Wells Fargo Campus Analytics Challenge: Live Green and Live Happy

Karl Roush, karlroush@gatech.edu
Georgia Institute of Technology

Problem Definition & Summary

The problem presented in this challenge was "to create a machine learning algorithm that minimizes carbon footprint for each customer while maintaining their total quality of life". The target is to reduce each individual's CFP, while also maintaining QoLI. This document fulfils *deliverables one and two*. The code I wrote can be viewed at [https://github.com/karoush], once GitHub finishes maintenance, or in the attached files.

As a summary, the data was split into two files (raw data and activity-specific CFP weighting). This data was extracted, **an index was created**, and a general model based on human intuition was formed. The data was then run through several machine learning algorithms, which confirmed and improved the model. These process are addressed generally below, and more specifically in the following sections. A **mathematical formulation of the problem** can be found in **Analysis.**

To create a reference to compare the model against, I generated my own model from the data. This model adjusts the duration of each activity in proportion with the QoLI. As an example, if an individual valued "AC" at 50 and had a duration of 4 hours, this would be adjusted to a duration of 2 hours. Essentially, the model brings the duration of activities in line with their importance-reducing CFP in areas that do not heavily affect total quality of life.

A number of machine learning models were tested*, but a Support Vector Machine algorithm from the Pandas python library was chosen since it had the highest accuracy. Post-testing analysis indicated that accuracy was 98.55%, and that the ML model often underestimated the minimum CFP for an individual by 0.0145.

Recommendations for the future would be a more custom algorithm, tailored exclusively for this task. A model generated by a neural network would work best, but at the moment I do not have sufficient computing power available to warrant the increased accuracy over SVM. Beyond this, additional training data is always recommended since it allows the model to be more accurate.

The data product generated (model of how to minimize CFP, while preserving QoLI) is **a good example of machine learning**. A human can easily see a trend in the data and produce a model based on their intuition, but this model cannot be verified due to the large volume of data. By utilizing different ML methods, this model could be verified and improved. In other words, this data product is a good example of ML since it not only verifies a human generated model, but also improves it, allowing it to be generalized to large data sets and new information.

Note: all equations, graphs, and visualizations are presented in a larger size at the end of the document.

***:** Logistic Regression, LinearDiscriminantAnalysis, KNeighborsClassifier, DecisionTreeClassifier, GaussianNB, SVM
CFP: Carbon footprint
ML: machine learning
QoLI: quality of life importance

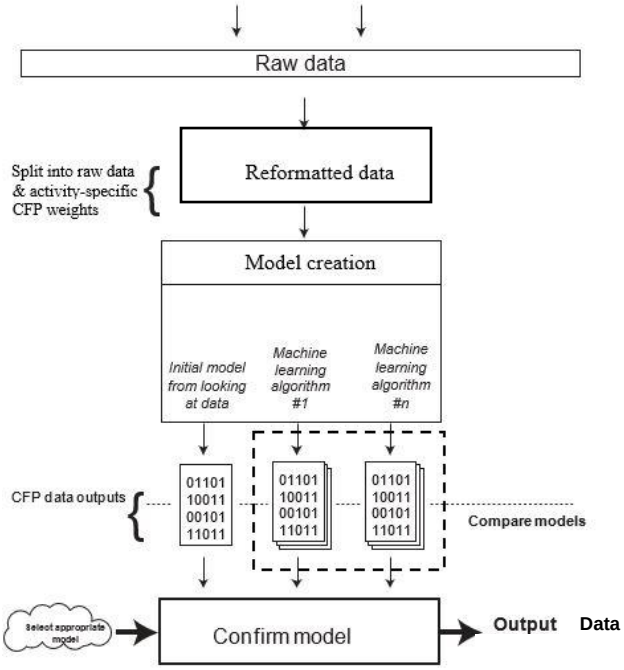


Figure 1: *Diagram of work processes*

1 Initial Data Reformatting & Extraction

The data was provided in the form of a .XLSX, Microsoft Excel spreadsheet, with two worksheets. I split the file into two .csv files (one for the data, one for the weights), to make the data easier to work with in the Python 3.6 environment. This has the added benefit of allowing the data to be easily modified or swapped out.

Missing values were also present in the "consumption" and "quality of lift importance" (QoLI) columns. This was handled through an application of an **index**.

When reading the data from the file, I created an **index** of the pollution data by activity (i.e. weighting and each activity, by individual). This also had the consumption and QoLI values when they were not zero. From these values, I was able to calculate a median value and fill in the missing entries with it. A median was chosen since an average can be easily skewed by outliers. However, I implemented an option to fill these missing values with a "0", but the default is to replace it with the median value.

This index, since it contains critical values in a readily accessible format, also allowed for pre-testing data analysis, discussed in the **Insights** section. My **rationale for including these values** was to allow for analysis & reference in a workable format.

It is also worth noting the misspelling in a couple of the "activities". This caused conflict when creating the index since the two fields would conflict. I defaulted to using the fields provided on the weights tab since that is what governs the raw data set.

2 Feature Engineering

With the data split into two .CSV files and extracted, it was time to move onto creating the models. However, not all the features in the data were critical for analysis. **The following features were excluded:**

- Group
- Units of duration

The group feature was a number liked linked certain activities. For example, *"Household heating => 70F, …, Use of air conditioner"* fall under one group (specifically #1) since they all refer to heating/AC. This grouping was not used since addressing each activity (instead as a group) would allow for a more targeted reduction in carbon footprint.

	Indnum	Group	Activity	Units	Consumption	QoLI
2	1	1	Household heating => 70F	hours	2	88
3	1	1	Household heating < 70F	hours	10	85
4	1	1	Use of heat pump	hours		50
5	1	1	Use of air conditioner	hours	20	45
6	1	2	shower - short	count	5	98

Figure 2: *Example of grouping from provided raw data*

Units of duration provided a context to the reader for each activity. It was not used since the model does not consider units- only the relationship between an activity and its CFP.

Missing values were replaced with the appropriate median from the created pollution index (see section 1). There is an option to replace the missing values with "0", but the median is a more accurate replacement since it makes less assumptions about the data.

Some features were transformed, most critically of which was the QoLI. This was given as a number with no context. I assumed it to be a relative percentage (as that is the most logical conclusion), but I have no way of checking this assumption. As such, this feature was divided by 100 to convert it to a numerically appropriate value.

Additionally, several of the features were abstracted. An **index** of the activity names was created, with each activity being given a numeric value. This allows the program to reference each activity by a number instead of a long string and allows for easier modifications in the future.

```
93 def activity2num(activity):
94     return {
95         'Household heating => 70F': 1,
96         'Household heating < 70F': 2,
97         'Use of heat pump':3,
98         #...
99         'hazardous or electric items disposed':26,
100        'large items disposed':27,
101    }[activity]
```

Figure 3: *Shortened index for referencing activities*

3 Analysis

The general form of finding CFP is simply taking the duration of an activity and multiplying it by a specific CFP/duration. These are summed across activities to find an individual's CFP, seen **mathematically** below:

$$General\ CFP = \sum_{activities} duration_{importance} * specific\ CFP \qquad (1)$$

As described in the **Problem Definition & Summary**, the goal is to minimize this equation. This can be done through several ways:

- Minimize the specific CFP
- Minimize duration, constrained by QoLI

The first option is entirely feasible but would require drastic changes in the individual's lifestyle. If an individual had natural gas for home heating, they could swap to solar and make their CFP zero; though with a lot of work. The second option is the better approach but *must be constrained by QoLI*. Otherwise, you could just reduce the duration to zero and eliminate CFP for that activity.

With these conditions in mind, an initial model was generated from the data (see **Problem Definition & Summary**). This model adjusts the duration of each activity in proportion with the QoLI. As an example, if an individual valued "AC" at 50 and had a duration of 4 hours, this would be adjusted to a duration of 2 hours. Essentially, the model brings the duration of activities in line with their duration- reducing CFP in areas that do not heavily affect total quality of life. **Mathematically**, this is represented below:

$$Total\ CFP = \sum_{activities} time * \frac{importance}{100} * specific\ CFP \qquad (2)$$

Note the difference to the function describing total CFP (1), generally. The term [duration * importance/100] adjusts the duration of an activity to be in line with the importance to the individual. From there, the activity CFP is just [~ * specific CFP] as it varies based on what the individual is using for that activity. An individual's CFP is simply the sum of their activities' CFPs.

Moving to the machine learning algorithms, **the following models were tested** (built into the Pandas python library):

- Logistic regression
- Linear discriminant analysis
- K-nearest neighbors
- Decision Tree
- Gaussian Naive Bayes
- Support vector machine

To determine which algorithm to use, each one was run on 10-fold cross validation to estimate accuracy. This works by splitting the dataset into 10 parts- nine training, one testing, and is repeated for all combinations of train-test splits.

The evaluation metrics used to compare models were accuracy and standard deviation. All algorithms were run using the parameters as *"kfold= model_selection.KFold (n_splits= 10, random_state =seed)".* Running the cross produced the following results:

Model	Accuracy	Standard Dev.
LR	0.966667	0.040825
LDA	0.975000	0.038188
KNN	0.983333	0.033333
DT	0.975000	0.038188
GNB	0.975000	0.053359
SVM	0.991667	0.025000

Table 1: *Accuracy and Standard Deviation Results from initial 10-fold cross, by algorithm*

The SVM algorithm had the highest accuracy, so it was selected for usage. Since the initial 10-fold cross only produces a rough model, fine tuning was needed. This was accomplished by adjusting the *n_split* parameter. Essentially, my program tested from n=10 to n=number of samples (1002) and kept the highest accuracy value. My **thought process behind this fine tuning** was that since *n_split* is the largest influencer in the model, adjusting it will have the greatest effect. Many of the smaller fine-tuning options were also adjusted, but given their mathematical complexity, it is best to focus solely on discussing *n_split*, whose final value came out to be ~995 (varies based on other parameters).

After training, the SVM model provided an evaluation metric of accuracy (max=0.933). The data produced by the SVM model was also compared to the original model output data. *The difference (ML-model) between the two output datasets was -1.45% (actual accuracy= 98.55%), providing an insight that the ML often underestimated the minimum CFP for an individual by about 0.0145.*

4 Insights

I would also like to highlight the median importance by activity. It appears that showering, heating, and transportation are the most valued. This is to be expected since they play a crucial role in everyday life. However, high CFP impact of "air travel- small plane" and "self-clean" have relatively low importance, making them prime targets for reducing an individual's total CFP.

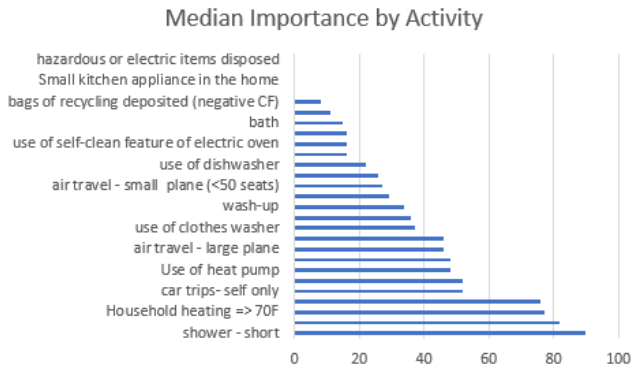


Figure 4: *Median importance for different activities*

The other insight worth pointing out is the drastic percentage change between initial CFP and minimized CFP. This can be seen in the graph below. The average reduction of CFP was 26.16%, which certainly is a non-trivial amount when considered across 1002 individuals.

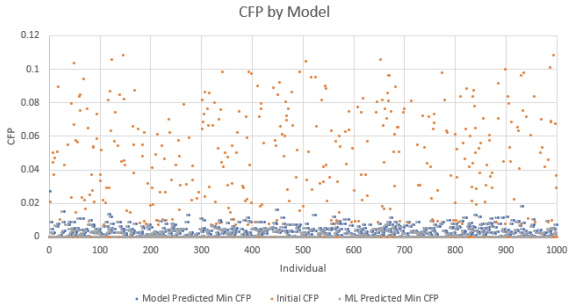


Figure 5: *CFP by model (orange= initial, blue= model, grey= ML)*

5 User Interface and Application Visualization (Deliverable 2)

Given how the model takes in several parameters and returns a value, a model like on online converter would work best.

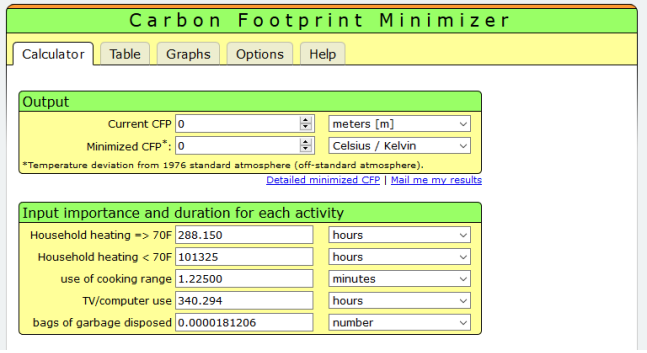


Figure 6: *Example of web application (modified for CFP)*

Alternatively, short "Buzzfeed" or "Facebook" style quizzes are very popular, so the calculator could be presented in that fashion. Each question in this quiz would ask about an activity and its corresponding info. Whatever the implementation, it is important to add a comparison, so people know if their CFP is "bad".

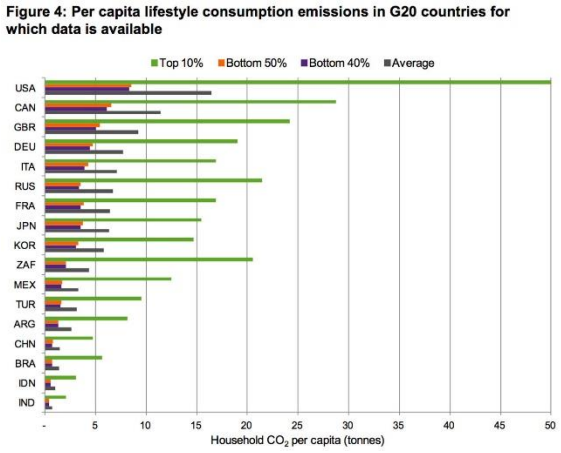


Figure 7: *Example of CFP comparison chart*

Equations, Graphs, Visualizations

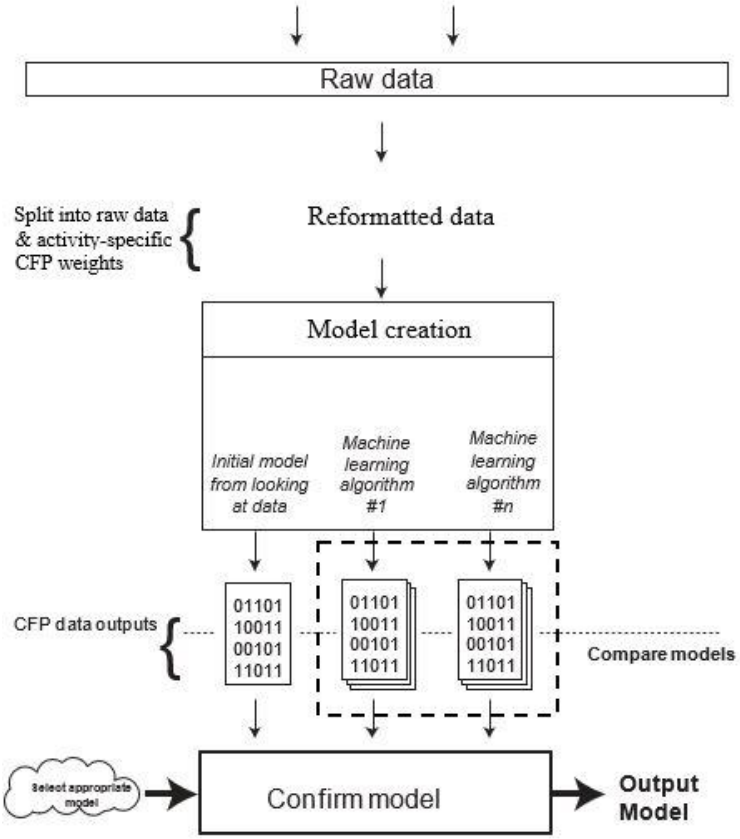
[1] Equation for an individual's CFP, as defined in the problem statement

$$General\ CFP = \sum_{activities} duration_{importance} * specific\ CFP$$

[2] Model generated equation for an individual's CFP

$$Total\ CFP = \sum_{activities} dur * \frac{importance}{100} * specific\ CFP$$

[3] ***Figure 1:*** *Diagram of work processes*



[4] ***Figure 2:*** *Example of grouping from provided raw data*

	Indnum	Group	Activity	Units	Consumption	QoLI
2	1	1	Household heating => 70F	hours	2	88
3	1	1	Household heating < 70F	hours	10	85
4	1	1	Use of heat pump	hours		50
5	1	1	Use of air conditioner	hours	20	45
6	1	2	shower - short	count	5	98

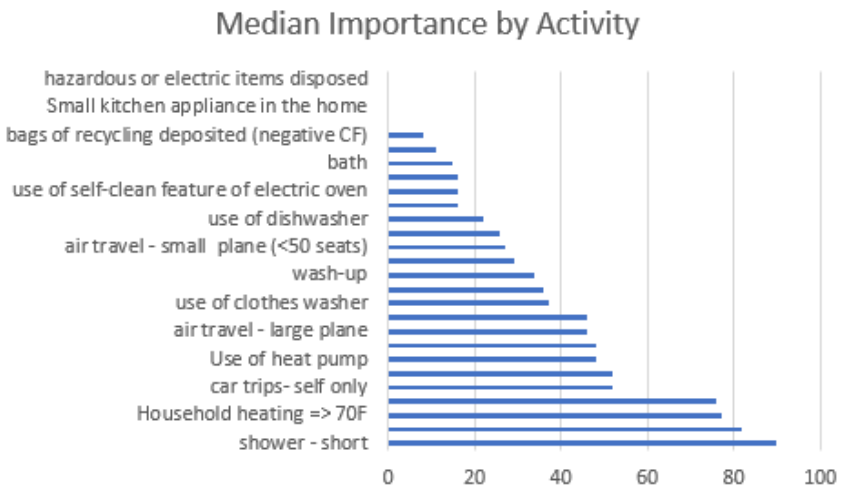
[5] ***Figure 3:*** *Shortened index for referencing activities*

```
def activity2num(activity):
    return {
        'Household heating => 70F': 1,
        'Household heating < 70F': 2,
        'Use of heat pump':3,
        #...
        'hazardous or electric items disposed':26,
        'large items disposed':27,
    }[activity]
```

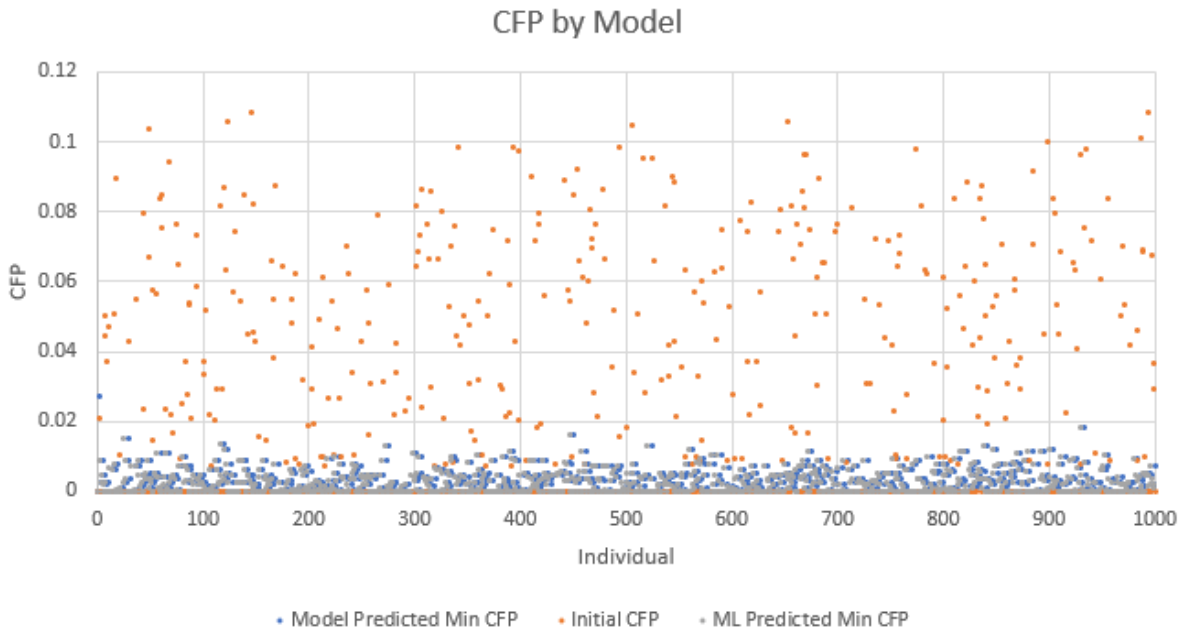
[6] ***Table 1:*** *Accuracy and Standard Deviation Results from initial 10-fold cross, by algorithm*

Model	Accuracy	Standard Dev.
LR	0.966667	0.040825
LDA	0.975000	0.038188
KNN	0.983333	0.033333
DT	0.975000	0.038188
GNB	0.975000	0.053359
SVM	0.991667	0.025000

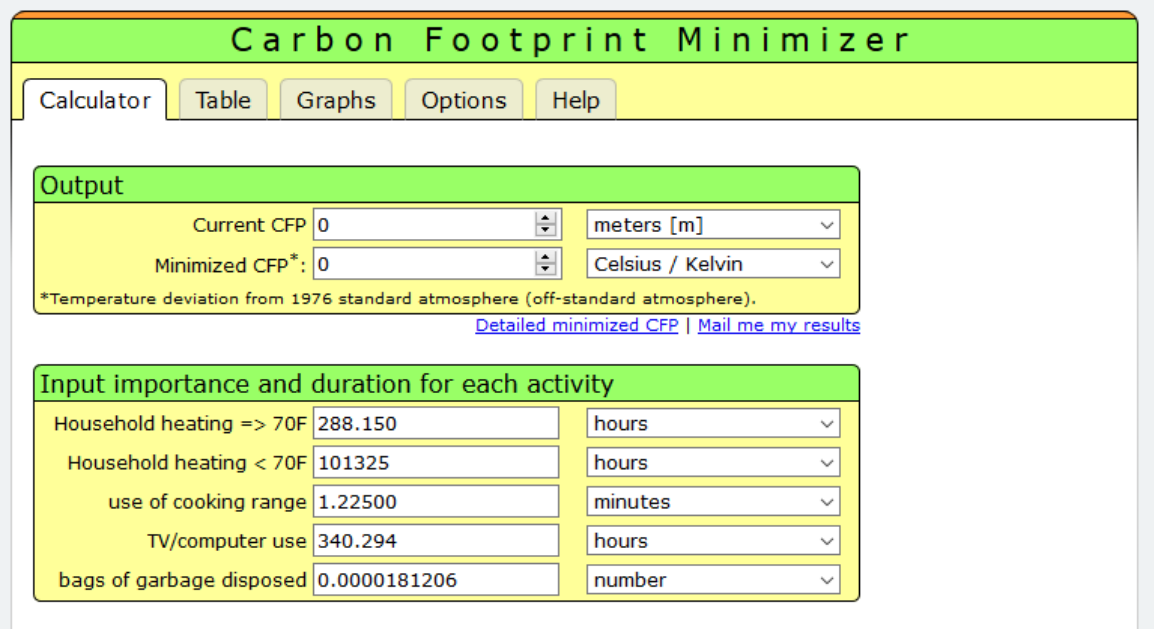
[7] ***Figure 4:*** *Median importance for different activities*



[8] ***Figure 5:*** *CFP by model (orange= initial, blue= model, grey= ML)*



[9] ***Figure 6:*** *Example of web application (modified for CFP)*



[10]***Figure 7:*** *Example of CFP comparison chart*