Results

Optimization

Cost incurred

Data Cleaning

Introduction

Light Cost Optimization Game

Objective: Understand the current cost through light and movement sensors & its optimization to ensure the lowest electric cost & maintain Lab's functionality & comfort

Kenia Cevallos, Abel Murrieta, Mireia Rivière, Chandana Udupa,
Internet of Things, BABD

Results

Optimization

Cost incurred

Data Cleaning

Introduction

•Historical data from previous month

• Forecasting weather data for next 5 days

Light

- 4 csv
- Time Stamp
- Zone: 1, 2, 3
- Value: Total light the LDR sensor detects

Movement

- 4csv
- Time Stamp
- Zone: 1, 2, 3
- N_passages: account as "1" every time the PIR sensor detects movement in any of the Zones

Weather

- Json file
- From 1st to the 5th of April 2019 by frames of 3 hours

+

- New csv file
- Weather historical data by day of the observations

1) Data Cleaning & Understanding

2) The best solution that minimize the total cost using past data

Only an optimized scheduling or will be necessary to buy the new LED lights or to automate the curtains?

3) The best schedule for next week? (from 1st April to 5th April)

Objective: Understand the current cost through light and movement sensors & its optimization to ensure the lowest electric cost & maintain Lab's functionality & comfort

•*Historical data from previous month*

• *Forecasting weather data for next 5 days*

Light

- 4 csv
- Time Stamp
- Zone: 1, 2, 3
- Value: Total light the LDR sensor detects

Movement

- 4csv
- Time Stamp
- Zone: 1, 2, 3
- N_passages: account as "1" every time the PIR sensor detects movement in any of the Zones

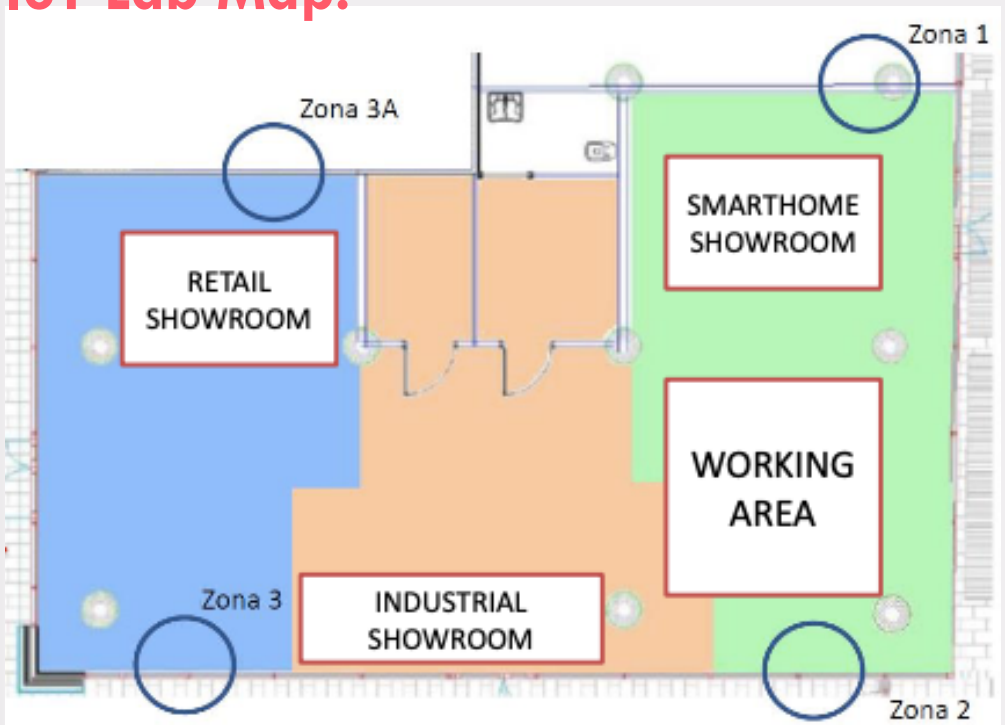
Weather

- Json file
- From 1st to the 5th of April 2019 by frames of 3 hours

+

- New csv file
- Weather historical data by day of the observations

IoT Lab Map:



Weekly schedule done on the weekend:

Lights:

•currently employed are 12 fixtures x 4 neon lights each

•each neon light consumes 30 W not dimmable

•can't make different schedules among the Lab zones.

Curtains:

•are manual

•can be set on one of three levels that filter respectively 0, 30 and 60% of outdoor light

•Curtains position are decided in the weekend, can't change its position after

Assumption 1

Lux 450 – Watt 30Wx12x4 consumption equivalent (as both are the max for the current capacity)

Assumption 2

Neon lights are either ALL on or ALL off

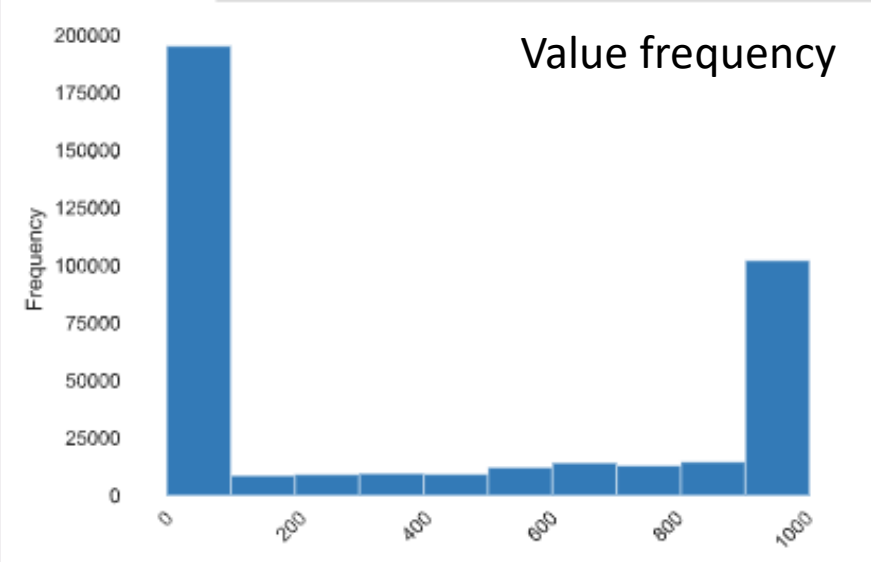
Light Dataset

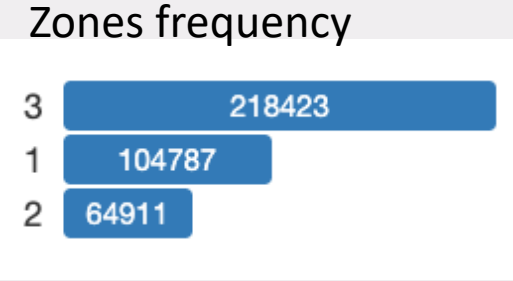
Dataset statistics

Number of variables	8
Number of observations	388121
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	6
Duplicate rows (%)	< 0.1%

Variable types

CAT	6
NUM	1
DATE	1



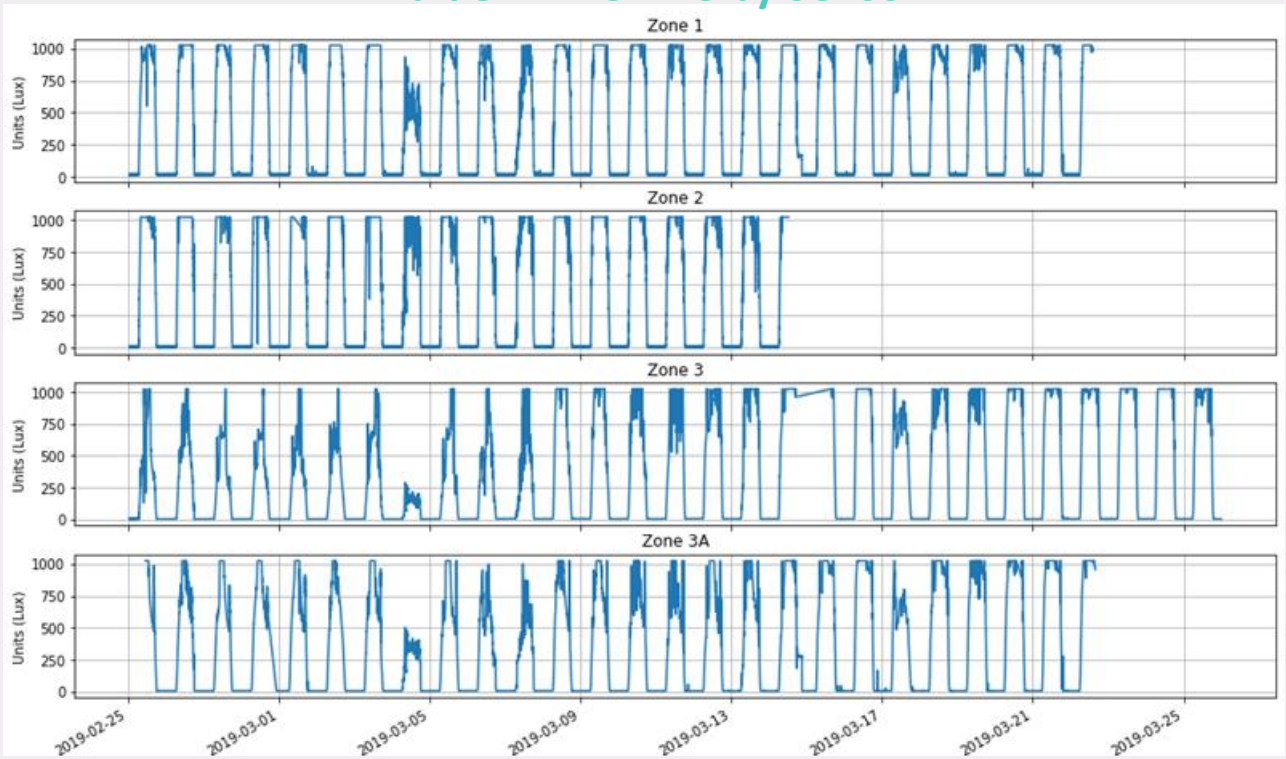


Info: The dataset has no missing values, 6 duplicate cells and no outliers. A total of 388121 observations that account for the lux value detected by 4 different LDR sensors. As it can be observed, the zone 3, which has two sensor, is the one that detects most of the observations, the zone 2 is the least. The mean value of "Value" column is 440.58, what implies that many current lux values represent lack of comfort for IoT Lab workers.

Extra info: The total value will be grouped first by a mean consumption by sensors, and then the light in each zone is averaged. This step is crucial for the analysis as the cost of the light is computed on an hourly consumption, therefore the mean is extracted, but each sensor understands the lights used in different areas and hence the means of the different sensors have to be taken into account separatelly.

Light Dataset

"Value" Time Line by Sensor



The time series by zone (zone 1 & 2 represent one light sensor, and 3 two sensors) shows clear discontinuities. Sensor data is highly susceptible to external changes and as well may have stopped working and later be replaced or repaired (what it seems like it happened during this month)

```
light14= light.loc['2019-03-01':'2019-03-14']
```

Assumption 3

Sensor mistakes: 2 weeks reference are representative enough to do an accurate optimization model for the whole month

Movement Dataset

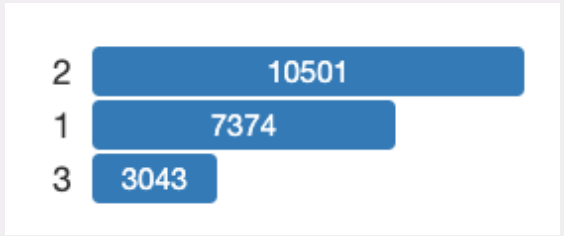
Dataset statistics

Number of variables	7
Number of observations	20918
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	3
Duplicate rows (%)	< 0.1%

Variable types

CAT	5
DATE	1
BOOL	1

Zones frequency



Info: The dataset has no missing values, 3 duplicate cells and no outliers. A total of 20918 observations that account for the sum every time the PIR sensor detects movement. As it can be observed, the zone 2, which is in between the two other areas, is the one that detects most of the observations. Instead, zone 3 (where the entrance is) has the least what reflects a pretty active working environment and is prove that this value can't be interpreted as a count. Zone 2 (in between the other zones) is the one that has higher frequency

Movement Dataset

date_parsed	day-of-week	n_passages
2019-03-01	Friday	1342
2019-03-03	Sunday	18
2019-03-04	Monday	2153
2019-03-05	Tuesday	2572
2019-03-06	Wednesday	1714
2019-03-07	Thursday	2254
2019-03-08	Friday	1092
2019-03-10	Sunday	29
2019-03-11	Monday	794
2019-03-12	Tuesday	2077
2019-03-13	Wednesday	1466
2019-03-14	Thursday	1583
2019-03-15	Friday	209
2019-03-16	Saturday	92
2019-03-18	Monday	389
2019-03-19	Tuesday	435
2019-03-20	Wednesday	522
2019-03-21	Thursday	1047
2019-03-22	Friday	133
2019-03-25	Monday	997

hour_of_timestamp	n_passages
7	372
8	1094
9	2581
10	2476
11	2408
12	2473
13	1898
14	2054
15	1835
16	1861
17	1128
18	463
19	144
20	131

Assumption 4

Based on the movement, we will take into consideration that the light detected after the working hours is due to external conditions, and therefore, has no impact in the electricity cost consumption of the Lab

Assumption 5

Based on the movement, the working hours are from 7 and to 19 pm

Days with no movement: 2019-03-02 & 2019-03-09 & 2019-03-17 & 2019-03-23 & 2019-03-24
Days with the lowest records of movement (outliers = 105): 2019-03-03 & 2019-03-10 & 2019-03-16

This two images summarize for the movement Dataset the sum of movement detection by day and by hour:

- By day: Some weekend days don't have recorded any movement, and the rest are below the 10% of the mean of all the days. The fact that there is movement detected but is very low (an outlier) shows how sensitive the sensor is and may incur to mistakes. The 10% of the mean of the total n° of passages by day is 104.59
- By hour: Another example on how sensitive are the sensors the movement starts at 7am until 20pm, instead the light sensor detects light during the 24 hours of the day. Moreover, only one day (14th of March) detects movement at 20 pm, therefore in the **prediction** for the next 5 days we will assume that lot Lab workers will not work after 19 pm.

Weather Datasets

Csv: *Historical weather summary by day*

Json: *weather summary by hour range of 3 hours for next 5 days forecasting*

Dataset statistics

Number of variables	7
Number of observations	36
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%

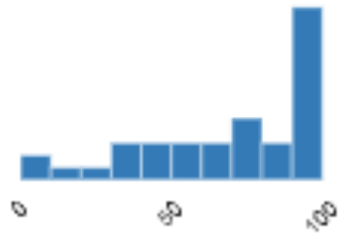
Variable types

NUM	5
CAT	2

	id	name	country	coord.lat	coord.lon
0	6542283	Milan	IT	45.4668	9.1905

clouds.all

Real number ($\mathbb{R}_{\geq 0}$)



Distinct count	14
Unique (%)	36.8%
Missing	0

Mean	67.6842105263
Minimum	0
Maximum	100
Zeros	1

```
weather = weather[["Date time","Cloud Cover","Conditions", "Temperature", "Wind Chill", "Wind Speed", "Precipitation"]]
```

Results

Optimization

Cost incurred

& features addition

Data Cleaning

introduction

Assumption 1
Lux 450 – Watt 30Wx12x4 consumption equivalent

Assumption 2
Neon lights: on or off

Assumption 3
2 weeks reference

Assumption 4 & 5
Cost of light in working hours: 7 to 20 (for forecasting until 19)

Assumption 6
Curtains position has been 0

FEATURES ADDITION:

hour_of_timestamp

Lux – Watt consumption equivalent

day-of-week

Understand if there is some pattern by day of the week

Date division in: day, month, year

Maintaining the date column

Weather

Include by day: Cloud Cover", "Conditions", "Temperature", "Wind Chill", "Wind Speed", "Precipitation"

Data frame "Value" Grouping

```
light14.groupby(["date", "id", "hour_of_timestamp"]).agg({"value":"mean", "hour_of_timestamp":'first', "id":'first'})
light14.groupby(["date", "hour_of_timestamp"]).agg({"value":"mean", "hour_of_timestamp":'first'})
```

```
nat_light_ =light_.loc[light_['date'].isin(values)]
```

values = ['2019-03-02','2019-03-03','2019-03-09','2019-03-10","2019-03-17","2019-03-23','2019-03-24'] = Weekend days

model = XGBRegressor()

```
y = X['value']
X.drop(['value', 'date', 'Conditions'], axis=1, inplace=True)
```

MAE Test is 10.98
RMSE Test is 16.81
r2: 0.99

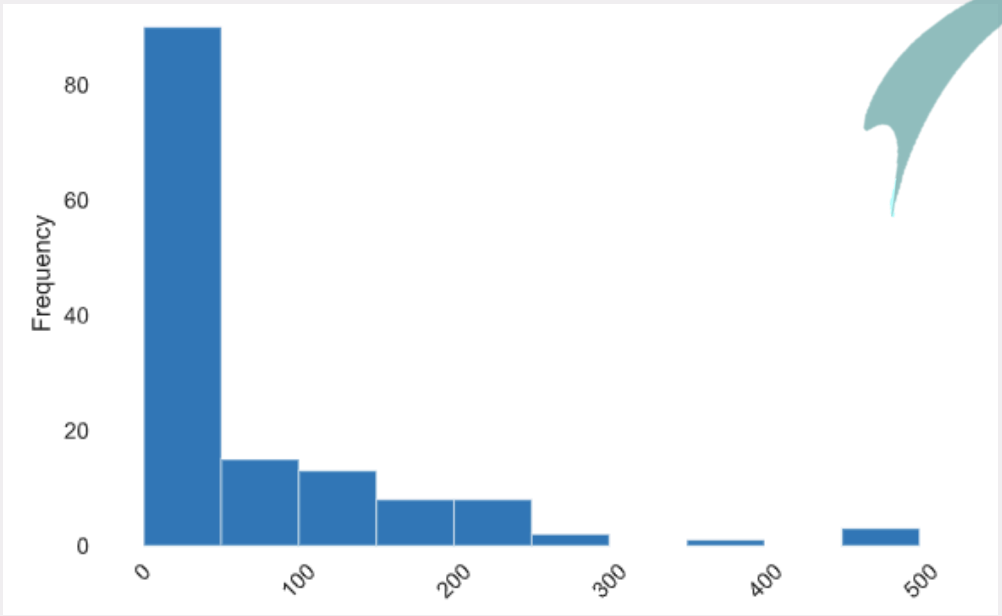
We have created a Data Frame with the weekend days grouped by hours, in order to create an algorithm that predicts the light that the LDR sensor detects when there is no one working and hence the lights are turned off.

This model is predicting according to weather conditions and hour of the day what is the natural light the LDR detects

Final Data frame creation:

Once the algorithm is applied to the 14 days of light observation, columns of artificial lux, natural light, andlack of comfort are computed. But a modification needs to be done as assumption 1 and 2 take a role: the artificial lux can take the value of either 0 or 450. The graph of frequency represents the artificial lux computed: the artificial light takes different values. Therefore, the algorithm helped to be a reference for the natural and artificial light, but if the artificial light is recorded below 200 we will assume the lights are off, and if it's above that are on. (the model is highly accurate, so if it detects artificial light above 200lux it is highly probable that the light was on. Transforming accordingly the values of artificial and natural light.
The final results of the frequency of the lights position are also below (90% of the time are off)

Artificial light values frequency



Assumption 6

Curtains position has been 0, as the values with or without movement are very similar

Artificial light after processing

Final Data frame creation:

Light14 columns relevant for the cost analysis are:

- Natural light: from the predictions when applying the XGBoost algorithm,
- Artificial light: either 450 or 0
- Total light: Natural light + Artificial Light (the light that the sensor detects)
- Lack of comfort lux: the difference of the total light with 450 if below, and with 650 if above. If the value is in between lack of comfort = 0
- Cost: E(kWh) = P(W) × t(hr) / 1000

Once the hourly light consumption has been understood, the cost can be computed…

```
Total cost = ((1440 * (light14[light14["value"] == 450].shape[1]))/1000)*0.30)
light14["lack_of_comfort_cost"]=(light14["lack_of_comfort"]*0.01)
```

The costs concurred by the electricity from the 1st of March of 2019 to the 14th is 4.32€

The costs concurred by lack of comfort from the 1st of March of 2019 to the 14th is 304.49€

The total costs concurred from the 1st of March of 2019 to the 14th is 308.81€

617.62 € monthly

02

New Led Lights

Extra costs: 4000€

Support three levels of dimming (0, 50 and 100% of luminosity)

01

Optimized Scheduling

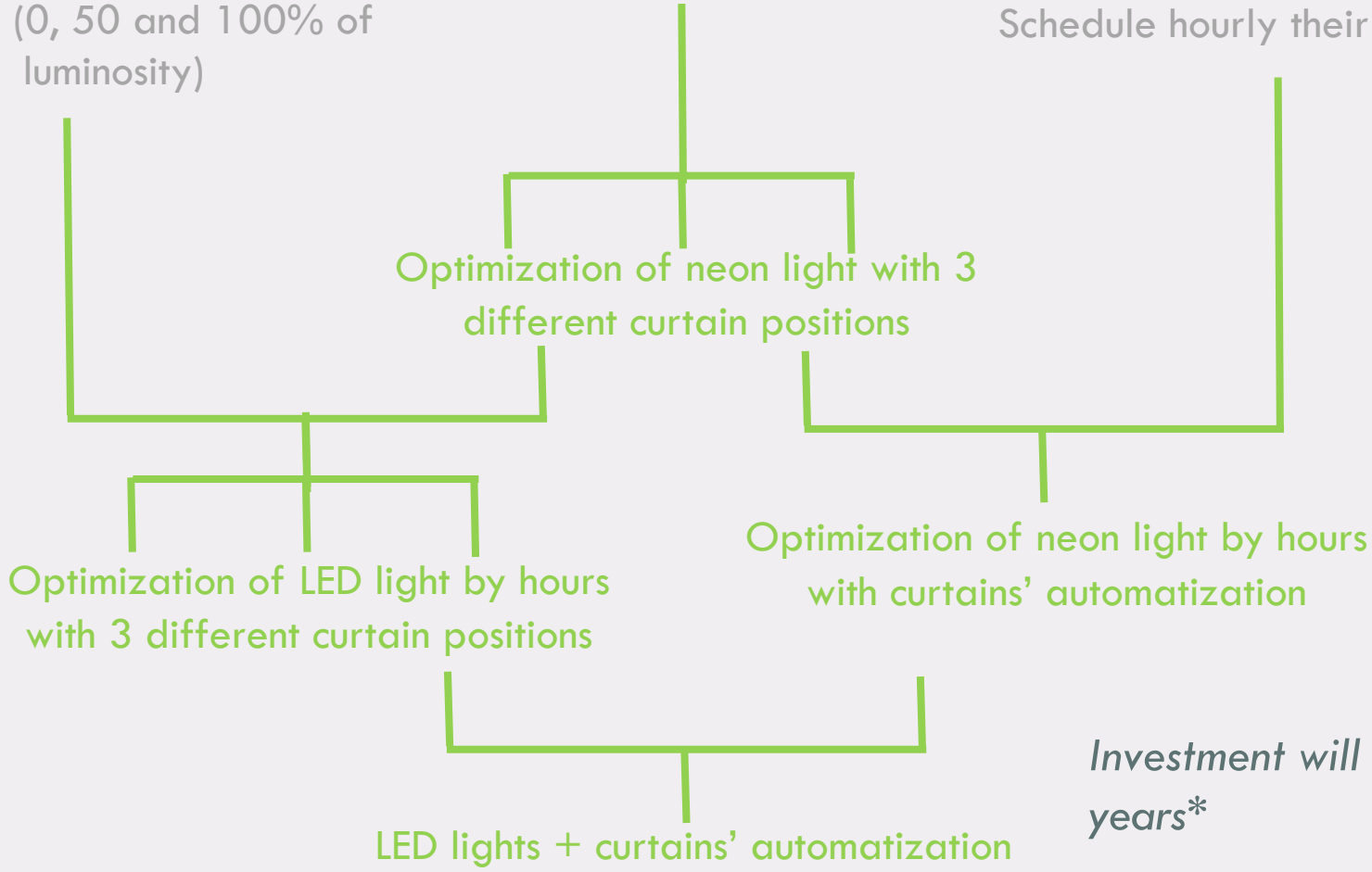
Extra costs: 0

03

Automate the curtains

Extra costs: 2000€

Schedule hourly their position



*Investment will be amortized in 10 years**

01

Optimized Scheduling

Extra costs: 0

Optimization of neon light with 3 different curtain positions

Curtains position: 0

Current consumption

Total costs:
- Electricity: 4.32€
- Lack of Comfort: 304.49€

TOTAL: 308.81€

Curtains position: 30%

["nat_light"] *(1-0.3)

Total costs:
- Electricity: 6.48€
- Lack of Comfort: 59.66€

TOTAL: 66.14€

Curtains position: 60%

["nat_light"] *(1-0.6)

Total costs:
- Electricity: 7.34€
- Lack of Comfort: 104.72€

TOTAL: 112.06€

Best performing: when curtain position = 30%

02

(+33.34 € monthly of led lights investment)

New Led Lights

Extra costs: 4000€

Support three levels of dimming (0, 50 and 100% of luminosity)

```
[light14.value_LED < 450, "artificial_light_LED50" ] = 225
[light14.value_LED] = light14["artificial_light_LED50"] + light14["nat_light"]

[light14.value_LED < 450, "artificial_light_LED50" ] = 450
[light14.value_LED] = light14["artificial_light_LED50"] + light14["nat_light"]
```

Best performing: when curtain position = 60%

Curtains position: 0

["nat_light"] *(1-0.0)

Total costs:
- Electricity: 4.80€
- Lack of Comfort: 153.91€

TOTAL: 158.71€
175.38€

Curtains position: 30%

["nat_light"] *(1-0.3)

Total costs:
- Electricity: 4.80€
- Lack of Comfort: 15.61€

TOTAL: 20.41€
37.07€

Curtains position: 60%

["nat_light"] *(1-0.6)

Total costs:
- Electricity: 4.80€
- Lack of Comfort: 3.24€

TOTAL: 8.04€
24.70€

```
#Curtain position 2
def artificial_lt_recompute1(row):
    if (row['value'] > 650):
        row["nat_light"] = row["nat_light"] *(1-0.3)
        row["value"] = row["artificial_light"] + row["nat_light"]
        return 450
    else:
        pass

#Curtain position 3
def artificial_lt_recompute2(row):
    if (row['value'] > 650):
        row["nat_light"] = row["nat_light"] *(1-0.6)
        row["value"] = row["artificial_light"] + row["nat_light"]
        return 450
    else:
        pass
```

(+16.67€ monthly of automated curtains investment)

03

Automate the curtains

Extra costs: 2000€

Schedule hourly their position

Curtains position: hourly

Total costs:

- **Electricity: 4.32€**
- **Lack of Comfort: 22.50€**

TOTAL: 26.82€ 35.15€

02

New Led Lights

Extra costs: 4000€

Support three levels of dimming (0, 50 and 100% of luminosity)

Curtains & light hourly

Total costs:

- **Electricity: 2.68€**
- **Lack of Comfort: 1.00€**

TOTAL: 3.64€
28.64€

03

Automate the curtains

Extra costs: 2000€

Schedule hourly their position

(+50 € monthly of led lights and automated curtains investment)

```
[light14.value_LED < 450, "artificial_light_LED50" ] = 225
[light14.value_LED] = light14["artificial_light_LED50"] + light14["nat_light"]

[light14.value_LED < 450, "artificial_light_LED50" ] = 450
[light14.value_LED] = light14["artificial_light_LED50"] + light14["nat_light"]
```

Best performing for each scenario and inclusion of the cost of investment (amortized in 10 years)

Optimization of neon light

LED lights + curtains' automatization

Curtains & light hourly

Saves: 556.02€ monthly

TOTAL: 3.64€
28.64€

Curtains position: 60%

Saves: 568.21€ monthly

TOTAL: 8.04€
24.70€

Curtains' automatization

Curtains position: hourly

Saves: 538.67€ monthly

TOTAL: 26.82€
35.15€

Standard optimization

Curtains position: 30%

Saves: 485.34€ monthly

TOTAL: 66.14€

5 days Forecast:

Using the json weather file….

After some preprocessing the columns of interest where selected: “hour_of_timestamp”, “Cloud Cover”, “Wind Speed”, “Precipitation”. Matching the columns for which the natural light algorithm was trained with.

```
predictions = model.predict(weath)
weath['nat_light'] = predictions
```

Optimization model application:

Led Lights with curtains at position 60%

Total costs:

- **Electricity: 2.40€**
- **Lack of Comfort: 0.0€**
- **Investment: 11.11€**

TOTAL: 13.51€ for the first week of April 2019

Future improvements:

schedule different light for different zones, as the quantity and timing of natural lux in each is different