Optimatizing well being at work

BERRADI Aymane

Introduction:

1. Background :

Well-being at work is characterized by the way employees express their satisfaction with respect to their thermal environment, air quality of their workplace or environmental noise during work days. This subjective perception of the environmental conditions, such as feeling too warm or too cold for instance, has a tremendous impact on the health, the productivity and the well-being at work of each individual. Designing a data driven algorithm which manages to predict individuals' comfort with respect to their workplace environment is pivotal to providing adapted tunings of building management systems (such as the set points of heating, ventilating, and air conditioning systems) so as to ensure and monitor the best comfort in the building.

2.Data:

The data set contains several columns regarding work environment and confort classes.

1. ID: integer, uniquely identifies each observation,
2. String: defines date under format yyyy-mm-d hh:mm:ss,
3. Temperature: real number, temperature inside the room,
4. Humidity: real number, humidity of ambient air in the room,
5. Humex: real number, indicator of air quality in the room,
6. CO2: Integer, CO2 level in the room, in ppm (parts per million),
7. Bright: Integer, characterizes the brightness of the room,
8. Score: The classes are {1,2,3,4,5}, 5 being the optimal comfort and 1 the worst.

	ID	Temperature	Humidity	Humex	CO2	Bright	Score
count	8000.00000	8000.00000	8000.000000	8000.000000	8000.000000	8000.000000	8000.000000
mean	3999.50000	22.94535	33.790750	22.668762	586.471000	41.596375	2.930125
std	2309.54541	1.62307	8.241068	2.578996	202.641522	76.855898	0.893780
min	0.00000	17.90000	16.000000	15.500000	361.000000	1.000000	1.000000
25%	1999.75000	22.10000	27.000000	21.100000	452.000000	1.000000	2.000000
50%	3999.50000	23.20000	33.000000	22.800000	493.000000	1.000000	3.000000
75%	5999.25000	24.10000	41.000000	24.300000	693.250000	58.000000	4.000000
max	7999.00000	28.30000	58.000000	32.100000	2168.000000	882.000000	5.000000

Figure 1: Data Description

It seems that there is no ouliers for all variables, a further study for them will be held, for the CO2 the maximun value is 2168 ppm which is considered as dangerous level.

Data Visualisation

Le'ts see the distribution of our target variable score.

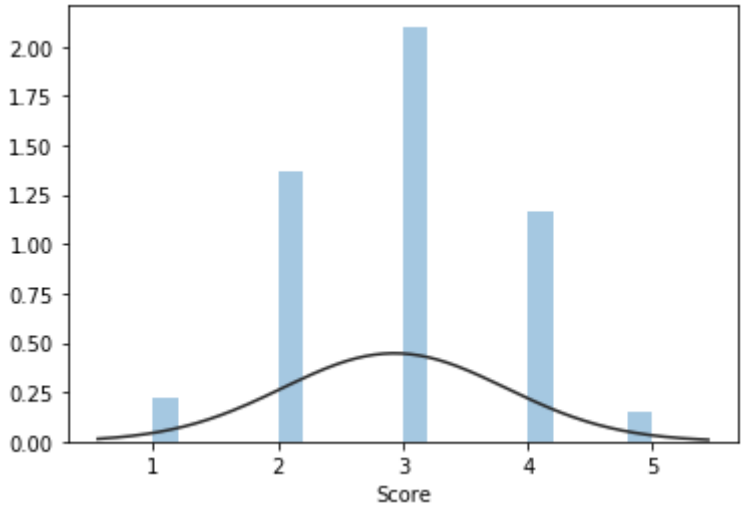
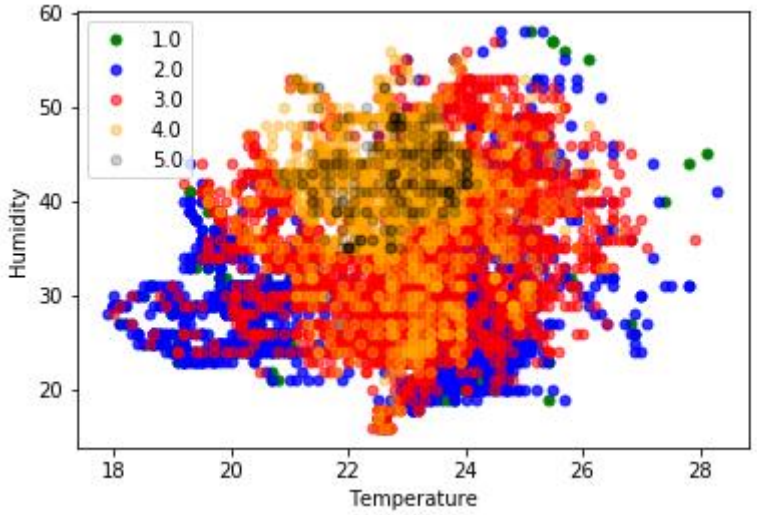


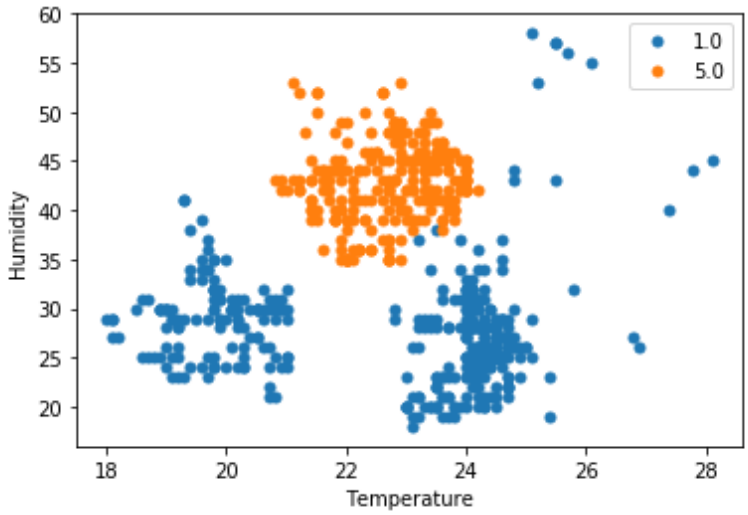
Figure 2:Distribution of the target variable

The Score variable has a **Normal Distribution**.

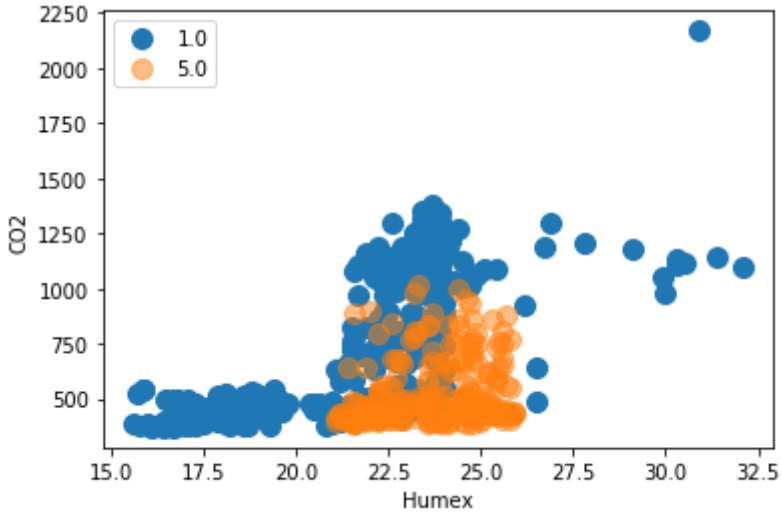
We tried first to plot just **Score** in function of **physical variables** (Temperature, Humidity).



We played with opacity of colors to have a maximum understanding of different distributions of Scores, but still a little bit confusing, so we decided to choose Score=(1,5) respectively the worst and best cases.



We can see that the best case (orange color) corresponds to a range of **(20.8 24.2)** of Temperature, and **(35 53)** of Humidity, we do the same thing to [Humex, CO2] that represent the quality of air.



We can see that the best case of Score occurs in the range of Humex **(21.1 25.9)** and CO2 **(382 1013)**.

Now we will look at "Score" in function of each variable.
We compute the average of **Scores** for each value of the column, and we apply to it the function round().

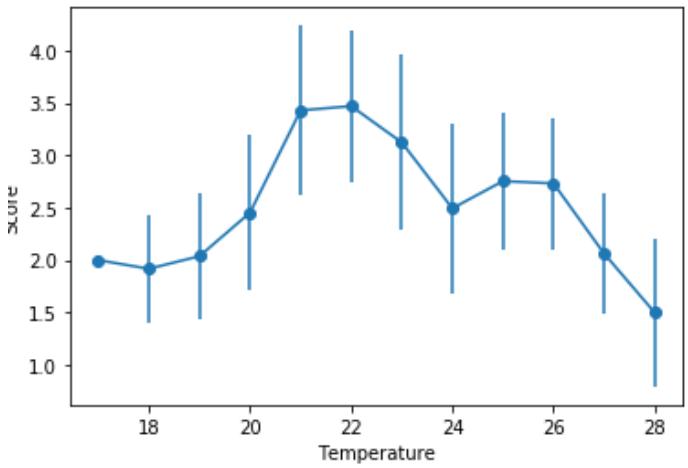


Figure 3: Example of the result for the temperature variable

From the previous graphs, we can pick the **optimum values** of each column, namely: **Temperature**, **Humidity** and **Humex** that leads to the best values of Score.

Models:

We will start by the basic model **decision tree** since we are dealing with a classification problem, the we will try other classifiers:

1. RandomForestClassifier
2. GradientBoostingClassifier
3. HistGradientBoostingClassifier
4. XGBClassifier
5. ExtraTreesClassifier

We note that the parameters of every model were tuned previously and the code will not be included in this notebook, we use the **cross validation** method to evaluate every model via **accuracy_score** metric.

```
cross(tree)
```

```
Accuracy of the DecisionTreeClassifier: 0.57 (+/- 0.02)
```

```
cross(forest)
```

```
Accuracy of the RandomForestClassifier: 0.68 (+/- 0.02)
```

```
cross(grad_boost)
```

```
Accuracy of the GradientBoostingClassifier: 0.73 (+/- 0.02)
```

```
cross(hist_boost)
```

```
Accuracy of the HistGradientBoostingClassifier: 0.74 (+/- 0.02)
```

```
cross(xgb_model)
```

```
Accuracy of the XGBClassifier: 0.73 (+/- 0.02)
```

```
cross(extra_tree)
```

```
Accuracy of the ExtraTreesClassifier: 0.72 (+/- 0.02)
```

Figure 4: Model Results

The results of the accuracy show that **HistGradientBoostingClassifier** performs better than other models, let's predict the target **Score** using each classifier, for this we need to split our data to train and test.

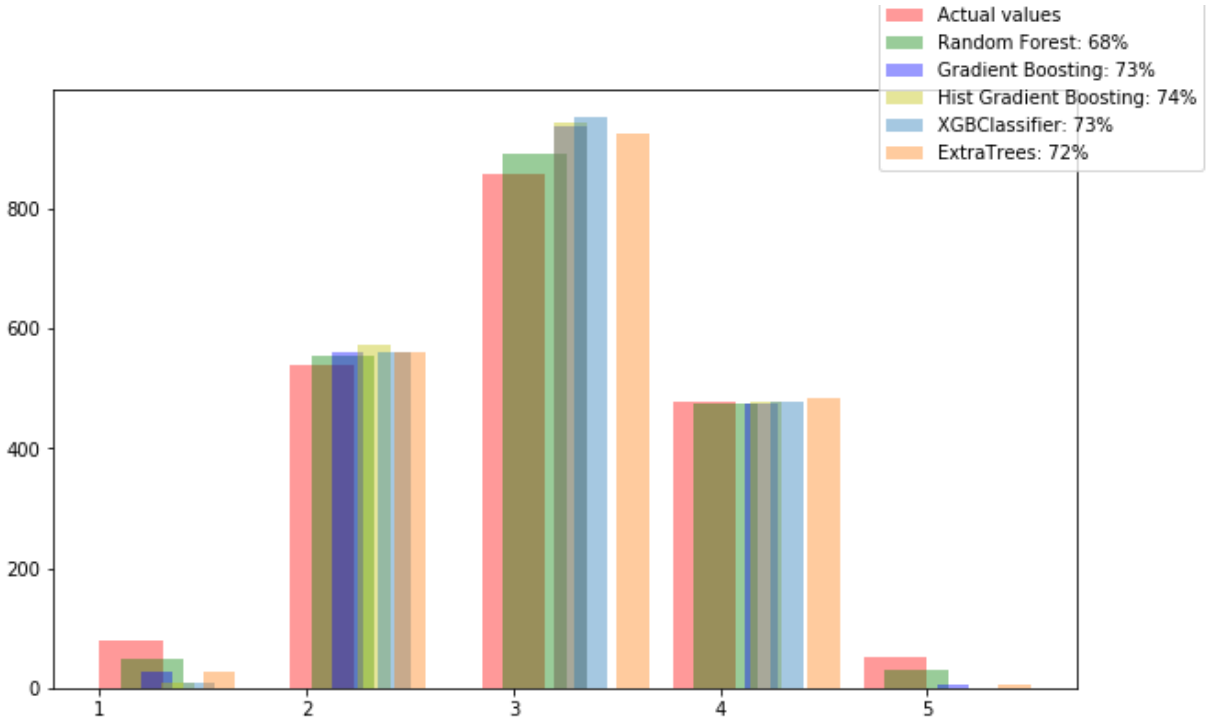
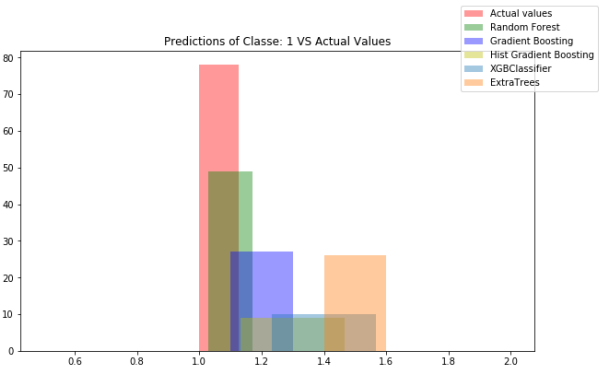
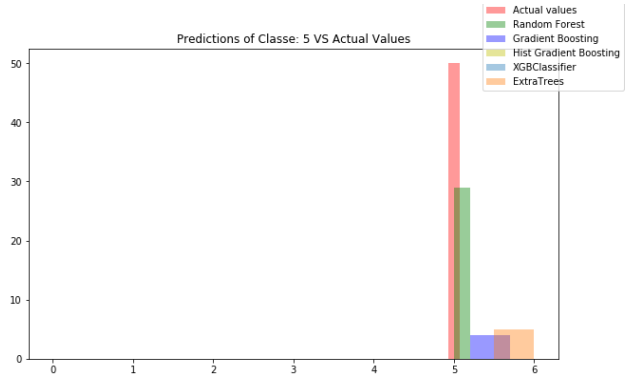


Figure 5: Prediction of the target variable using different models

The previous graph shows us differents histograms of models comparing to actual values of the Score variable, we can observe that all classifiers **don't perfom well on predicting the classes 1 and 5**.





By looking at the two previous graphs, we conclude obviously that all models perform very bad at the prediction of **the Score values 1 and 5,** perhaps it's due to **the low data dedicated for this two values.**

Conclusion:

As we can see, that all the models were skillful to predict the target variable, so we can set as future work an improvement of the model, either by stacking all models or using the voting method between all them.