

# Contextual bandits on multi-class prediction of obesity risk

## Summary

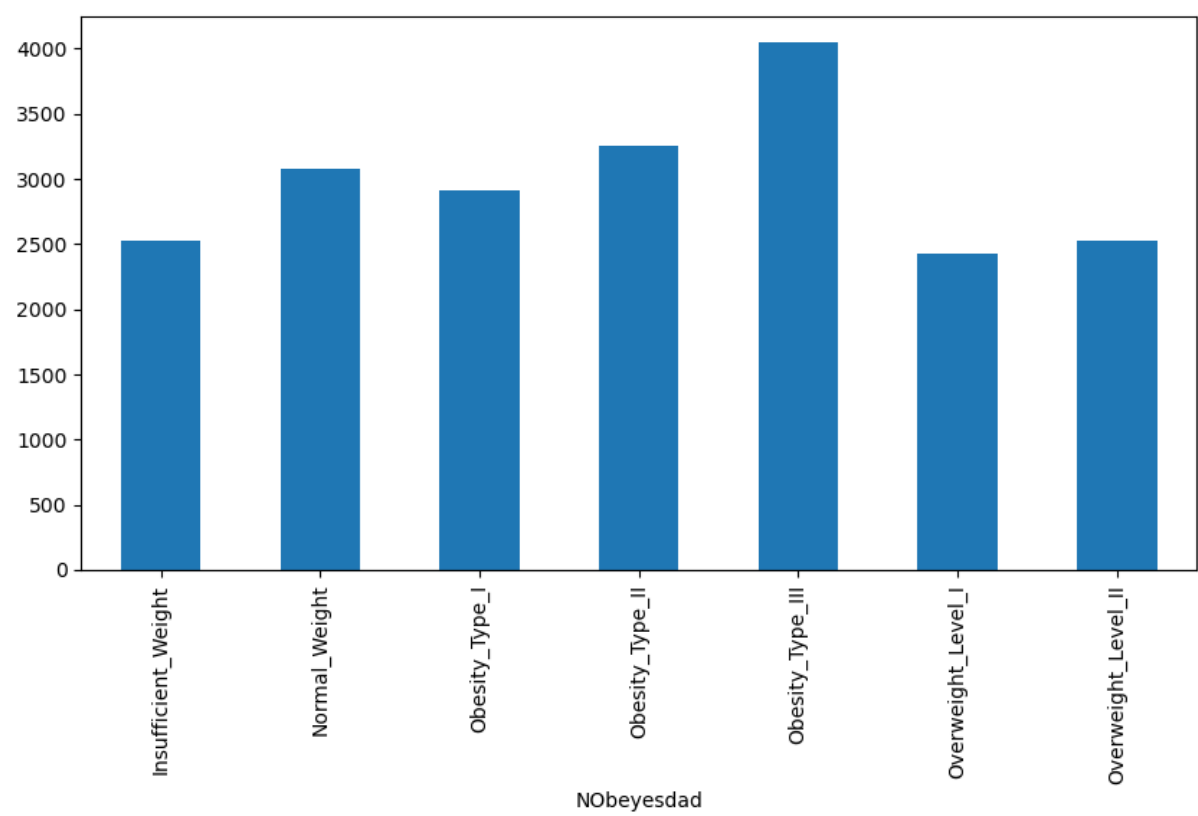
Contextual bandits model applied on obesity risk dataset. The original dataset was transformed via preprocessing and feature engineering according to an EDA. A simulator was then created from the dataset to mimic an online learning setting. Lastly, the contextual bandits model was run on the simulator. The final average regret was 0.267.

## Data

The original problem of the dataset was to predict a person's **NObeyesdad** (a measure similar to BMI) based on a set of 17 related features. **NObeyesdad** has 7 classes, ranging from **insufficient\_weight** to **overweight\_level\_II** similar to BMI classifications. Some of the given features are simpler like gender, weight, age. Others are more complicated. For example, **FCVC** is frequency of consumption of vegetables, **NCP** is number of main meals, and **FAF** is physical activity frequency.

## EDA

Distributions of the data were plotted. We found that the dataset with respect to **NObeyesdad** is roughly balanced.



## Preprocessing

Our EDA showed us that there are no N.A. values to deal with, but there are features with string values that need to be handled. Accordingly, `NObeyesdad` labels and string features were converted to numerical features with `sklearn.preprocessing.LabelEncoder` and `sklearn.preprocessing.OrdinalEncoder` respectively.

## Feature Engineering

Some features were constructed as combinations of the original ones.

```
X["BMI"] = X["Weight"]/X["Height"]**2>
X["BMI_group"] = group_series(X["BMI"], [18.5, 25, 30, 35, 40])
X["FAVC-FCVC"] = X["FAVC"] - X["FCVC"]
X["BMI*FAF"] = X["BMI"] * X["FAF"]
X["FAF-TUE"] = X["FAF"] - X["TUE"]
X["FCVC*NCP"] = X["FCVC"] * X["NCP"]
X["BMI/NCP"] = X["BMI"]/X["NCP"]
X["Age_group"] = group_series(X["Age"], [10, 20, 30, 40, 50, 60, 70])
```

Notably, `BMI_group` is closely related to `NObeyesdad` which helped decrease average regret by a significant amount. Without BMI statistics but including other engineered features, the average regret was 0.339.

After feature engineering, correlations of features were then calculated. The only pair of features with > 0.95 correlation was `BMI` and `BMI_group` (0.983). Based on this, `BMI` feature was removed.

## Scaling

Scaling was not used due to potential data leakage. It might be worth noting that scaling significantly decreases regret. Using `sklearn.preprocessing.RobustScaler`, the final average regret was 0.188.

## Model

The contextual bandits model was created with `Vowpal Wabbit --cb_explore`. A useful addition was to set the model to `--first 100`, which tells the model to explore (select each action with uniform probability) for the first 100 steps. Afterwards, the model will exploit for the rest of the steps. This helped prevent premature optimization that greatly increased regret.

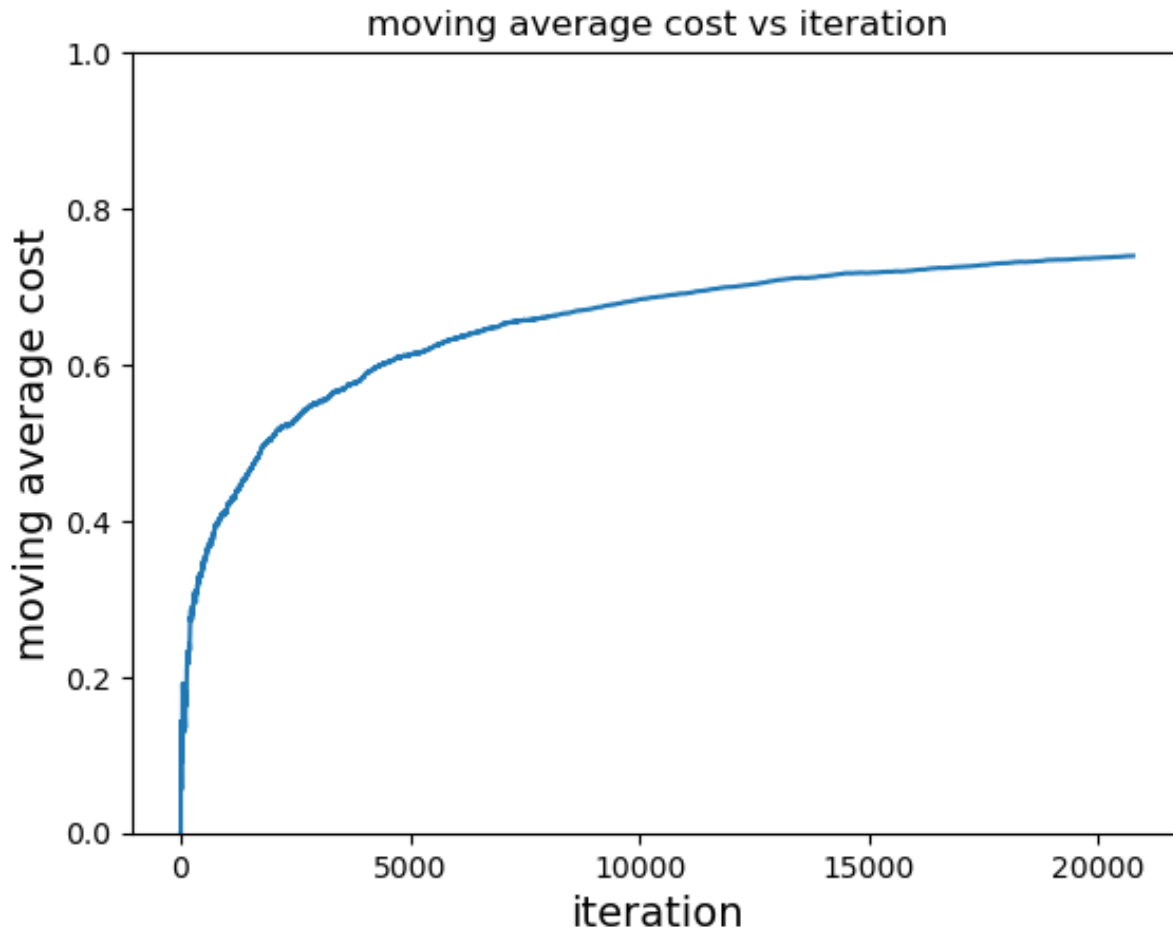
## Simulator

It was difficult for us to find a contextual bandits environment readily available, so we chose to convert a classification task to one instead. Each `NObeyesdad` class becomes an arm (7 total arms). The simulator converts the features of each datapoint into a context. The `NObeyesdad` class associated with the features becomes the correct arm to pull.

Our simulator performs a loop through all the datapoints in the dataset. For each datapoint, it gives the features of the datapoint as the context to the contextual bandits model. The contextual bandits model in turn gives a probability distribution of actions. An action is sampled from the distribution. The cost of the action is then calculated, with -1 signifying that the action was the correct arm (correct `NObeyesdad` class) and 0 otherwise. Finally, the model learns from the result.

## Result

The model was run on all of the training data. Costs were recorded at each step of the simulator and the negative of the moving average of the costs was calculated (plotted below).



The final average regret was calculated as  $1 - (-\text{average\_cost}) = 0.267$ . The final accuracy of the model was 0.864.

## Flaws and Future Improvements

The dataset we used is not a typical use case of contextual bandits. It will be good to explore performance on something like a recommender dataset. These types of datasets also have reward stochasticity, which this dataset/simulator does not have.

Another area to look at is environments with larger features spaces and bigger action spaces. Exploration mode `first` performed the best on the simulator of the few tested, even though `Vowpal Wabbit` has much more complicated approaches. `Open Cover` is the most complex model in the library (stated by `Vowpal Wabbit`). It learns `n` policies and achieved 0.331 regret with 1. This could be due to the simplicity of the dataset, which favored models that are quicker to exploit like `first`. It is also interesting to note that as we increased the complexity of `Open Cover` by increasing the number of covers, model performance degraded. This reflects the results of the original authors [1]. It will be interesting to test when more complex models, especially `Open Cover` with higher number of policies, works well.

For large action space (LAS) environments, `Vowpal Wabbit` provides a LAS algorithm which eliminates similar actions to allow for better exploration. This is a possible future direction to explore as well.

## References

[1] Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L. & Schapire, R.. (2014). Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. *Proceedings of the 31st International Conference on Machine Learning*, in *Proceedings of Machine Learning Research* 32(2):1638-1646 Available from <https://proceedings.mlr.press/v32/agarwalb14.html>.