2022 Final Data-Science Project

A Minor Change That Creates Meaningful Impact

Submitted by:

Dotan Beck - 313602641
Daniel Slotin - 204300925
Jonathan Schneider - 313173346
Amit Zeevi - 206171902

Presented to:

Dr. Gail Gilboa Freedman
Dr. Naveh Eskinazi

Table Of Content

Introduction 2

Business S.M.A.R.T 3

Data Exploration 6

Data Preparation 8

The Model 11

Results 16

Evaluation 17

Bibliography & Appendix 20

Introduction

This report demonstrates a new, insightful tool that we believe any business would love to get, and may act upon it immediately:

Finding the “QUICK WIN” - An Easy, fast and economical solution to implement. A change that will have “immediate”, visible impact.

After data preparation and normalization, we created a model using K-means to detect “risk groups” and then found the most significant changeable criteria within these groups using unique data frames, answering the question:

Which MINOR change in the company’s culture will achieve MEANINGFUL impact?

In the final section of the report, we suggest an even more thorough and accurate way of measuring worker’s habits in order to detect more precise vectors of improvement, AND to be able to authenticate improvement throughout periods of time.

Business S.M.A.R.T

PreSee's platform allows businesses to carefully evaluate their workers health and foresee potential risks. The data diagnostics allow a broad picture worker's health. Taking the CRISP approach, Our goal is to illuminate an area that taking a minor action in improving the company's lifestyle in that area will assure a real, significant gain. Whether a company wants to take care of worker's health to the long run and run continues health check-ups, or just wants to do a one-time diagnostic and act upon it, surely the first step towards this improvement could be the "QUICK WIN" criteria.

Furthermore, as creating meaningful health changes is surely difficult, PreSee's services would be more attractive for a business owner if they also provide a specific domain that is predicted to bring ACHIEVABLE IMPACT.

Specific: We want to find a SPECIFIC detail - what is the changeable habit criteria that a MINOR improvement in it will create an meaningful IMPACT?

More precisely, we are looking for the CHANGEABLE, unhealthy habit - smoking \ lack of exercise \ overweight \ work-stress \ blood-pressure - that is the most common within the company's risk groups. These groups are defined according to results of K-means division, and are supposed to be unique according to an UNCHANGEABLE factor such as family disease history, ECG abnormalities, etc..

Measurable: In order to measure our success, we want to make sure our focus is on a significant group of people, and not mere individuals. Therefore, we set the total size of the focused risk group\groups (sum of risk groups) a minimum bar of 10% of all workers. Then, within these groups, we will focus on the changeable habit with highest total rates.

Furthermore, in our last section of this report we suggest a more measurable way of collecting information, setting goals and measuring improvement. The full explanation will be in "EVALUATION" section.

Achievable: PreSee's diagnostics provide a wide range of data concerning the detection of high chances of heart disease and other health issues. And of course, it is collected responsibly and properly by professionals, therefore it is reliable. This allows us to identify actual risk groups with meaningful health-related features.

Realistically, It is unlikely that no risk-groups will be found in our model. Of Course, we do hope for the best health for our population and for workers of businesses throughout the globe, but heart diseases are the top 2nd reason of death in adult population in Israel[1] and even more specifically, in the year 2021 there has been a rise of 15% in cardiac events (most likely due to COVID-19 IMPACT). 48% are overweight[2] 20% smoke.

Finding groups will be achievable, and finding an unhealthy habit within these groups is likely to be the same. Our suggested solution is itself a tool that is meant to illuminate an ACHIEVABLE IMPACT.

Relevant: In "Achievable" section above are detailed some of the worrying statistics of the general population. BUT our model is meant to focus on risk groups and suggest a vector to improve that will likely help the GENERAL health of all workers, yet is SURE to affect the risk groups. Why are these risk groups relevant to focus on? Because family history of heart diseases suggests a DOUBLED risk of occurrence of cardiovascular events[3] According to an NCBI study[4], There is a strong correlation between ECG abnormalities and future sudden cardiac deaths. These categories that we define as "unchangeable", highlight the groups within the population that are the most prone to serious health issues. It is only natural to prioritize solutions that will affect them the most.

Even in a wide-scale change of health-culture within one of PreSee's business customers, the ability to simultaneously work on specific risk-groups and risk-groups-related factors is an important feature.

Furthermore, COVID-19 has a higher chance of deadly impact on different risk-groups, and as the data gathers on the virus's different variants, taking care of these groups should definitely be a priority.

Time Bound: Our model is based on ONE-TIME tests of all workers of the company, performed at a specific time. Therefore, we do not need a constant stream of information nor we do not rely on future data.

Furthermore, our model can integrate data to achieve immediate results, as long as all of a company's workers stats are reported properly. In our last section of this report we suggest a more measurable way of collecting information about worker's habits (such as defining smoking on a scale of 1-5 levels) that can allow to precisely measure what would be the most significant criteria, AND how it changes the risk-groups OVERALL SCORE (and the general overall score). Implementing this can allow a business to SET REALISTIC GOALS for NEXT YEAR'S TESTS.

This allows to intelligently measure effectiveness of decisions made by business owners on their company's health-culture, to measure YEAR-TO-YEAR PROGRESS, and to investigate the effects on risk groups simultaneously and separately form the general group.

Date Exploration

We started by exploring the data for getting better understanding of the information we got.

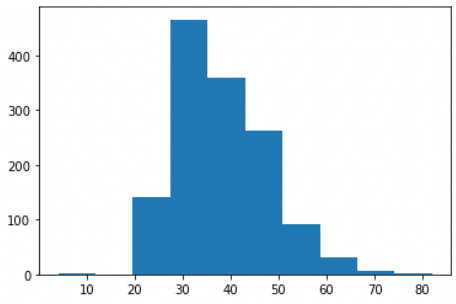
The dataset contains 1359 records with 23 fields for each data point.

- 494 of the people are women and 866 are men.
- 194 are smoking while 1165 are not.
- 205 of them reported levels of 4, 5 of work stress, and 540 reported levels of 2 or less.
- 424 reported exercise levels of 4, 5, and 579 reported exercise levels of 2 or less.

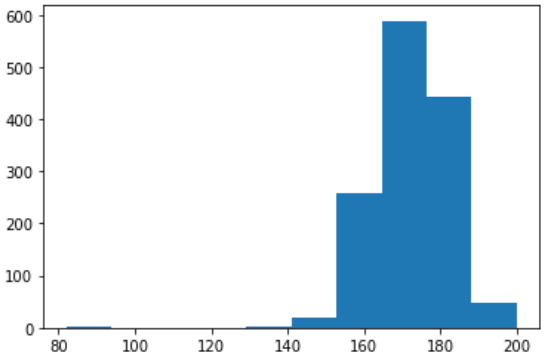
We wanted to learn about the distribution of our data in few features:

Age distribution:

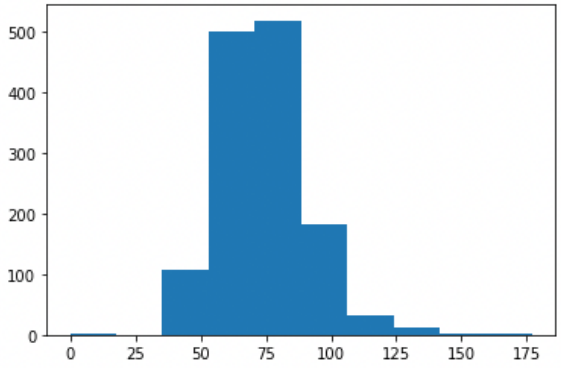
```
data = pd.read_csv(input_file)
plt.hist(data["age"])
plt.show()
```



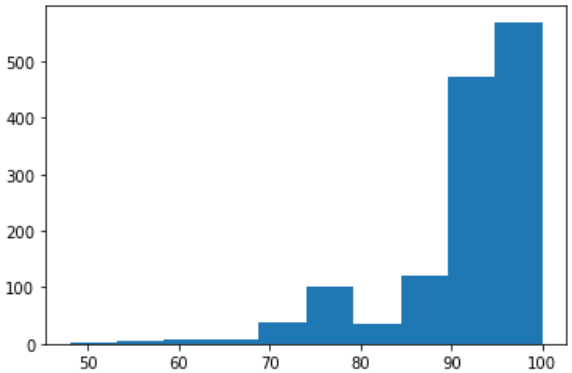
Height distribution:



Weight distribution:

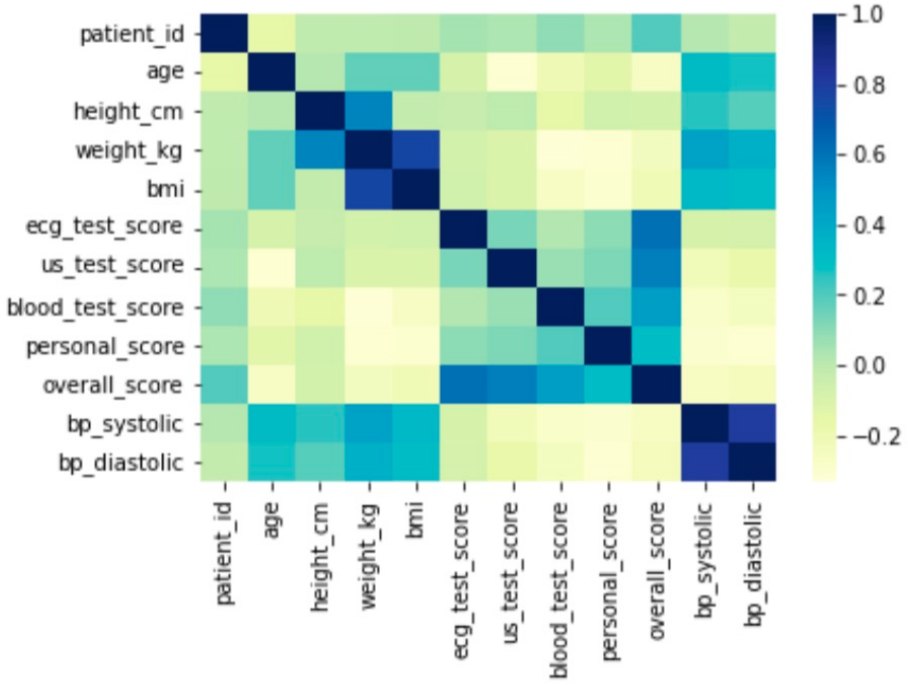


Overall score distribution:



In addition, we created a heat-map in order tp check the corolation between different segments of our data.

```
In [6]: import seaborn as sns
        # Plot the correlation heatmap
        sns.heatmap(df.corr(), cmap="YlGnBu", annot=);
```



We found some intresting connections such as -

- Ecg score to overall score
- Weight and BP
- BP and age

In addition, the heatmap suggests that there isn't strong linear connection between the different tests that PreSee does, which implies that all of them giving new information – therefore – important.

Furthermore, the map helped us to better understand the affects of each feature on the overall score.

Data Preparation

After the basic data exploration, we realized that we need to change our data to be NUMERIC, since our model can't deal with English letters, only numbers.

- First, we changed the ‘gender’ columns from male or female to 0 or 1 accordingly.
- Second, we changed the ‘smoking’, ‘heart_disease_hist’, ‘heart_disease_family_hist’, ‘bp_medication’ and ‘diabetes’ columns from Yes or No to 0 or 1 accordingly.
- Third, we changed ‘work_stress_level’ and ‘exercise_level’ columns from X/5 to X*0.20.

After doing so, since each measure has a different scale, we NORMALIZED the results. For example: ‘heart_disease_hist’ column was at that point 0 or 1 for each person. This is obviously a very important indicator when evaluating chance of heart failure[5]. But the technical difference between having it or not - is only 1 unit!
However, the ‘height’ column, which is not a significant indicator concerning heart disease, has a range of values between 150 – 195, meaning differences can be up to 45 units!

At first, we used Excel formulas that create a binary form for every relevant column:

- ‘bmi_bit’ = if normal BMI (18.5-25) add 1, else, 0. Excel formula: (IF(AND(bmi>18.5,bmi<25),1,0))
- ‘Ecg_score_bit’ = Since more than 75% of people got score 100/100 appendix 1 and there is no scale like that online (probably its PreSee's scale) then we decided that if its 100 than add 1, else add 0. Excel formula: IF(acg=100,1,0)
- ‘us_test_score_bit’ = Same as the Ecg_score_bit appendix 2.
  Excel formula: IF(us_test_score=100, 1, 0)
- ‘blood_test_score_bit’ = If the score was higher (better) then median score (=87) appendix 3
- ‘bp_systolic_bit’ = If the systolic bp was too high (> 125)[6] add 0, else 1.
  Excel formula : IF(bp_systolic > 125, 0, 1)

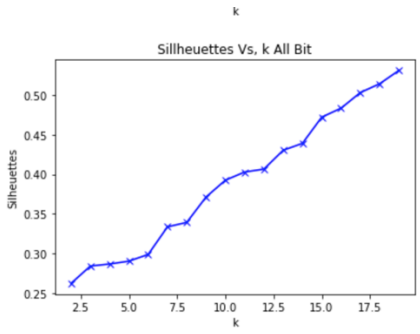
- ‘bp_diastolic_bit’ = If the diastolic bp was too low (< 80)[6] add 0, else 1.

  Excel formula : IF(AE2<80, 1, 0)

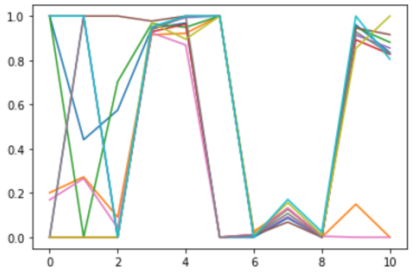
Then, we ran our model (that we will talk more about in the next section) with the binary data.

We noticed a problem. The K-means algorithm we used didn't provide useful results. The algorithm kept splitting the data to similar sized groups in a linear way as long as we kept increasing the K. In addition to that, there wasn’t a meaningful distinction between groups.

Here are the results from this run:




```
Counter({2: 118,
         1: 154,
         3: 168,
         7: 112,
         6: 237,
         8: 96,
         0: 127,
         4: 152,
         5: 131,
         9: 41})
```



So, instead of using the binary-formed information, we used a normalization formula:

(cell_value – cell_col_avg) / cell_col_stdev

This gave us the option to use the same scale for all columns while keeping the scaled property of the data. In addition, it helped us find outliers data points easily.

We cleaned **65 outlier** data-points. Most were obvious mistakes, like weight = 0, or ‘null’ values, while the minority was outliers like 1 person who is 82 years old in the company, who is significantly older than any other worker.

Now, using python code we extracted only the features that aren't influenced directly by the life routine of the person, meaning they are "UNCHANGEABLE" factors:

```
df = original_df[["gender", "bmi_norm", "acg_norm" ,  "us_norm", "blood_norm" , "heart_hist_factor" , "family_hist_n
df
```

Out[56]:

	gender	bmi_norm	acg_norm	us_norm	blood_norm	heart_hist_factor	family_hist_norm	bp_med_norm_half	diabetes_factor	bp_sys_norm	bp_di_norm
0	1.000000	-1.727775	0.239149	0.221099	1.591687	3.0	0.356637	0.088143	2.0	-0.517110	-0.209134
1	0.000000	-1.194470	0.239149	0.221099	0.715967	3.0	-2.801801	0.088143	2.0	0.574966	0.677458
2	0.000000	-0.661166	0.239149	0.221099	0.059177	3.0	0.356637	0.088143	2.0	-0.175837	0.184907
3	1.000000	0.405443	0.239149	0.221099	0.825432	3.0	0.356637	0.088143	2.0	-1.609187	-1.588277
4	0.000000	0.405443	0.239149	0.221099	-1.801728	3.0	0.356637	0.088143	2.0	-0.175837	-0.110624
...	...	...	...	...	...	...	...	...	...	...	...
1290	1.000000	-1.194470	0.239149	0.221099	0.934897	3.0	-2.801801	0.088143	2.0	0.916240	0.775968
1291	0.000000	1.738705	0.239149	0.221099	-0.050288	3.0	0.356637	0.088143	2.0	-0.107582	0.381927
1292	0.000000	0.138791	0.239149	0.221099	0.387572	3.0	0.356637	0.088143	2.0	0.165437	-0.406155
1293	0.366589	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.000000
1294	0.482059	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.000000	0.000000

295 rows × 11 columns

We also decided to not use the height and weight features, since the BMI includes both in a much more informative way.

The Model

To answer our main question, we needed to split the data into distinctive risk-groups. We wanted every group to have its own uniqueness and inside every group the data points should be as similar as possible to each other.

Since there are plenty of options to divide the data to groups with the features mentioned above, we decided to use the ML algorithm K-means, with the 'Euclidean' distance metric to evaluate the silhouettes of the clusters.

While approaching the task of building the K-means model, we had 2 main questions:

1. Should we separate men and women?
2. What is the right 'K' division to choose?

So, for each K (in range 2-10) we built 3 models: one with male only, second with female only, and third all together. Then we checked which model is best.

Code:

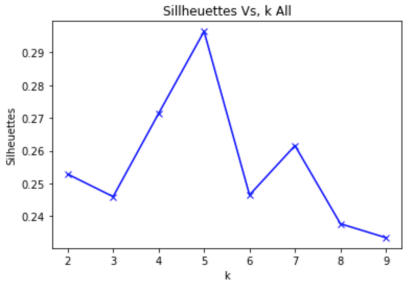
```
distance_metric='euclidean'
k_range = np.arange(2, 10, 1)
```

```
# Build model function
def get_clusterer(points,k):
    clusterer = KMeans (n_clusters=k)
    preds = clusterer.fit_predict(points)
    return clusterer,preds

clusterer,preds=get_clusterer(df,5)
clusterer_male,preds_male=get_clusterer(df_male,1)
clusterer_female,preds_female=get_clusterer(df_female,1)
#TEST for Bits information
#clustererBit,predsBit = get_clusterer(dfBit,5)
```

```
#bulding the model for each K in range 2 to 10.
def get_silheuettes(df,preds):
    Silhouettes = []
    for k in k_range:
        clusterer,preds=get_clusterer(df,k)
        Silhouettes.append(silhouette_score (df, preds, metric='euclidean'))
    return Silhouettes

Silhouettes = get_silheuettes(df,preds)
Silhouettes_male = get_silheuettes(df_male,preds_male)
Silhouettes_female = get_silheuettes(df_female,preds_female)
#TEST for Bits information
#SilhouettesBit =  get_silheuettes(dfBit,predsBit)
```

```
# plot the silhouttes results.
def show_silhouettes (Silhouettes, str):
    plt.plot(k_range, Silhouettes, 'bx-')
    plt.xlabel('k')
    plt.ylabel('Silheuettes')
    plt.title('Sillheuettes Vs, k' + " " + str)
    plt.show()


show_silhouettes(Silhouettes, "All")
show_silhouettes(Silhouettes_male, "Male")
show_silhouettes(Silhouettes_female, "Female")
#TEST for Bits information
#show_silhouettes(SilhouettesBit, "All Bit")
```

Results:



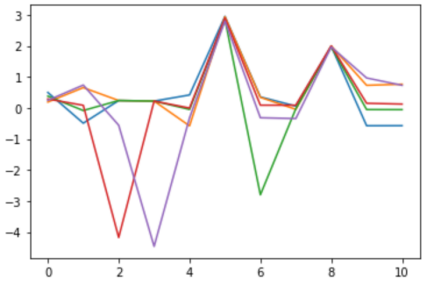
The similarities of the graphs suggested that there is no real benefit in separating male from female. Furthermore, we found that K=5 is best K of all.

We built the model for K=5 and looked at the uniqueness of each cluster:

```
# Build model for K=5
K=5
def get_clusterer(points,k):
    clusterer = KMeans (n_clusters=k)
    preds = clusterer.fit_predict(points)
    return clusterer,preds
clusterer,preds=get_clusterer(df,K)
```

```
def show_results (clusterer):
    centers = clusterer.cluster_centers_
    print('centroids:')
    for i in range(K):
        print(i,':',centers[i,:],'\n')
    for i in range(K):
        plt.plot(centers[i,:])

show_results(clusterer)

#df
```

```
centroids:
0 : [ 0.49818399 -0.49126573  0.23839416  0.2204016   0.41898185  2.95268139
  0.35551222  0.06020899  1.99053628 -0.57404476 -0.57206092]

1 : [ 0.19036145  0.65538987  0.23914857  0.22109908 -0.57809385  2.94216867
  0.34141587 -0.04564787  2.          0.72940267  0.76528589]

2 : [ 0.38095238 -0.08341907  0.23914857  0.22109907 -0.04768156  2.85714286
 -2.80180093 -0.02782032  1.96825397 -0.05232791 -0.05433243]

3 : [ 2.88135593e-01  8.45568175e-02 -4.17826713e+00  2.21099075e-01
 -3.90439107e-03  2.89830508e+00  8.89730055e-02  8.81425340e-02
  2.00000000e+00  1.50398137e-01  1.24798681e-01]

4 : [ 0.24590164  0.73766604 -0.55743459 -4.4654764  -0.31946412  2.80327869
 -0.31647252 -0.34301117  1.96721311  0.96659171  0.73074992]
```



We received 3 clusters that are defined by one main feature, and 2 more groups that don't have any uniqueness.

The 3 clusters that have a distinguishing feature are: low US score, low ECG score, and family hist.

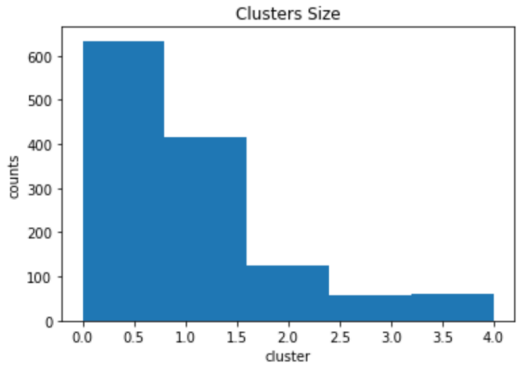
The 2 other clusters are over-all healthy people that are not in our defined risk groups.

At the next stage, before analyzing each of the clusters separately, we wanted to understand what the distribution of the people is in each clusters.

```
def show_labels_results (clusterer):
    print('labels:', clusterer.labels_, '\n')
#show_labels_results(clusterer)

print(len(clusterer.labels_))

##How many data points we have in each group

labels = clusterer.labels_

collections.Counter(labels)
```

```
1295

Counter({0: 634, 2: 126, 1: 415, 4: 61, 3: 59})
```

```
def plot_histogram(labels):
        n_unique_categories = 5
        plt.hist(labels, bins=n_unique_categories)
        plt.xlabel("cluster")
        plt.ylabel("counts")
        plt.title("Clusters Size")
plot_histogram(labels)
```



We noticed that the 2 non-distinctive groups are the majority (80%) while the family hist. group is 10%, low US score is 5%, and low ECG score is 5%.

So, since most of the people are in the clusters that don't have a distinctive feature, we chose to focus on the other 3 risk groups and explore them.

In order to decide which habit we should focus on, we now explored each cluster separately.

We did it in few steps:

1. Creating a new column at the data frame with the K-means label for each raw.

```
# insert label column
new_df = original_df
new_df.insert(0, "Kmeans_label", labels)
```

```
new_df
```

	Kmeans_label	patient_id	gender	gender_norm	age	age_norm	height_cm	weight_kg	bmi	bmi_norm	...	diabetes_factor	diabetes_norm
0	0	1.0	1.000000	1.313968	41.000000	0.365088	170.0	52.0	18.000000	-1.727775	...	2.0	0.05568
1	2	2.0	0.000000	-0.760465	26.000000	-1.336637	173.0	60.0	20.000000	-1.194470	...	2.0	0.05568
2	0	3.0	0.000000	-0.760465	38.000000	0.024743	172.0	65.0	22.000000	-0.661166	...	2.0	0.05568
3	0	4.0	1.000000	1.313968	39.000000	0.138191	160.0	67.0	26.000000	0.405443	...	2.0	0.05568
4	1	5.0	0.000000	-0.760465	33.000000	-0.542499	176.0	79.0	26.000000	0.405443	...	2.0	0.05568
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1290	2	1357.0	1.000000	1.313968	36.000000	-0.202154	160.0	51.0	20.000000	-1.194470	...	2.0	0.05568
1291	1	1358.0	0.000000	-0.760465	27.000000	-1.223188	169.0	88.0	31.000000	1.738705	...	2.0	0.05568
1292	0	1359.0	0.000000	-0.760465	32.000000	-0.655947	176.0	77.0	25.000000	0.138791	...	2.0	0.05568
1293	0	0.0	0.366589	0.000000	37.781903	0.000000	0.0	0.0	24.479505	0.000000	...	0.0	0.00000
1294	0	0.0	0.482059	0.000000	8.814589	0.000000	0.0	0.0	3.750202	0.000000	...	0.0	0.00000

1295 rows × 44 columns

2. Creating unique data frame for each group:

```
def getSize(df, newdf):
    print('overall',df.shape[0]*100/newdf.shape[0], '% from all people')
```

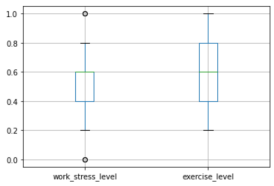
```
 # family hist.
df_fm = new_df[new_df["Kmeans_label"] == 2];
print(" --------- family hist ------------")
getSize(df_fm, new_df)

## low US score
df_us = new_df[new_df["Kmeans_label"] == 4];
print(" --------- low US score ------------")
getSize(df_us, new_df)


## low ECG score
df_ecg = new_df[new_df["Kmeans_label"] == 3];
print(" --------- low ECG score ------------")
getSize(df_ecg, new_df)

## low ECG score

df_others = new_df[new_df["Kmeans_label"] <= 1 ];

print(" --------- others ------------")
getSize(df_others, new_df)
```

```
 --------- family hist ------------
overall 9.72972972972973 % from all people
 --------- low US score ------------
overall 4.710424710424711 % from all people
 --------- low ECG score ------------
overall 4.555984555984556 % from all people
 --------- others ------------
overall 81.003861003861 % from all people
```

3. Display the relevant information statistics about each group, while focusing on the changeable features: weight, smoking, work stress level, and exercise level.

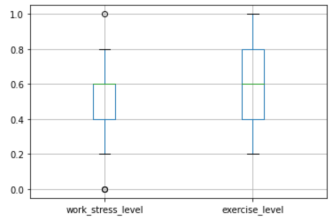
```
def plot_statistics(dataframe,selected_columns):
    print('7 numbers statistics\n',dataframe.describe())

    #print('Box and Whisker\n',dataframe.boxplot())
    print('Box and Whisker for selected columns\n',dataframe.boxplot(column=selected_columns) )


feat = ["age","weight_kg","smoking", "work_stress_level", "exercise_level"]
print(" --------- family hist ------------")
plot_statistics(df_fm[feat],['work_stress_level', 'exercise_level'])
```

```
 --------- family hist ------------
7 numbers statistics
              age   weight_kg     smoking  work_stress_level  exercise_level
count  126.000000  126.000000  126.000000         126.000000      126.000000
mean    38.230159   72.849206    0.801587           0.523810        0.539683
std      8.720700   14.858838    0.400397           0.209925        0.237682
min     21.000000   43.000000    0.000000           0.000000        0.200000
25%     31.250000   63.000000    1.000000           0.400000        0.400000
50%     37.000000   72.000000    1.000000           0.600000        0.600000
75%     44.000000   84.000000    1.000000           0.600000        0.800000
max     60.000000  110.000000    1.000000           1.000000        1.000000
Box and Whisker for selected columns
 AxesSubplot(0.125,0.125;0.775x0.755)
```



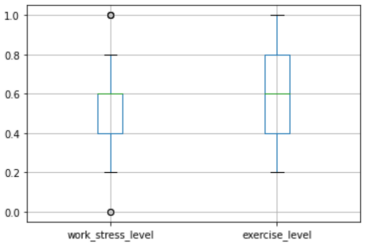

```
print(" --------- low US score ------------")
plot_statistics(df_us[feat],['work_stress_level', 'exercise_level'])

#older people.
#practice like avg.
#work stress level meadinan 3/5 but avg 0.5 ike everyone.
```

```
 --------- low US score ------------
7 numbers statistics
              age   weight_kg     smoking  work_stress_level  exercise_level
count  61.000000   61.000000  61.000000          61.000000       61.000000
mean   49.147541   80.983607   0.885246           0.491803        0.603279
std     8.481030   15.355229   0.321370           0.205179        0.269547
min    30.000000   48.000000   0.000000           0.000000        0.200000
25%    42.000000   70.000000   1.000000           0.400000        0.400000
50%    48.000000   82.000000   1.000000           0.600000        0.600000
75%    56.000000   90.000000   1.000000           0.600000        0.800000
max    65.000000  125.000000   1.000000           1.000000        1.000000
Box and Whisker for selected columns
 AxesSubplot(0.125,0.125;0.775x0.755)
```



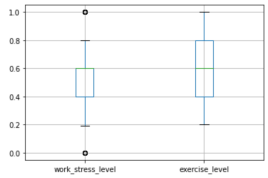

```
print(" --------- low ECG score ------------")
plot_statistics(df_ecg[feat],['work_stress_level', 'exercise_level'])
```

```
 --------- low ECG score ------------
7 numbers statistics
              age   weight_kg    smoking  work_stress_level  exercise_level
count  59.000000   59.000000  59.000000          59.000000       59.000000
mean   39.186441   74.932203   0.745763           0.535593        0.572881
std     9.631977   14.138312   0.439169           0.239081        0.261185
min    24.000000   49.000000   0.000000           0.000000        0.200000
25%    31.000000   65.500000   0.500000           0.400000        0.400000
50%    38.000000   75.000000   1.000000           0.600000        0.600000
75%    46.000000   84.000000   1.000000           0.600000        0.800000
max    64.000000  107.000000   1.000000           1.000000        1.000000
Box and Whisker for selected columns
 AxesSubplot(0.125,0.125;0.775x0.755)
```




```
print(" --------- Others ------------")
plot_statistics(df_others[feat],['work_stress_level', 'exercise_level'])
```

```
 --------- Others ------------
7 numbers statistics
              age    weight_kg      smoking  work_stress_level  \
count  1049.000000  1049.000000  1049.000000        1049.000000
mean     36.960530    72.683508     0.863877           0.525374
std       8.362687    14.722332     0.342591           0.187107
min       8.814589     0.000000     0.000000           0.000000
25%      31.000000    62.000000     1.000000           0.400000
50%      36.000000    72.000000     1.000000           0.600000
75%      42.000000    83.000000     1.000000           0.600000
max      65.000000   125.000000     1.000000           1.000000

       exercise_level
count     1049.000000
mean         0.549480
std          0.252555
min          0.200000
25%          0.400000
50%          0.600000
75%          0.800000
max          1.000000
Box and Whisker for selected columns
 AxesSubplot(0.125,0.125;0.775x0.755)
```



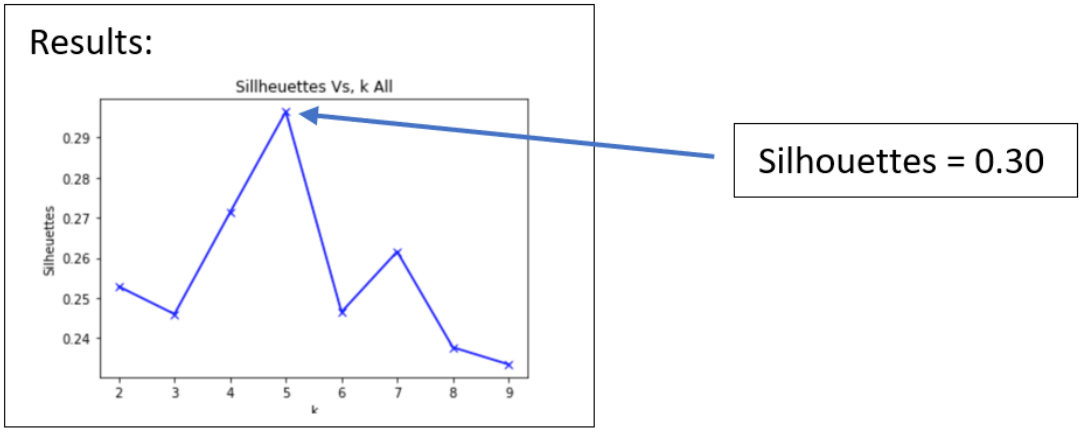
Results

The results were that the 3-risk group are exercising similar amount as the people in the ‘others’ (healthy) group and have the same amount of work stress as well. But, at the same time we saw that 20% of the people with family history and 25.5% of the people with the low ECG are smokers, which is way over the average of 13.5% of the general group!

Therefore, since smoking has huge effect on health, especially on those that are already in risk groups, we suggest this company to try to focus on reducing smoking to achieve a quick, meaningful impact on risk-groups while also improving the general health of all the company's workers.

Evaluation

Exploring our data, we found the best 'K'-division for our K-Means model using male and Female together (as well as female and male separately) is **K=5**:



We notice that for K=5 we get the highest silhouette - 0.30.

0.3 isn't a very high silhouet, so the clusters aren't necessarily well defined. Therefore, we weren't surprised when we saw that for every run of the algorithm we noticed a slight difference in the size of the groups.

Despite the differences, there weren't any changes in the main features that characterized the groups. This implies that our intentions of improving the "unchangeable" factors are indeed relevant and accurate.

The Extra Feature:

In addition to our main model, we wanted to measure the impact that our suggestions will have on the health of the company's employees in a numeric way. In other words, to accurately measure what factor, if improving it slightly will increase the average overall score of the risk group/groups.

In order to perform this, we suggest to iterate through every unhealthy habit, and "improve" it's value for every person in the risk group, by only 1 "TICK" (minor change):

- Training -> add ⅕ or leave ‘1’.
- Work-stress -> reduce ⅕ or leave ‘0’.
- Weight -> reduce by 7% - if high BMI. Else - no change required.
- Smoking - reduce 1/5 level or leave ‘0’.

After improving each habit, we save the new AVERAGE OVERALL SCORE of the risk-groups. Before improving the next unhealthy habit, we restore the original value.

The feature that the 1-TICK change resulted in the best overall score will be the main feature that we would suggest to focus on in order to achieve a quick, measurable impact.

In order to be able perform this, we suggest that Pre’See data collection will reflect these fractals. Practically, only the ‘smoking’ criteria needs modification.

Currently, PreSee decided to ask the patients whether they are smoking or not, therefore we only got a binary answer. We suggest that for this model they will start collecting the smoking habits of people in a more precise way.
This will give us the opportunity to use the “TICK” method on this feature as well. We realize that “stop smoking” is not an easily achievable task, yet reducing smoking habit by 1 TICK is much more practical. An example for a smoking-scale:

Non smoking = 5

1-2 cigarette per day = 4

3-5 cigarette per day = 3

6-10 cigarette per day = 2

10+ cigarette per day = 1

This will create our model even more **measurable,** and we’ll know exactly how much our suggestion will improve the overall score of the company’s employees. It makes it possible to intelligently measure the effectiveness of business owners' decisions about

their company's health culture, to measure year-by-year progress, and to study the effects on risk groups simultaneously and separately from the general group.

*Also, We need to be careful about data bias and pay attention to how the question is asked in order to get a credible answer.

Bibliography

1. https://publichealth.doctorsonly.co.il/2018/09/148469/
2. https://news.walla.co.il/item/3432891
3. https://www.ucihealth.org/blog/2017/02/family-history-heart-attacks
4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6111047/
5. https://www.jstor.org/stable/3703625
6. https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings

Appendix

1.

```
count    1359.000000
mean       96.512141
std        14.044626
min        40.000000
25%       100.000000
50%       100.000000
75%       100.000000
max       100.000000
Name: ecg_test_score,
```

2.

```
count    1359.000000
mean       97.317145
std        11.337191
min        36.000000
25%       100.000000
50%       100.000000
75%       100.000000
max       100.000000
Name: us_test_score,
```

3.

```
count    1359.000000
mean       83.320088
std         9.313385
min        39.000000
25%        79.000000
50%        87.000000
75%        90.000000
max       101.000000
Name: blood_test_score,
```