Engenharia da Computação

Carlos Henrique Gomes dos Santos/1760092 /ENG 171i

Tema: Previsão de produtividade dos funcionários de vestuário

Birigui – SP

2021

Carlos Henrique Gomes dos Santos

Previsão de produtividade dos funcionários de vestuário

Relatório apresentado como requisito parcial para avaliação final – disciplina Mineração de Dados, do curso superior em Engenharia da Computação, ministrado pelo Prof Dr. Murilo Varges da Silva

Birigui – SP
2021

1. Resumo

Foi escolhido no site *UCI – Machine Learning Repository* a base de dados Previsão de produtividade dos funcionários de vestuário, onde a mesma deveria ter alguns dados ausentes, utilizando o algoritmo em linguagem *Python* fornecido pelo Prof. Dr Murilo para o processamento da base de dado escolhida, gerando dados estatísticos e filtrando dados da base que podem ser utilizadas para gerar informações relevantes para o processo.

Para isso então foi feito o download da base de dados e verificado quais atributos são relevantes, após isso é realizado o estudo do algoritmo fornecido, e a extração das informações com base nos dados fornecidos.

2. Informações sobre a base de dados

A Indústria de Vestuário é um dos principais exemplos da globalização industrial desta era moderna. É uma indústria altamente intensiva em mão-de-obra com muitos processos manuais. Satisfazer a enorme demanda global por produtos de vestuário depende principalmente do desempenho de produção e entrega dos funcionários nas empresas de fabricação de vestuário. Por isso, é altamente desejável entre os tomadores de decisão da indústria de vestuário acompanhar, analisar e prever o desempenho da produtividade das equipes de trabalho em suas fábricas.

3. Informações de atributos

I. date : Data em MM-DD-YYYY
II. day: Dia da Semana
III. quarter: Uma parte do mês. Um mês foi dividido em quatro partes
IV. departament : Departamento associado com a intância
V. team_no : Número de equipe associado com a intância
VI. no_of_workers : Número de trabalhadores em cada equipe
VII. no_of_style_change : Número de mudanças no estilo de um determinado produto
VIII. targeted_productivity : Produtividade direcionada definida pela Autoridade para cada equipe para cada dia.
IX. smv : Valor do minuto padrão, é o tempo alocado para uma tarefa
X. wip : Trabalho em andamento. Inclui o número de itens inacabados para produtos

XI. over_time : Representa a quantidade de horas extras por cada equipe em minutos

XII. incentive : Representa a quantidade de incentivo financeiro (no BDT) que permite ou motiva um determinado curso de ação.

XIII. idle_time : A quantidade de tempo em que a produção foi interrompida por diversos motivos

XIV. idle_men : O número de trabalhadores que estavam ociosos devido à interrupção da produção

XV. actual_productivity : O real % de produtividade que foi entregue pelos trabalhadores. Varia de 0-1.

XVI. Productivity: foi criado para se ter dados binários, actual_productivity maior que 5, o campo recebe 1, se for menor recebe 0

4. **Pré-processamento**

A primeira etapa do algoritmo é mostrar as 15 primeiras linhas da base de dados, logo após é mostrado informações gerais referentes a base nome de coluna, tipo de dado (*objetc, float, int*), após é mostrado a descrição de dados, quantidade de dados, é apresentado também a quantidade de dados faltantes em cada uma das colunas (figura 1).

Figura 1 - Campos da base de dados

```
Data columns (total 16 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   date                   1197 non-null   object 
 1   quarter                1197 non-null   object 
 2   department             1197 non-null   object 
 3   day                    1197 non-null   object 
 4   team                   1197 non-null   int64  
 5   targeted_productivity  1197 non-null   float64
 6   smv                    1197 non-null   float64
 7   wip                    691 non-null    float64
 8   over_time              1197 non-null   int64  
 9   incentive              1197 non-null   int64  
 10  idle_time              1197 non-null   float64
 11  idle_men               1197 non-null   int64  
 12  no_of_style_change     1197 non-null   int64  
 13  no_of_workers          1197 non-null   float64
 14  actual_productivity    1197 non-null   float64
 15  productivity           1197 non-null   int64  
```

Fonte: Elaboração Própria

Conseguimos por meio deste algoritmo outras informações, tais como, valor mínimo e máximo, desvio padrão, e alguns dados referente a média relacionadas a parte dos dados ¼ ½ e ¾ (figura 2).

Figura 2 - Descrição de dados

DESCRIÇÃO DOS DADOS

	team	targeted_productivity	smv	wip	over_time	...
count	1197.000000	1197.000000	1197.000000	691.000000	1197.000000	...
mean	6.426901	0.729632	15.062172	1190.465991	4567.460317	...
std	3.463963	0.097891	10.943219	1837.455001	3348.823563	...
min	1.000000	0.070000	2.900000	7.000000	0.000000	...
25%	3.000000	0.700000	3.940000	774.500000	1440.000000	...
50%	6.000000	0.750000	15.260000	1039.000000	3960.000000	...
75%	9.000000	0.800000	24.260000	1252.500000	6960.000000	...
max	12.000000	0.800000	54.560000	23122.000000	25920.000000	...

Fonte: Elaboração Própria

Podemos também filtrar dados de colunas especificas, gerando informações importantes, que chamamos de *target*, no nosso caso será utilizada a *actual_produtivy*,

Quando se trata da substituição de dados ausentes para completar a base de dados, temos diversas opções, substitui por um número específico de escolha, pela mediana, média ou a moda, foi utilizado em nosso problema a média.

Figura 3: valores ausentes wip

	date	quarter	department	day	team	targeted_productivity	smv	wip	over_time	incentive	actual_productivity
0	1/1/2015	Quarter1	sweing	Thursday	8	0.80	26.16	1108.0	7080	98	0.940725
1	1/1/2015	Quarter1	finishing	Thursday	1	0.75	3.94	NaN	960	0	0.886500
3	1/1/2015	Quarter1	sweing	Thursday	12	0.80	11.41	968.0	3660	50	0.800570
4	1/1/2015	Quarter1	sweing	Thursday	6	0.80	25.90	1170.0	1920	50	0.800382
5	1/1/2015	Quarter1	sweing	Thursday	7	0.80	25.90	984.0	6720	38	0.800125
6	1/1/2015	Quarter1	finishing	Thursday	2	0.75	3.94	NaN	960	0	0.755167
7	1/1/2015	Quarter1	sweing	Thursday	3	0.75	28.08	795.0	6900	45	0.753683
8	1/1/2015	Quarter1	sweing	Thursday	2	0.75	19.87	733.0	6000	34	0.753098
9	1/1/2015	Quarter1	sweing	Thursday	1	0.75	28.08	681.0	6900	45	0.750428
10	1/1/2015	Quarter1	sweing	Thursday	9	0.70	28.08	872.0	6900	44	0.721127
11	1/1/2015	Quarter1	sweing	Thursday	10	0.75	19.31	578.0	6480	45	0.712205
12	1/1/2015	Quarter1	sweing	Thursday	5	0.80	11.41	668.0	3660	50	0.707046
13	1/1/2015	Quarter1	finishing	Thursday	10	0.65	3.94	NaN	960	0	0.705917
14	1/1/2015	Quarter1	finishing	Thursday	8	0.75	2.90	NaN	960	0	0.676667

Fonte: Elaboração Própria

Figura 4:Substituição de dados ausentes

	date	quarter	department	day	team	...	wip	over_time	incentive	no_of_workers	actual_productivity
0	1/1/2015	Quarter1	sweing	Thursday	8	...	1108.000000	7080	98	59.0	0.940725
1	1/1/2015	Quarter1	finishing	Thursday	1	...	1190.465991	960	0	8.0	0.886500
2	1/1/2015	Quarter1	sweing	Thursday	11	...	968.000000	3660	50	30.5	0.800570
3	1/1/2015	Quarter1	sweing	Thursday	12	...	968.000000	3660	50	30.5	0.800570
4	1/1/2015	Quarter1	sweing	Thursday	6	...	1170.000000	1920	50	56.0	0.800382
5	1/1/2015	Quarter1	sweing	Thursday	7	...	984.000000	6720	38	56.0	0.800125
6	1/1/2015	Quarter1	finishing	Thursday	2	...	1190.465991	960	0	8.0	0.755167
7	1/1/2015	Quarter1	sweing	Thursday	3	...	795.000000	6900	45	57.5	0.753683
8	1/1/2015	Quarter1	sweing	Thursday	2	...	733.000000	6000	34	55.0	0.753098
9	1/1/2015	Quarter1	sweing	Thursday	1	...	681.000000	6900	45	57.5	0.750428
10	1/1/2015	Quarter1	sweing	Thursday	9	...	872.000000	6900	44	57.5	0.721127
11	1/1/2015	Quarter1	sweing	Thursday	10	...	578.000000	6480	45	54.0	0.712205
12	1/1/2015	Quarter1	sweing	Thursday	5	...	668.000000	3660	50	30.5	0.707046
13	1/1/2015	Quarter1	finishing	Thursday	10	...	1190.465991	960	0	8.0	0.705917
14	1/1/2015	Quarter1	finishing	Thursday	8	...	1190.465991	960	0	8.0	0.676667

Fonte: Elaboração Própria

5. Redução e normalização

Normalização com Z-Score, obtendo os resultados da figura a seguir.

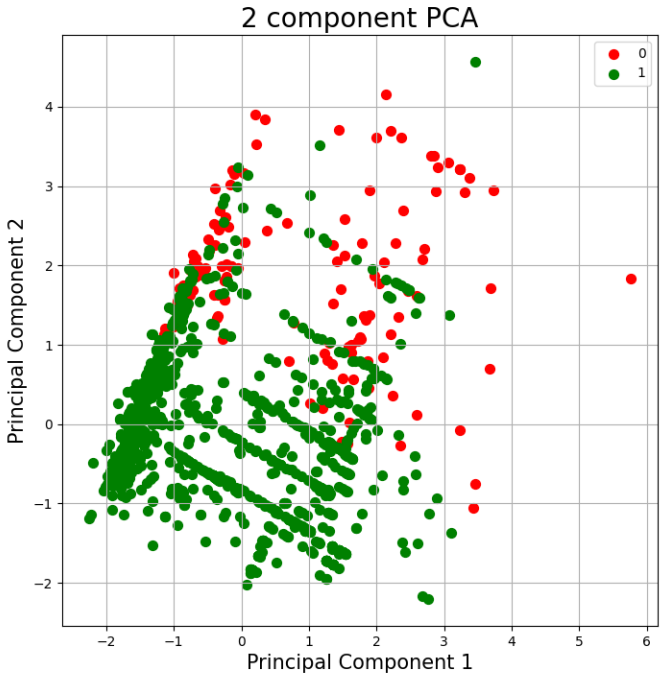
Figura 5: Normalização Z-Score

```
Dataframe Normalized

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 5 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   targeted_productivity  1197 non-null   float64
 1   smv                    1197 non-null   float64
 2   over_time              1197 non-null   float64
 3   actual_productivity    1197 non-null   float64
 4   productivity           1197 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 46.9 KB
None
       targeted_productivity           smv     over_time  actual_productivity  productivity
count           1.197000e+03  1.197000e+03  1.197000e+03         1.197000e+03   1197.000000
mean           -3.531938e-16  9.942849e-17  1.187206e-16        -1.899529e-16      0.888889
std             1.000418e+00  1.000418e+00  1.000418e+00         1.000418e+00      0.314401
min            -6.741257e+00 -1.111853e+00 -1.364470e+00        -2.869632e+00      0.000000
25%            -3.028349e-01 -1.016778e+00 -9.342886e-01        -4.862294e-01      1.000000
50%             2.081510e-01  1.808523e-02 -1.814710e-01         2.176107e-01      1.000000
75%             7.191368e-01  8.408561e-01  7.147405e-01         6.576699e-01      1.000000
max             7.191368e-01  3.610851e+00  6.378797e+00         2.203414e+00      1.000000
   targeted_productivity       smv  over_time  actual_productivity  productivity
0               0.719137  1.014552   0.750589             1.175271             1
1               0.208151 -1.016778  -1.077682             0.865044             1
2               0.719137 -0.333878  -0.271092             0.373436             1
3               0.719137 -0.333878  -0.271092             0.373436             1
4               0.719137  0.990783  -0.790895             0.372357             1
5               0.719137  0.990783   0.643044             0.370887             1
6               0.208151 -1.016778  -1.077682             0.113678             1
7               0.208151  1.190077   0.696816             0.105193             1
8               0.208151  0.439527   0.427953             0.101840             1
9               0.208151  1.190077   0.696816             0.086567             1


Explained variance per component:
[0.4393662178411049, 0.33474130642960115, 0.14655455319738134, 0.0793379225319128]
```

Fonte: Elaboração Própria

Projeção PCA utilizando Z-Score para normalização de dados

Figura 6: PCA com normalização Z-Score



Fonte: Elaboração Própria

Figura 7: Dados PCA redução Z-Score

```
Dataframe PCA

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   principal component 1  1197 non-null   float64
 1   principal component 2  1197 non-null   float64
 2   productivity           1197 non-null   int64
dtypes: float64(2), int64(1)
memory usage: 28.2 KB
None
       principal component 1  principal component 2  productivity
count           1.197000e+03           1.197000e+03   1197.000000
mean            2.374412e-17          -1.187206e-17      0.888889
std             1.326248e+00           1.157620e+00      0.314401
min            -2.251149e+00          -2.204719e+00      0.000000
25%            -1.245757e+00          -7.478856e-01      1.000000
50%            -1.033216e-01          -2.200092e-01      1.000000
75%             1.184165e+00           5.765601e-01      1.000000
max             5.759872e+00           4.567964e+00      1.000000
   principal component 1  principal component 2  productivity
0               0.522922              -1.748895             1
1              -1.666666              -0.019562             1
2              -0.724959              -0.503976             1
3              -0.724959              -0.503976             1
4              -0.204574              -0.728123             1
5               0.696527              -1.199031             1
6              -1.427220               0.454370             1
7               1.100301              -0.783680             1
8               0.452864              -0.468799             1
9               1.106237              -0.771932             1
```

Fonte: Elaboração Própria

Normalização utilizando o Min-Max, mostrado a imagem a seguir

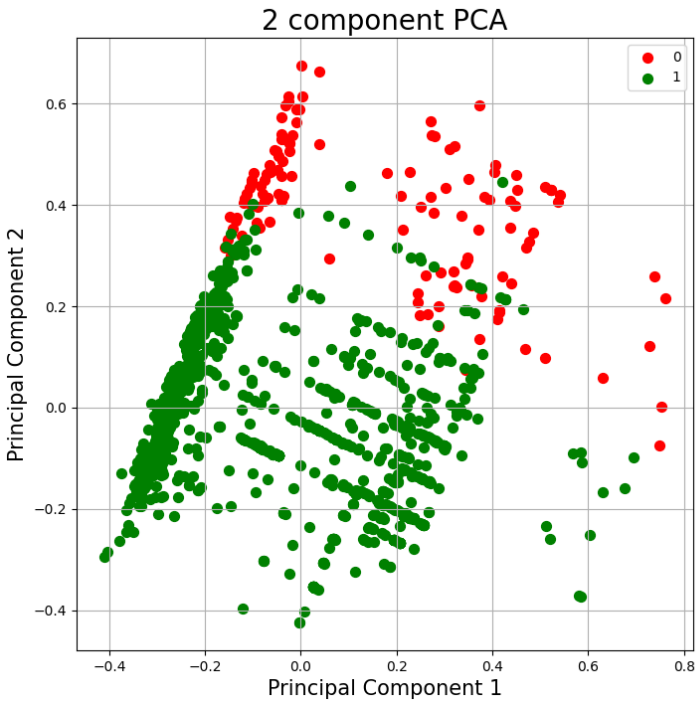
Figura 8: Normalização Min-Max

```
Dataframe Normalized

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   targeted_productivity  1197 non-null   float64
 1   smv                    1197 non-null   float64
 2   over_time              1197 non-null   float64
 3   actual_productivity    1197 non-null   float64
 4   productivity           1197 non-null   int64
dtypes: float64(4), int64(1)
memory usage: 46.9 KB
None
       targeted_productivity          smv    over_time  actual_productivity  productivity
count            1197.000000  1197.000000  1197.000000          1197.000000   1197.000000
mean                0.903606     0.235427     0.176214             0.565663      0.888889
std                 0.134097     0.211832     0.129198             0.197203      0.314401
min                 0.000000     0.000000     0.000000             0.000000      0.000000
25%                 0.863014     0.020132     0.055556             0.469817      1.000000
50%                 0.931507     0.239257     0.152778             0.608558      1.000000
75%                 1.000000     0.413473     0.268519             0.695303      1.000000
max                 1.000000     1.000000     1.000000             1.000000      1.000000
   targeted_productivity       smv  over_time  actual_productivity  productivity
0               1.000000  0.450252   0.273148             0.797332             1
1               0.931507  0.020132   0.037037             0.736180             1
2               1.000000  0.164731   0.141204             0.639274             1
3               1.000000  0.164731   0.141204             0.639274             1
4               1.000000  0.445219   0.074074             0.639062             1
5               1.000000  0.445219   0.259259             0.638772             1
6               0.931507  0.020132   0.037037             0.588071             1
7               0.931507  0.487418   0.266204             0.586398             1
8               0.931507  0.328494   0.231481             0.585737             1
9               0.931507  0.487418   0.266204             0.582727             1


Explained variance per component:
[0.4774504747387705, 0.3477725081606633, 0.11224759689998637, 0.06252942020147328]
```

Fonte: Elaboração Própria

Projeção PCA utilizando mínimos e máximos para a normalização

Figura 9:PCA com normalização min-máx



Fonte: Elaboração Própria

Figura 10: Dados PCA com redução min-máx.

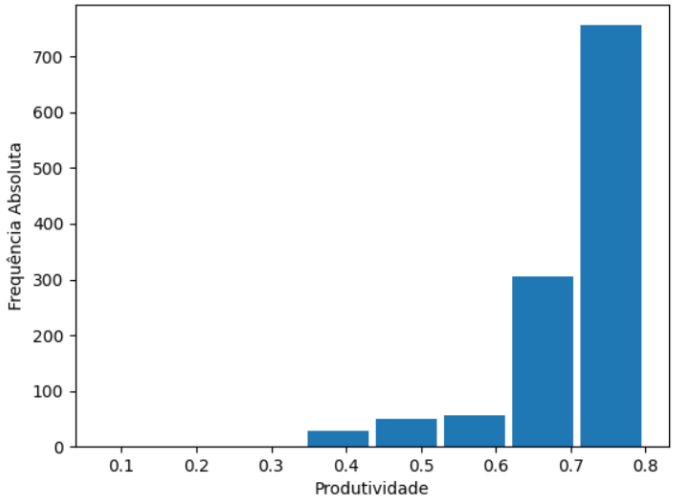
```
Dataframe PCA

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1197 entries, 0 to 1196
Data columns (total 3 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   principal component 1  1197 non-null   float64
 1   principal component 2  1197 non-null   float64
 2   productivity           1197 non-null   int64
dtypes: float64(2), int64(1)
memory usage: 28.2 KB
None
       principal component 1  principal component 2  productivity
count           1.197000e+03           1.197000e+03   1197.000000
mean           -2.448612e-17          -8.904044e-17      0.888889
std             2.377966e-01           2.029500e-01      0.314401
min            -4.096034e-01          -4.241171e-01      0.000000
25%            -2.259543e-01          -1.402938e-01      1.000000
50%            -2.265377e-02          -4.644733e-02      1.000000
75%             2.041985e-01           1.127214e-01      1.000000
max             7.608356e-01           6.750456e-01      1.000000
   principal component 1  principal component 2  productivity
0               0.113439              -0.324703             1
1              -0.300431              -0.049248             1
2              -0.115310              -0.064455             1
3              -0.115310              -0.064455             1
4               0.088408              -0.150519             1
5               0.162570              -0.186987             1
6              -0.245487               0.075150             1
7               0.230756              -0.134724             1
8               0.086524              -0.070881             1
9               0.232118              -0.131641             1
```

Fonte: Elaboração Própria

6. Visualização

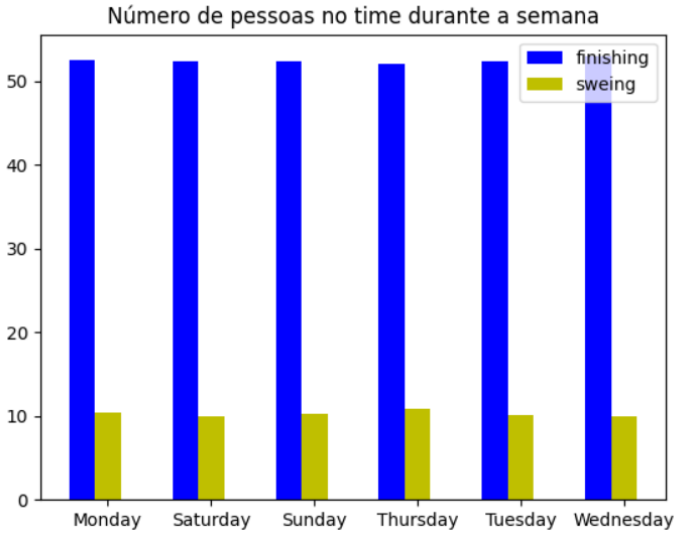
O objetivo desta etapa é abstrair conhecimento da base de dados e apresentar de forma gráfica, conseguindo desta forma extrair o maior número de informação dos dados fornecidos.

Figura 11: Produtividade com base na frequência



Fonte: Elaboração Própria

Figura 12: Número de pessoas no time durante a semana

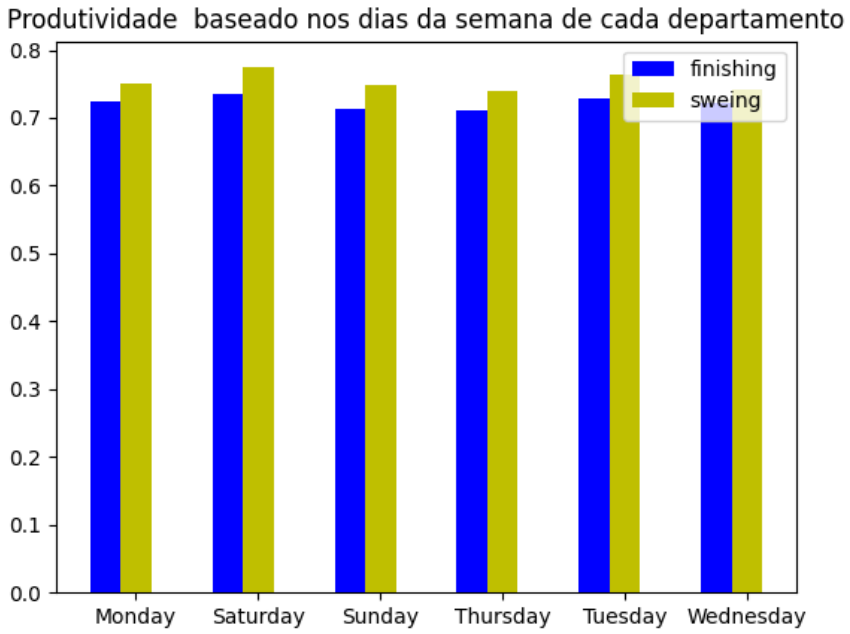


Fonte: Elaboração Própria

A figura 13 apresenta a produtividade de cada setor ao decorrer dos dias das semanas, para elaboração dessa gráfico foi pegada a variável actual_productivity, que se trata da produtividade e separada nos setores de costura e acababento, após isso foi correlacionado com a variável day, que representa os dias da semanas, com isso temos o

gráfico abaixo onde podemos ver qual setor possui um maior valor de produtividade ao decorrer da semana, levando em consideração todos valores da base de dados, conseguimos perceber que o setor de costura apresenta uma leve superioridade em comparação com os de acabamento, levando em conta que o valor varia de zero a um como explicado anteriormente.

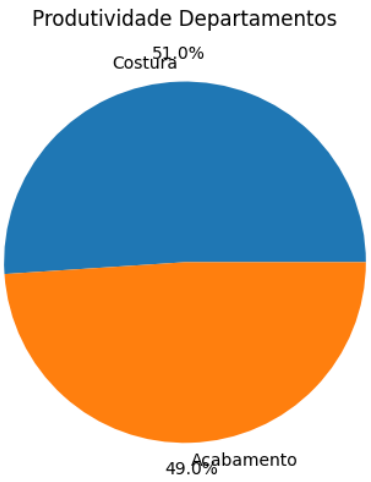
Figura 13: produtividade dias da semana



Fonte: Elaboração Própria

A figura 14 mostra em um gráfico pizza a produtividade de cada setor de uma forma geral, dessa forma é possível perceber que ambos setores apresenta valor de produtividade bem próximos.

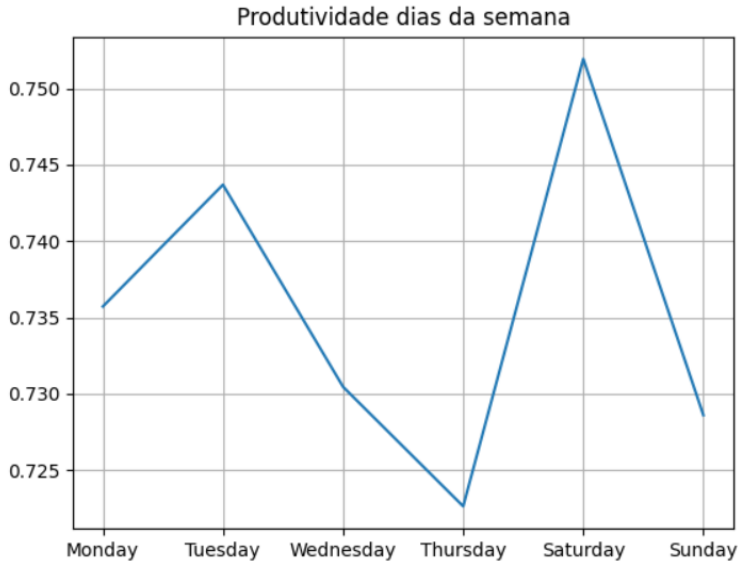
Figura 14: Produtividade Departamentos



Fonte: Elaboração Própria

Podemos também analisar qual dia da semana a empresa possui um maior valor de produtividade, com base nos dados que foi gerada e posteriormente saber o que ocasiona está situação, como demonstrado na figura 15. Onde na quinta-feira é apresentado o pior índice de produtividade e o sábado seu maior índice de produtividade.

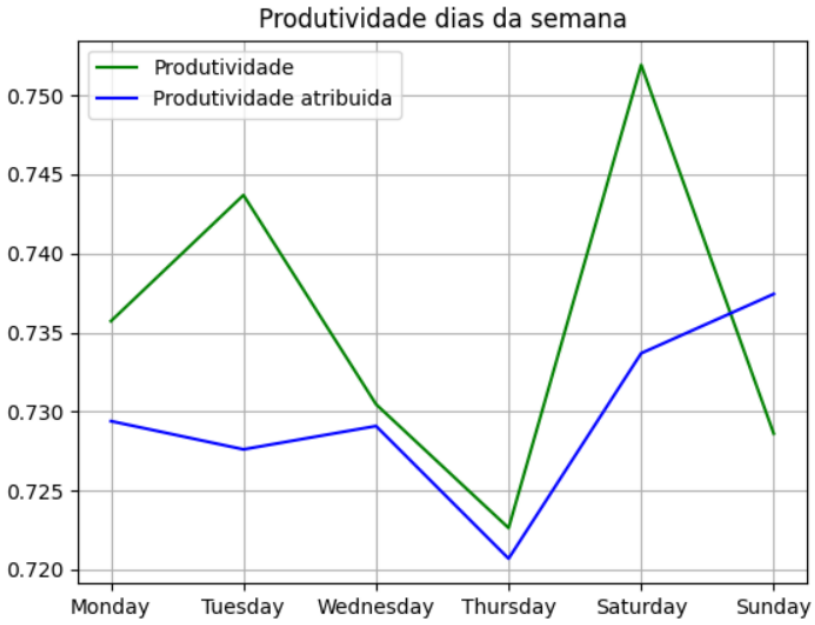
Figura 15: Produtividade geral nos dias da semana



Fonte: Elaboração Própria

Podemos também analisar com a figura 16, comparando o valor da produtividade real com o valor de produtividade que foi atribuído pela autoridade do setor.

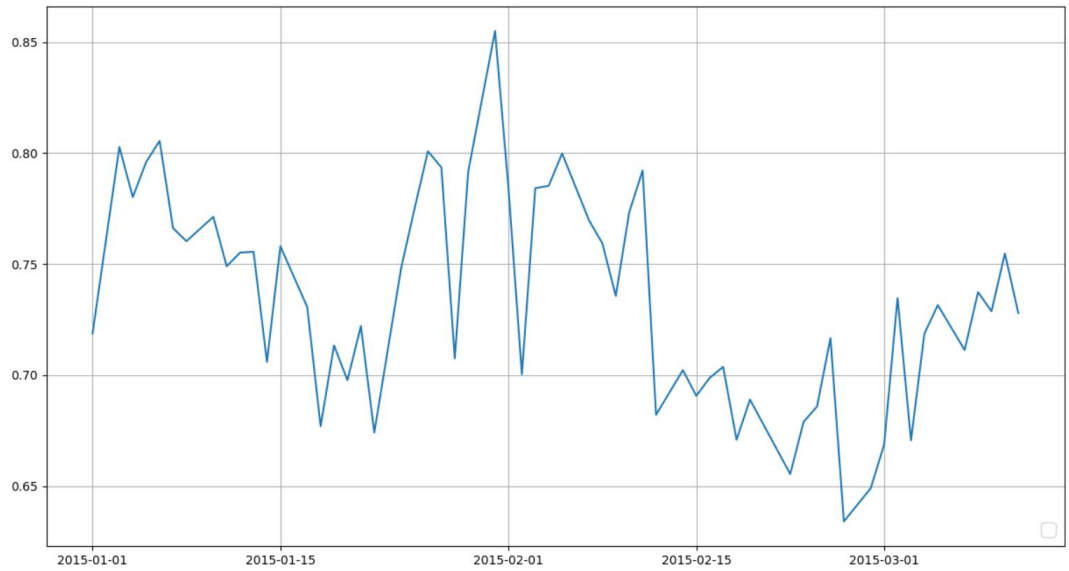
Figura 16: Comparação da produtividade real e a atribuída pela Autoridade



Fonte: Elaboração Própria

Na figura 17 temos a produtividade com base em todos os dias que foram coletados para a base, conseguimos analisar a produtividade em dias específicos e que dia teve melhor produtividade, conseguimos perceber também que a quantidade de informações coletadas poderia ser maior

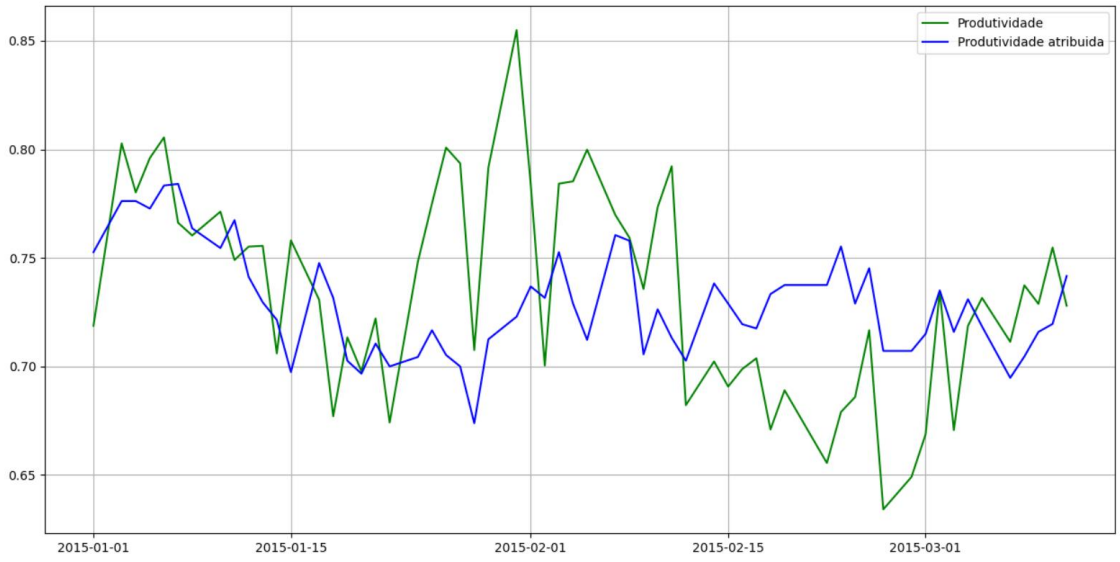
Figura 17: Produtividade ao decorrer dos dias



Fonte: Elaboração Própria

Comparando o valor real de produtividade com o valor atribuído pela autoridade do setor.

Figura 18: Comparação da produtividade real e a atribuída pela Autoridade (decorrer dos dias)



Fonte: Elaboração Própria

7. Descrição

Para esta etapa foi escolhida alguns campos numéricos para obter valores que descreva informações da nossa base de dados, usando média, moda, quartil, mediana, amplitude, variância, desvio padrão, desvio absoluto, covariância e correlação.

Cálculo de média: $\overline{x} = \frac{\sum x_i}{N}$, resultados figura 17.

Figura 19: Média

```
Média
targeted_productivity       0.729632
smv                        15.062172
over_time                4567.460317
incentive                  38.210526
actual_productivity         0.735091
dtype: float64
```

Fonte: Elaboração Própria

A mediana é valor de centro do conjunto de dados, é necessário ordenar os dados, resultados figura 18.

Figura 20: Mediana

```
Mediana
targeted_productivity       0.750000
smv                        15.260000
over_time                3960.000000
incentive                   0.000000
actual_productivity         0.773333
dtype: float64
```

Fonte: Elaboração Própria

Cálculo da amplitude é dados por, $\frac{\max - min}{2}$ como mostrado na figura 19.

Figura 21: Amplitude

```
Amplitude
targeted_productivity       0.730000
smv                        51.660000
over_time                25920.000000
incentive                 3600.000000
actual_productivity          0.886732
dtype: float64
```

Fonte: Elaboração Própria

A moda é o valor que mais se repete na base de dados, resultados figura 20.

Figura 22: Moda

```
Moda
   targeted_productivity   smv  over_time  incentive  actual_productivity
0                    0.8  3.94        960          0             0.800402
```

Fonte: Elaboração Própria

Cálculo do desvio padrão é dado por, $s^2 = \frac{\sum_{i=1}^{n}(x-\overline{x})}{n-1}$, resultado obtidos na figura 21.

Figura 23: Desvio Padrão

```
Desvio padrão
targeted_productivity       0.097891
smv                        10.943219
over_time                3348.823563
incentive                 160.182643
actual_productivity         0.174488
dtype: float64
```

Fonte: Elaboração Própria

Cálculo da variância é dado por, $s = \sqrt{\frac{\sum_{i=1}^{n}(x-\overline{x})}{n-1}}$, resultado figura 22.

Figura 24: Variância

```
Variância
targeted_productivity    9.582641e-03
smv                      1.197540e+02
over_time                1.121462e+07
incentive                2.565848e+04
actual_productivity      3.044603e-02
dtype: float64
```

Fonte: Elaboração Própria

Cálculo do desvio padrão é dado por, $DAM = \frac{\sum_{i=1}^{n}(x-\overline{x})}{n}$

Figura 25: Desvio Absoluto

```
Desvio absoluto
targeted_productivity       0.070840
smv                         9.879867
over_time                2900.524592
incentive                  40.753287
actual_productivity         0.135371
dtype: float64
```

Fonte: Elaboração Própria

Figura 26: Covariância e Correlação

Covariância

	targeted_productivity	smv	over_time	incentive	actual_productivity
targeted_productivity	0.009583	-0.074439	-2.903062e+01	0.513815	0.007201
smv	-0.074439	119.754046	2.473254e+04	57.195571	-0.233124
over_time	-29.030617	24732.539468	1.121462e+07	-2571.212375	-31.674054
incentive	0.513815	57.195571	-2.571212e+03	25658.479053	2.139222
actual_productivity	0.007201	-0.233124	-3.167405e+01	2.139222	0.030446

Correlação

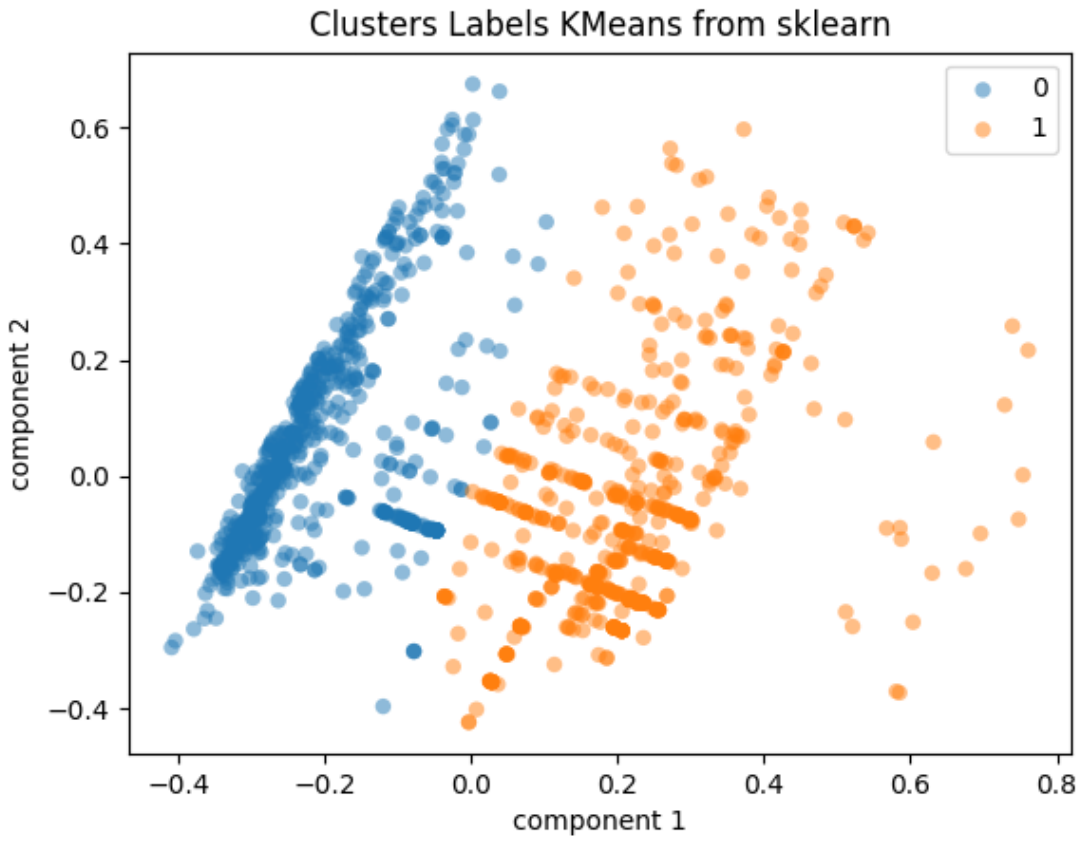
	targeted_productivity	smv	over_time	incentive	actual_productivity
targeted_productivity	1.000000	-0.069489	-0.088557	0.032768	0.421594
smv	-0.069489	1.000000	0.674887	0.032629	-0.122089
over_time	-0.088557	0.674887	1.000000	-0.004793	-0.054206
incentive	0.032768	0.032629	-0.004793	1.000000	0.076538
actual_productivity	0.421594	-0.122089	-0.054206	0.076538	1.000000

Fonte: Elaboração Própria

8. Análise de grupos

Utilizando dois cluster como mostrado na figura 27, e os dados na figura 28.

Figura 27: Kmens com 2 clusters



Fonte: Elaboração Própria

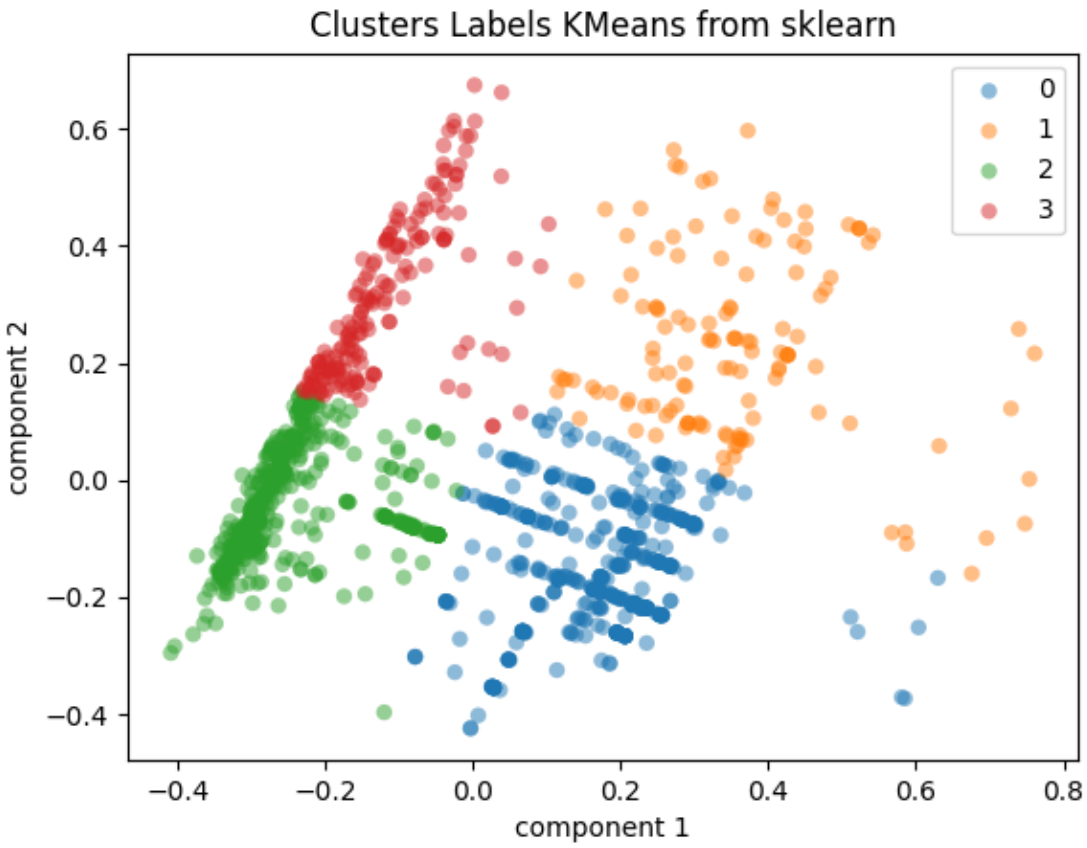
Figura 28: Dados Kmens com 2 clusters

```
Variância expleined por componente:
[0.47745047 0.34777251]
(1197, 16)
(1197, 2)

inertia_ =  62.37661516195151
Para n_clusters = 2, silhouette_score é 0.4645040442866176)
```

Fonte: Elaboração Própria

Utilizando quatro cluster como mostrado na figura 29, e os dados na figura 30.

Figura 29: Kmens com 4 clusters



Fonte: Elaboração Própria

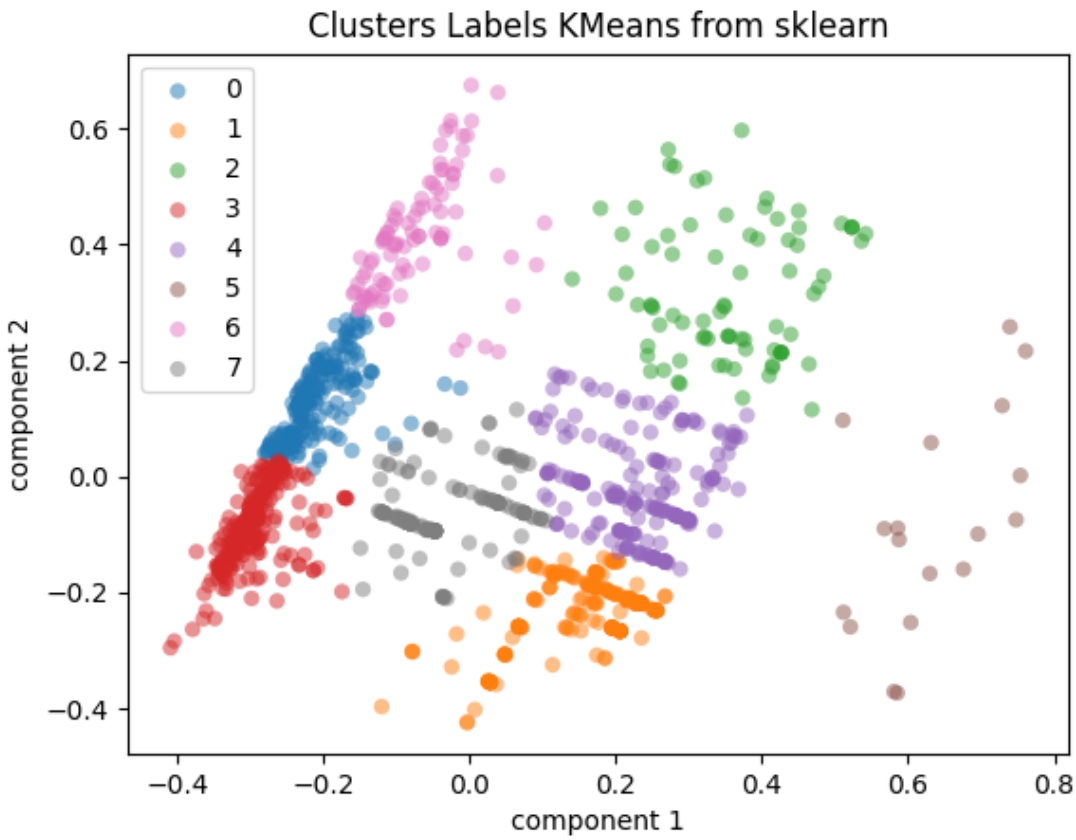
Figura 30: Dados Kmens com 4 clusters

```
Variância expleined por componente:
[0.47745047 0.34777251]
(1197, 16)
(1197, 2)

inertia_ =  27.524208226615464
Para n_clusters = 4, silhouette_score é 0.48520282885618327)
```

Fonte: Elaboração Própria

Utilizando oito cluster como mostrado na figura 31 e os dados na figura 32.

Figura 31: Kmens com 8 clusters



Fonte: Elaboração Própria

Figura 32: Dados Kmens com 8 clusters

```
Variância expleined por componente:
[0.47745047 0.34777251]
(1197, 16)
(1197, 2)

inertia_ =  12.19547102531525
Para n_clusters = 8, silhouette_score é 0.4733988529933448)
```

Fonte: Elaboração Própria

9. Regressão

Regressão é a técnica mais antiga e bem conhecida estatística de que a comunidade de mineração de dados utiliza. Basicamente, a regressão tem um conjunto de dados numéricos e desenvolve uma fórmula matemática que se ajusta aos dados. Quando você está pronto para usar os resultados para prever o comportamento futuro, você simplesmente pega seus novos dados, conecte-o a fórmula desenvolvida e você tem uma previsão! A principal limitação desta técnica é que ela só funciona bem com contínua de dados quantitativos

O R^2 determina se o modelo ajusta bem os dados. Quanto mais alto o valor de R^2 melhor o modelo ajusta seus dados.

9.1. Linear

Regressão linear é o processo de traçar uma reta através dos dados em um diagrama de dispersão. A reta resume esses dados, o que é útil quando fazemos previsões.

Figura 33: Linear

	team	targeted_productivity	smv	wip	over_time	incentive	no_of_workers	actual_productivity
0	8	0.80	26.16	1108.0	7080	98	59.0	0.940725
1	1	0.75	3.94	6.0	960	0	8.0	0.886500
2	11	0.80	11.41	968.0	3660	50	30.5	0.800570
3	12	0.80	11.41	968.0	3660	50	30.5	0.800570
4	6	0.80	25.90	1170.0	1920	50	56.0	0.800382

R2 no set de treino: 0.27
R2 no set de teste: 0.21
Erro absoluto no set de treino: 0.11

Fonte: Elaboração Própria

9.2. SVM (máquina de vetores de suporte)

SVM padrão toma como entrada um conjunto de dados e prediz, para cada entrada dada, qual de duas possíveis classes a entrada faz parte, o que faz do SVM um classificador linear binário não probabilístico. Dados um conjunto de exemplos de treinamento, cada um marcado como pertencente a uma de duas categorias, um algoritmo de treinamento do SVM constrói um modelo que atribui novos exemplos a uma categoria ou outra. Um modelo SVM é uma representação de exemplos como pontos no espaço,

mapeados de maneira que os exemplos de cada categoria sejam divididos por um espaço claro que seja tão amplo quanto possível.

Figura 34: SVM

	team	targeted_productivity	smv	wip	over_time	incentive	no_of_workers	actual_productivity
0	8	0.80	26.16	1108.0	7080	98	59.0	0.940725
1	1	0.75	3.94	6.0	960	0	8.0	0.886500
2	11	0.80	11.41	968.0	3660	50	30.5	0.800570
3	12	0.80	11.41	968.0	3660	50	30.5	0.800570
4	6	0.80	25.90	1170.0	1920	50	56.0	0.800382

R2 no set de treino: 0.05
R2 no set de teste: 0.01
Erro absoluto no set de treino: 0.13

10. Fonte: Elaboração Própria

10.1. Bayesian

No ponto de vista bayesiano, formulamos regressão linear usando distribuições de probabilidade em vez de estimativas pontuais. A resposta, y, não é estimada como um valor único, mas é assumida como sendo derivada de uma distribuição de probabilidade.

As técnicas de regressão bayesiana podem ser usadas para incluir parâmetros de regularização no procedimento de estimativa: o parâmetro de regularização não é definido em sentido difícil, mas sintonizado com os dados em mãos.

Figura 35: Bayesian

	team	targeted_productivity	smv	wip	over_time	incentive	no_of_workers	actual_productivity
0	8	0.80	26.16	1108.0	7080	98	59.0	0.940725
1	1	0.75	3.94	6.0	960	0	8.0	0.886500
2	11	0.80	11.41	968.0	3660	50	30.5	0.800570
3	12	0.80	11.41	968.0	3660	50	30.5	0.800570
4	6	0.80	25.90	1170.0	1920	50	56.0	0.800382

R2 no set de treino: 0.26
R2 no set de teste: 0.21
Erro absoluto no set de treino: 0.11

Fonte: Elaboração Própria

10.2. Nearest Neighbors

K-Nearest Neighbor é um método de predição que utiliza da distância entre a amostra atual e seus k vizinhos mais próximos no conjunto de treinamento para definir qual será o resultado de sua predição.

A regressão baseada em vizinhos pode ser usada nos casos em que os rótulos de dados são contínuos e não variáveis discretas. O rótulo atribuído a um ponto de consulta é computado com base na média dos rótulos de seus vizinhos mais próximos.

Figura 36: Nearest Neighbors

```
DataMiningSamples-master/4-Regression/NearestNeighborsRegression.py
   team  targeted_productivity    smv     wip  over_time  incentive  no_of_workers  actual_productivity
0     8                   0.80  26.16  1108.0       7080         98           59.0             0.940725
1     1                   0.75   3.94     6.0        960          0            8.0             0.886500
2    11                   0.80  11.41   968.0       3660         50           30.5             0.800570
3    12                   0.80  11.41   968.0       3660         50           30.5             0.800570
4     6                   0.80  25.90  1170.0       1920         50           56.0             0.800382
R2 no set de treino: 0.76
R2 no set de teste: 0.14
Erro absoluto no set de treino: 0.11
```

Fonte: Elaboração Própria

10.3. Neural Network

O Class MLPRegressor implementa um *perceptron* de várias camadas (MLP) que treina usando a retropagação sem função de ativação na camada de saída, que também pode ser visto como usando a função de identidade como função de ativação. Portanto, ele usa o erro quadrado como função de perda, e a saída é um conjunto de valores contínuos.

Valores abaixo foram obtidos após executar algumas vezes o algoritmo, apresentando variação entre os valores como mostrado na figura abaixo.

Figura 37: Neural Network

```
   team  targeted_productivity    smv     wip  over_time  incentive  no_of_workers  actual_productivity
0     8                   0.80  26.16  1108.0       7080         98           59.0             0.940725
1     1                   0.75   3.94     6.0        960          0            8.0             0.886500
2    11                   0.80  11.41   968.0       3660         50           30.5             0.800570
3    12                   0.80  11.41   968.0       3660         50           30.5             0.800570
4     6                   0.80  25.90  1170.0       1920         50           56.0             0.800382
R2 no set de treino: -212.19
R2 no set de teste: -455.07
Erro absoluto no set de treino: 2.01
```

Fonte: Elaboração Própria

Figura 38: Neural Network

```
   team  targeted_productivity    smv     wip  over_time  incentive  no_of_workers  actual_productivity
0     8                   0.80  26.16  1108.0       7080         98           59.0             0.940725
1     1                   0.75   3.94     6.0        960          0            8.0             0.886500
2    11                   0.80  11.41   968.0       3660         50           30.5             0.800570
3    12                   0.80  11.41   968.0       3660         50           30.5             0.800570
4     6                   0.80  25.90  1170.0       1920         50           56.0             0.800382
R2 no set de treino: -3.11
R2 no set de teste: -14.28
Erro absoluto no set de treino: 0.30
```

Fonte: Elaboração Própria

Erro absoluto (R^2) compara o ajuste do modelo escolhido com o de uma linha reta horizontal (a hipótese nula). Se o modelo escolhido se encaixar pior que uma linha

horizontal, então é negativo. É negativo apenas quando o modelo escolhido não segue a tendência dos dados; portanto, se encaixa pior que uma linha horizontal.

10.4. Decision Trees

O nó raiz ou superior da árvore (e há apenas uma raiz) é o nó de decisão que divide o conjunto de dados usando uma variável ou recurso que resulta na melhor métrica de divisão avaliada para cada subconjunto ou classe no conjunto de dados resultante do Dividido. A árvore de decisão aprende dividindo recursivamente o conjunto de dados da raiz em diante (de maneira gananciosa, nó por nó) de acordo com a métrica de divisão em cada nó de decisão. Os nós terminais são alcançados quando a métrica de divisão está em um extremo global.

Figura 39: Decision Trees

```
DataMiningSamples-master/4-Regression/TreeRegression.py
   team  targeted_productivity    smv     wip  over_time  incentive  no_of_workers  actual_productivity
0     8                   0.80  26.16  1108.0       7080         98           59.0             0.940725
1     1                   0.75   3.94     6.0        960          0            8.0             0.886500
2    11                   0.80  11.41   968.0       3660         50           30.5             0.800570
3    12                   0.80  11.41   968.0       3660         50           30.5             0.800570
4     6                   0.80  25.90  1170.0       1920         50           56.0             0.800382
R2 no set de treino: 0.90
R2 no set de teste: 0.12
Erro absoluto no set de treino: 0.10
```

Fonte: Elaboração Própria