Bonus)



Adaptive Boosting

Adaboost is one of the classifiers that we tried that didn't turn out to be so successful. In particular, it can be shown that Adaboost achieves zero training error after a constant multiple of $ln(N) = 8$. It appears that in our case that constant was too high which made the training time grow at an unfeasibly large rate.

info				read_data				AdaBoostClassifier				metrics	
time	date	duration	id	kind	select	standardize	split	n_estimators	random_state	learning_rate	algorithm	train_wf1	val_wf1
01:44:44	05/15/23	18.23 s	1	Numerical	True	True	all	1000	42	1.0	SAMME.R	0.639	0.5515
01:45:25	05/15/23	2.35 min	2	Numerical	True	True	all	10000	42	1.0	SAMME.R	0.639	0.5205
01:48:07	05/15/23	10.75 min	3	Numerical	True	True	all	100000	42	1.0	SAMME.R	0.639	0.5306
23:50:08	05/16/23	1.20 min	4	Numerical	True	False	all	10000	42	1.0	SAMME.R	0.639	0.5205

Model Evaluation

For evaluation, we originally partitioned the data into a training and a validation set. 10-Fold Cross-Validation Repeated 10 times was used to make decisions for each model. After the best hyperparameter setting was found for each model we released the validation set, and 10-Fold Cross-Validation Repeated 10 times was used to compare between models.

Below are shown the final results, Train WF1 suggests evaluating on the full training data and Val WF1 is after performing the 10-Fold Cross-Validation repeated 10 times.

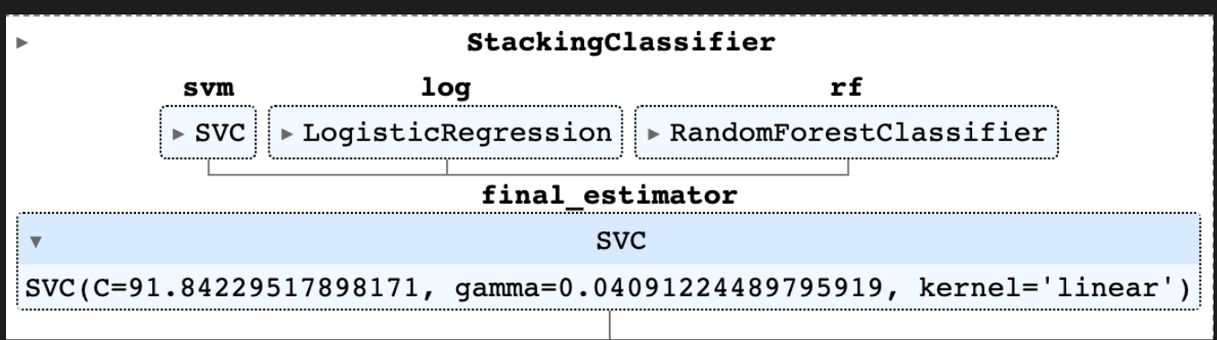
Model	Train WF1	Val WF1
Perceptron	0.82	0.79
SVM	0.993	0.9865
Logistic Regression	0.986	0.9838
Random Forest	1.0	0.9763
Adaptive Boosting	0.639	0.5515

Comparison and Ensemble

It's obvious that SVM, Logistic Regression and Random Forest are all closely matching in terms of their high performance. Instead of choosing between them we decided to choose them all by forming an Ensemble.

We started by trying voting, including all of SVM, Logistic Regression, Random Forest and initial attempts have shown accuracy 96.5% which was lower than the best model.

Next, we considered stacking over the same set of models. This time it has led to the best performance in terms of the WF1 metric.



Model	Train WF1	Val WF1
Stacking Ensemble	0.999	0.991

Model Delivery & Conclusions

The model we deployed is the final stacked ensemble model shown above. The main conclusions are that weight and height seem to be very indicative of the true target where the underlying relation seems to be linear.

Contributions

Task	Person
Data Analysis	Essam Wisam
Logistic Regression Initiation	Mariem Muhammed
Logistic Regression Analysis	Essam Wisam
SVM	Marim Naser
Gaussian Naive Bayes	Mohamed Saad
Perceptron Initiation	Essam Wisam
Perceptron Analysis	Mariem Muhammed
Hoeffding Test Set Size	Essam Wisam
Validation Curves, Regularization Effects & Overfitting Analysis	Mariem Muhammed
10dvc Check	Essam Wisam
Learning Curves	Mariem Muhammed
Ensemble Model (Bagging, Stacking)	Mohamed Saad
Adaboost (Gradient Boost)	Mohamed Saad
Cross Validation	Mariem Muhammed
Analyzing Support Vectors	Marim Naser
Weights Analysis	Mariem Muhammed
Handle Class Imbalance with Costs & Oversampling	Marim Naser
Stacking Ensemble	Mohamed Saad
Bias Variance Analysis	Essam Wisam
Visualize Models	Essam Wisam

Task	Person
Data Analysis	Essam Wisam
Logistic Regression Initiation	Mariem Muhammed
Logistic Regression Analysis	Essam Wisam
SVM	Marim Naser
Gaussian Naive Bayes	Mohamed Saad
Perceptron Initiation	Essam Wisam
Perceptron Analysis	Mariem Muhammed
Hoeffding Test Set Size	Essam Wisam
Validation Curves, Regularization Effects & Overfitting Analysis	Mariem Muhammed
10dvc Check	Essam Wisam