Análise dos datasets

Felipe B. Diniz

1 Introdução

Neste PDF encontram-se diversas análises dos datasets `sales.csv` e `comp_prices.csv` (que chamaremos apenas de *comp*), assim como a descrição dos métodos para a predição de quantidades a partir dos preços dados. Associado a este texto há o script para realizar as predições e um notebook com os códigos para as imagens que veremos aqui. Estes arquivos se encontram no meu github, em `hsdjas`. Começamos visualizando algumas entradas iniciais dos dados.

```
1 sales.head()
```

	PROD_ID	DATE_ORDER	QTY_ORDER	REVENUE
0	P6	2015-08-02	1.0	1808.99
1	P6	2015-08-17	1.0	1674.00
2	P6	2015-08-17	1.0	1673.95
3	P6	2015-08-11	1.0	1674.00
4	P6	2015-08-17	1.0	1674.00

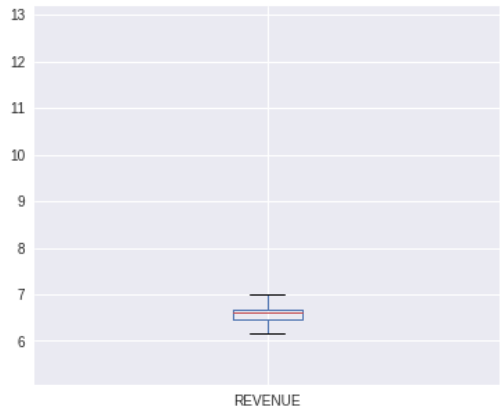
```
1 comp.head()
```

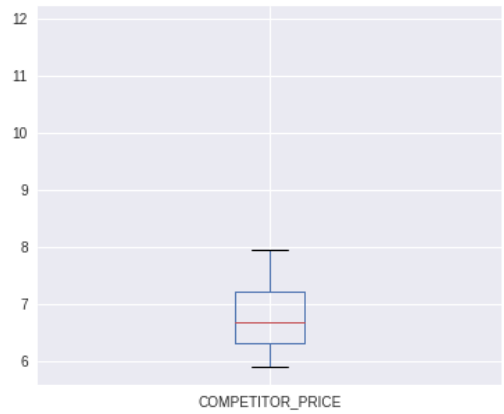
	PROD_ID	DATE_EXTRACTION	COMPETITOR	COMPETITOR_PRICE	PAY_TYPE
0	P6	2015-01-30 08:11:38	C3	1898.00	1
1	P6	2015-01-30 08:11:38	C1	1898.00	1
2	P6	2015-01-30 08:11:38	C6	1999.90	1
3	P6	2015-01-31 20:10:14	C2	1894.88	2
4	P6	2015-01-31 20:10:14	C3	1894.88	2

2 Limpeza de dados

2.1 Dados omissos e remoção de ouliers

Primeiramente, podemos notar que não há dados faltando em ambos os datasets. Então estamos encerrados neste aspecto. Para se ter uma noção melhor dos outliers de algumas colunas escolhidas, usaremosa visualização por boxplot. Estamos fazendo o plot dos dados em escala logarítmica uma vez que há valores muito grandes entre os dados.





Baseado nestes gráficos, propomos retirar os outliers segundo algum critério. Consideramos retirar no máximo até 1% dos dados, de modo a não perder informação demais. Retiramos os elementos de REVENUE que são maiores que 8 e os elementos de COMPETITOR PRICE que são maiores que 9 (ambos na escala logarítmica). A porcentagem dos dados retidos é ainda grande o suficiente para considerarmos plausível a remoção dos outliers.

2.2 Ajuste das colunas

Todas as datas se encontram no mesmo ano, 2015, então podemos omitir o ano e considerá-lo uma informação implícita do dataset. Além disso, consideramos desnecessário o horário da extração dos preços da competição. Algumas colunas se referem ao mesmo tipo de informação, então iremos renomeá-las como indicado abaixo:

sales column name	**comp column name**	**new name**
DATE ORDER	DATE EXTRACTION	DATE
REVENUE	COMPETITOR PRICE	REVENUE
PROD ID	PROD ID	PROD

Os dois tipos de pagamento (coluna PAY TYPE) ocorrem exatamente o mesmo número de vezes. Podemos nos perguntar se essa coincidência vale mais em geral. Mais precisamente, se cada a cada produto o tipo de compra é dividido também igualmente, se a cada competidor o tipo de compra é dividido igualmente, etc. Porque se essa divisão for meio a meio de qualquer ângulo que se olhe, então PAY TYPE acaba não sendo um atributo muito informativo e pode ser descartado. Abaixo mostramos em mais detalhes a distribuição deste atributo.

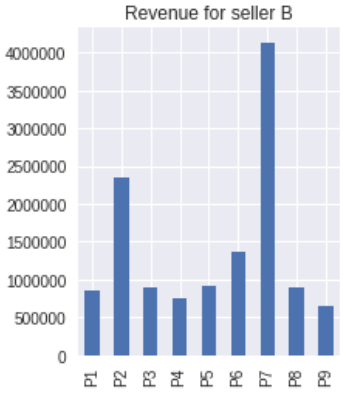
	C1	C2	C3	C4	C5	C6	P1	P2	P3	P4	P5	P6	P7	P8	P9
PAY TYPE 1	4929	4880	4842	3992	2142	4249	1354	4374	2925	843	945	4766	3872	2896	3059
PAY TYPE 2	4923	4874	4837	3987	2139	4244	1351	4371	2918	842	945	4766	3866	2891	3054

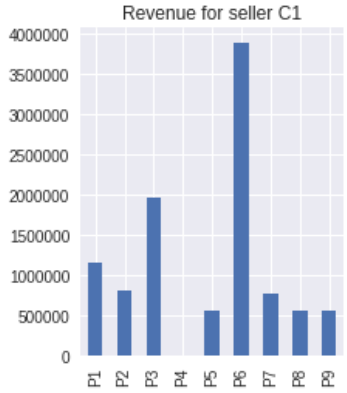
Podemos observar que os tipos de pagamentos se distribuem de maneira muito bem dividida, sendo praticamente meio a meio sempre. Por conta disso iremos descartar esta coluna.

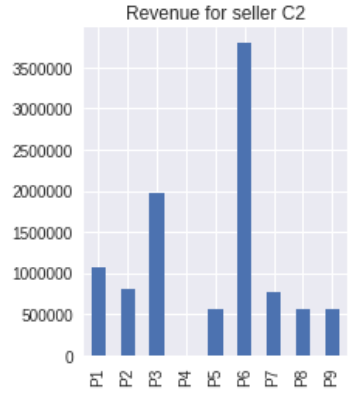
Logo no início do dataset *sales* podemos perceber que há um mesmo produto sendo vendido três vezes no mesmo dia, talvez até mais, mas vamos supor que seja três. O produto em questão é o P6, na data 2015-08-17. Como o nosso interesse é prever a quantidade de vendas por produto em cada momento, o correto seria que este produto aparecesse apenas uma vez, com o valor da coluna QTY ORDER sendo igual a 3. Antes de continuar devemos arrumar isso para todas as possíveis ocorrências. A função para realizar esta tarefa se chama `fix_qty_order` e é a função mais custosa de todo o script.

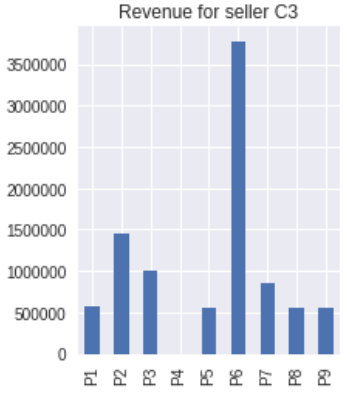
3 Análise de venda dos produtos

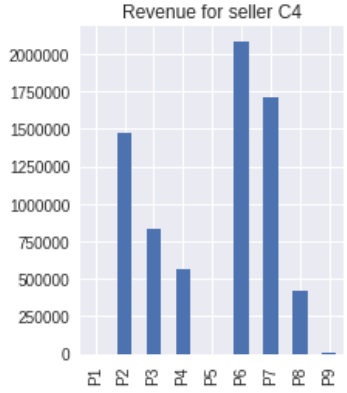
Nesta parte, vamos ver quais produtos interessam mais a cada uma das empresas. O "interesse" aqui é medido pela coluna REVENUE, que nos diz o rendimento que a empresa obteve em cima do produto. Para isto, nos utilizamos da visualização de gráficos de barras.

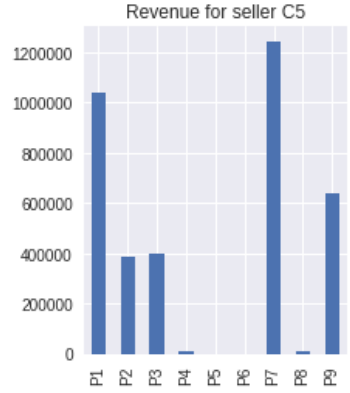


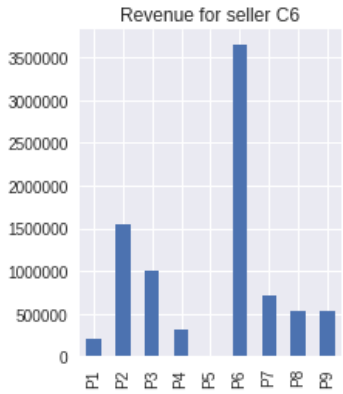












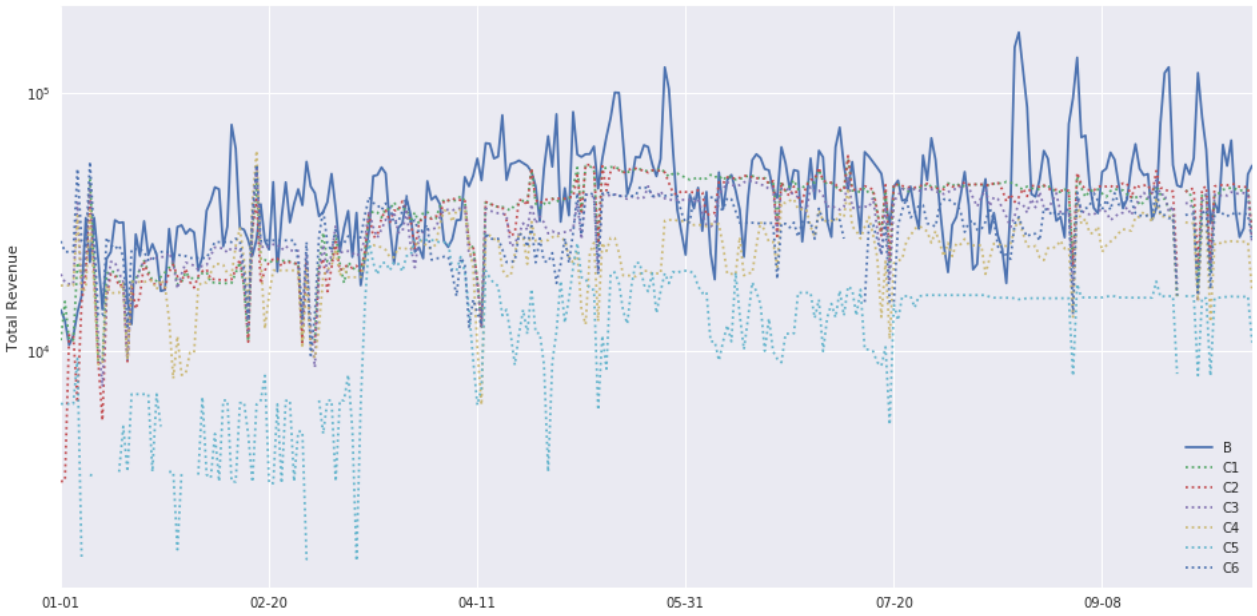
De maneira geral, os produtos de mais interesse são P6 e P7, com a exceção de P1 que é de bastante interesse para C5. A empresa B (a empresa do dataset *sales*) tem claro foco no produto P7 e praticamente nenhum interesse em P1 e P6. Pelo que podemos ver dos histogramas, todos os competidores, exceto C5, seguem a estratégia de vender mais o produto P6. O competidor C5 parece tentar vender mais P7, o produto mais vendido por B, e P1, um dos produtos menos vendidos por B. Note que o competidor C4 também vende bastante do P7. Podemos ver dois tipos de estratégias aqui:

1. Se concentrar em um único produto em que B possui poucos consumidores.
2. Se concentrar em dois produtos, um em que B possui poucos consumidores e outro em que B possui muitos consumidores.

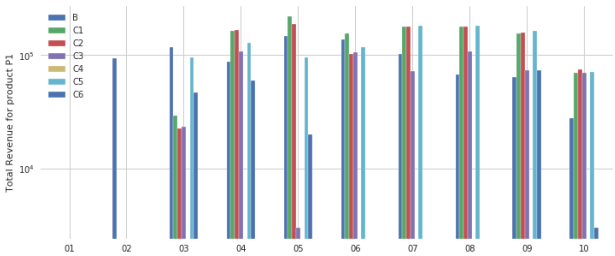
3.1 Séries temporais

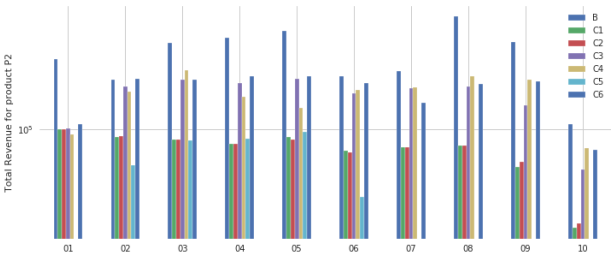
Vimos os produtos que interessam mais a cada empresa, e a partir disso traçamos uma ideia geral da estratégia escolhida por cada empresa. Pelos valores mostrados acima, é possível notar que a empresa B lucra muito mais que as outras empresas. Isto pode não ser necessariamente um quadro exato da realidade pois os dados extraídos da competição representam apenas uma parcela de duas vendas.

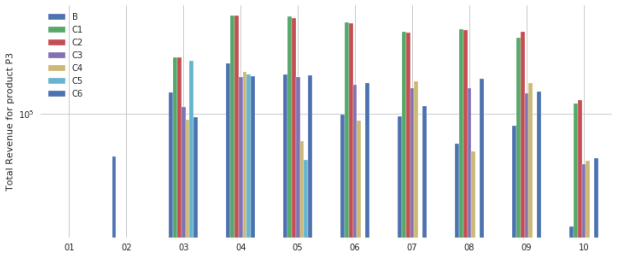
Outro dado interessante para ser extraído é o rendimento total da empresa em cada momento (a soma das vendas de todos os produtos em um dado dia).

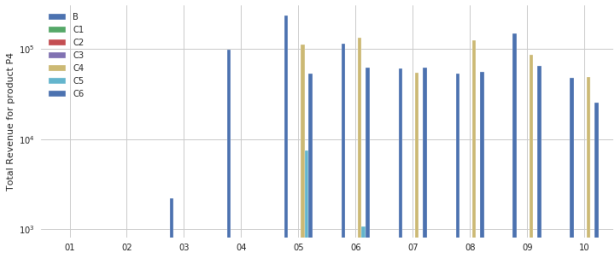


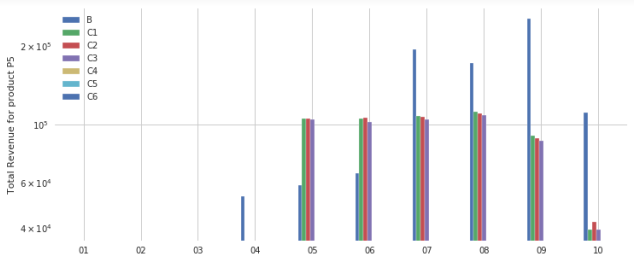
Como há muita flutuação nos valores, talvez seja mais interessante agrupá-los mensalmente. Dessa maneira esperamos poder ver mais facilmente os padrões. Com uma quantidade menor de datas, é possível fazer gráfico de barras, o que é mais claro neste caso.

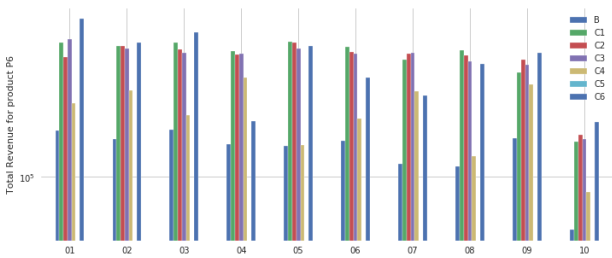


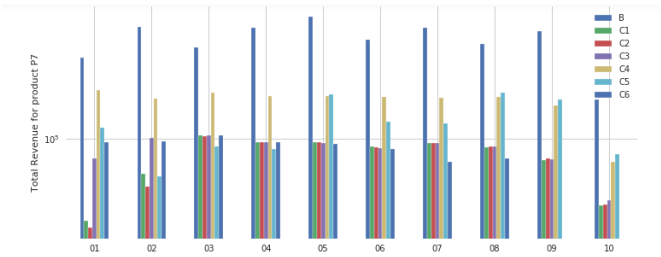


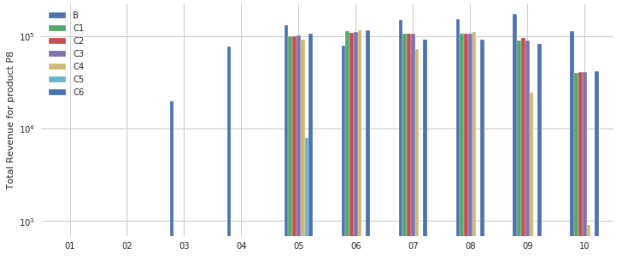


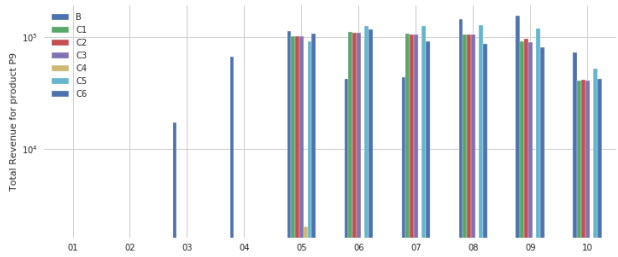












4 Clusterização com K-means

Já temos uma ideia geral de como é o dataset e como as features se relacionam entre si. Agora devemos preparar o dataset para o treinamento. Iremos proceder da seguinte maneira:

- As colunas serão PROD, DATE, QTY ORDER, REVENUE B, REVENUE C (consideramos toda a competição como uma coisa só) e MONTH

- Usaremos as datas no formato mês/dia. Podemos ter no máximo 9 linhas com a mesma data, pois cada instância de uma data será a média dos rendimentos de um produto específico vendido naquele dia.

- O dataset será normalizado.

4.1 Junção dos datasets

Agora que temos tudo mais organizado, fazemos um join nas linhas em que há coincidência de dia e produto. Após isso, haverá perda de linhas do dataset *sales* quando tivermos algum dia e produto sem match no outro dataset. Neste caso, para evitar perder informação, usamos a informação da linha de *sales* como se fosse a linha da competição. Isto significa que nestes casos usaremos apenas a informação de sales para o aprendizado, mas ainda assim é melhor do que descartar a informação. No fim das contas, teremos 2697 inputs para treinar o programa. Esta é uma redução considerável, pois no início tínhamos cerca de 300000 inputs para lidar só em *sales*.

```
RangeIndex: 2697 entries, 0 to 2696
Data columns (total 5 columns):
PROD         2697 non-null object
DATE         2697 non-null object
QTY_ORDER    2697 non-null float64
REVENUE_B    2697 non-null float64
REVENUE_C    2697 non-null float64
dtypes: float64(3), object(2)
memory usage: 105.4+ KB
```

4.2 Normalização dos dados

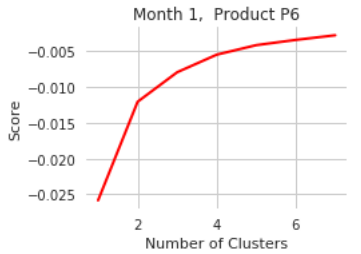
Além de converter todas as features em números, necessitamos arrumar a escala destes números, pois a coluna REVENUE possui valores muito mais altos que todas as outras colunas. Se não arrumarmos isso, a clusterização será enviesada em favor dos valores das colunas (pois o dataset será praticamente unidimensional), com as outras features tendo papel pouco relevante no processo. Existem diversas maneiras de normalizar as features. Aqui nós usaremos a *normalização pela média*, dada pela fórmula abaixo.

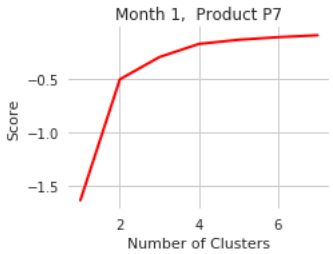
$$x_{new} = \frac{x - E[x]}{\max(x) - \min(x)}$$

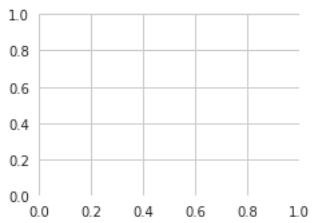
$E[x]$ é a média dos valores da coluna de x, $\max(x)$ é o máximo da coluna e $\min(x)$ o mínimo da coluna. Faremos este procedimento para todas as colunas.

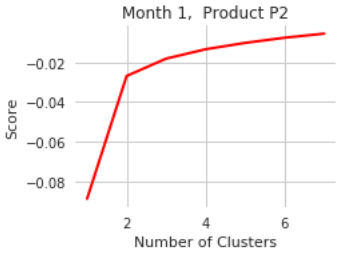
4.3 Treinamento

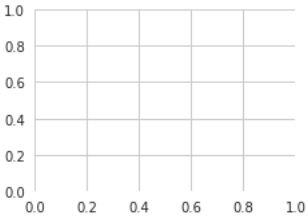
Para cada par (mês, produto) teremos um conjunto de clusters. Começamos agrupando os dados desta maneira. Para determinar o número de clusters, usamos o método *elbow* (mais detalhes em `https://en.wikipedia.org/wiki/Elbow_method_(clustering)`). Mostramos os gráficos dos resultados do treinamento abaixo.

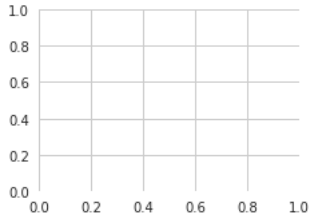

Month 1, Product P6
Score
Number of Clusters

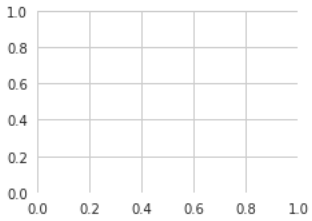

Month 1, Product P7
Score
Number of Clusters

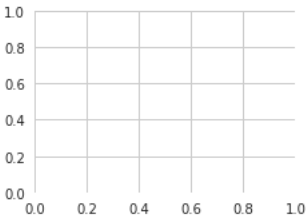


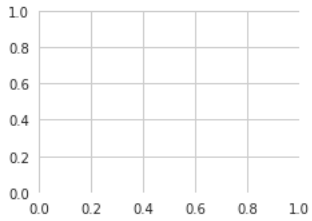

Month 1, Product P2
Score
Number of Clusters

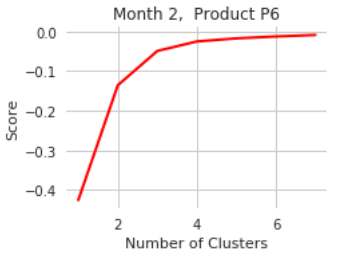


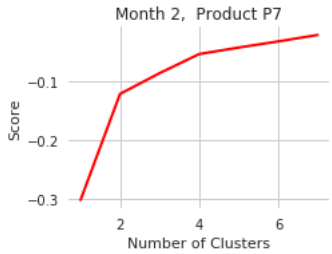


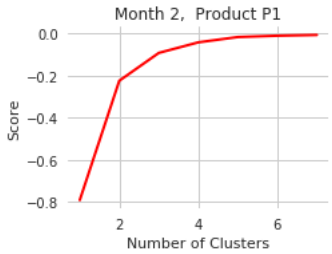


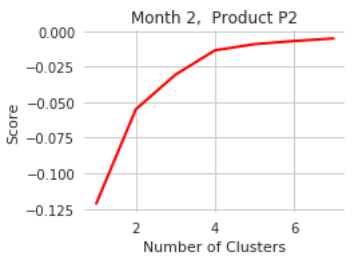


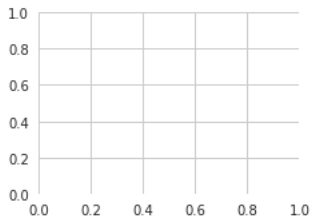


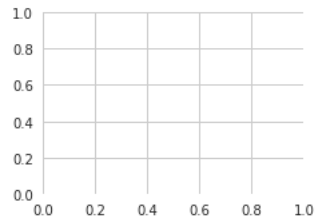

Month 2, Product P6
Score
Number of Clusters

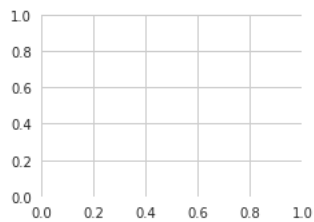

Month 2, Product P7
Score
Number of Clusters

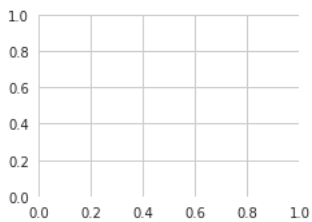

Month 2, Product P1
Score
Number of Clusters

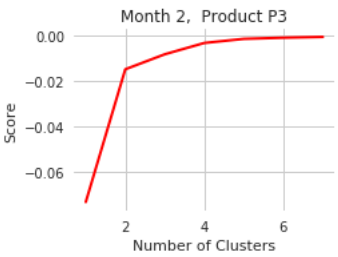

Month 2, Product P2
Score
Number of Clusters

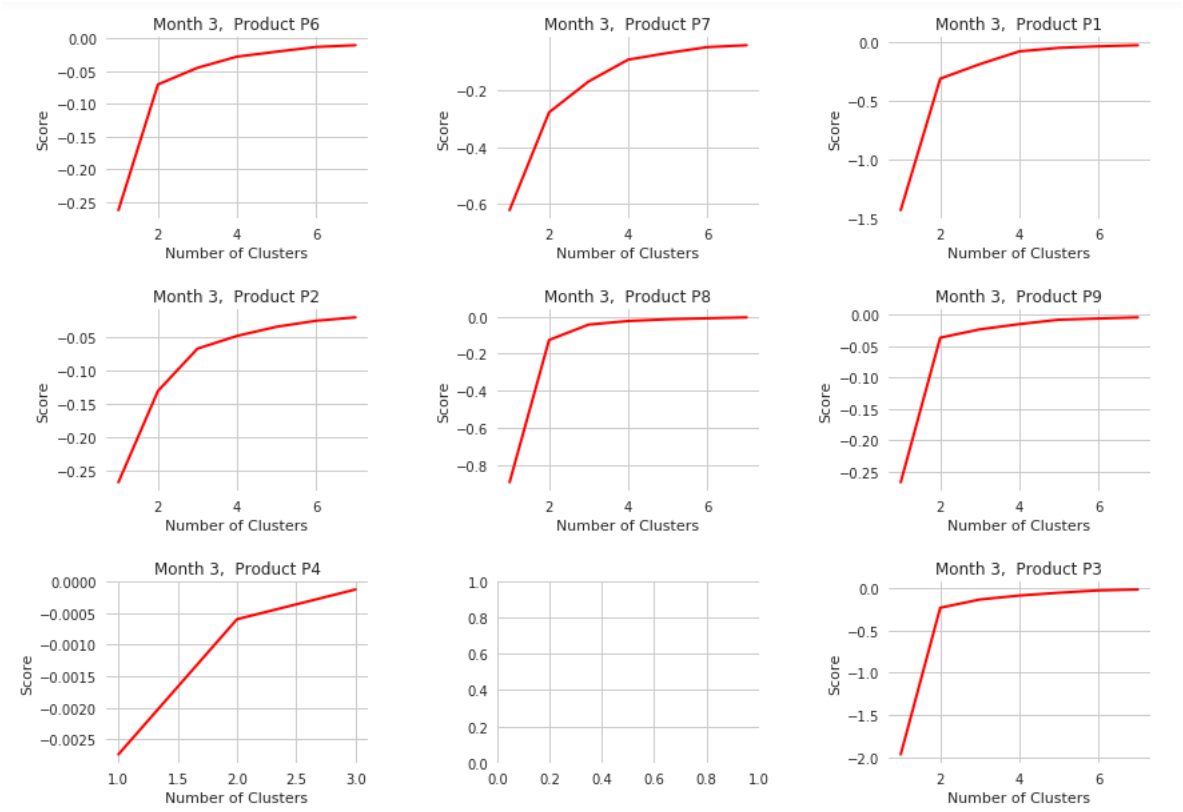


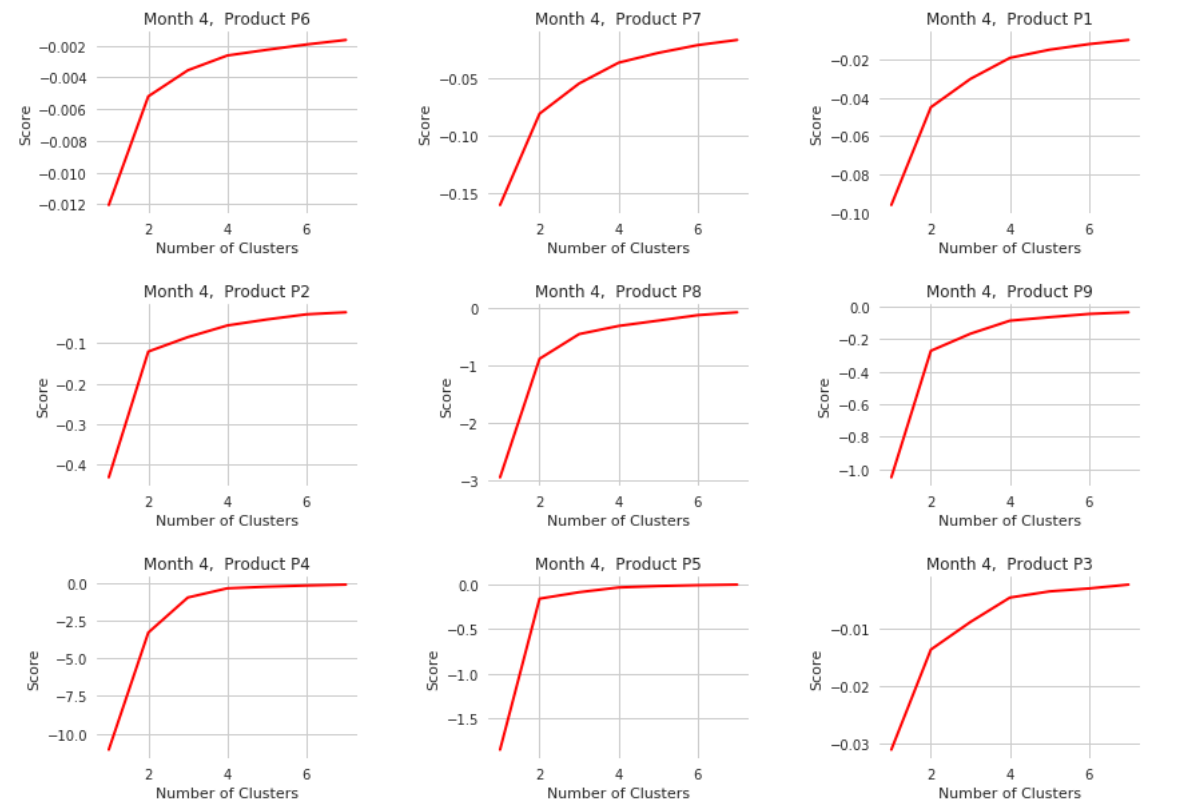


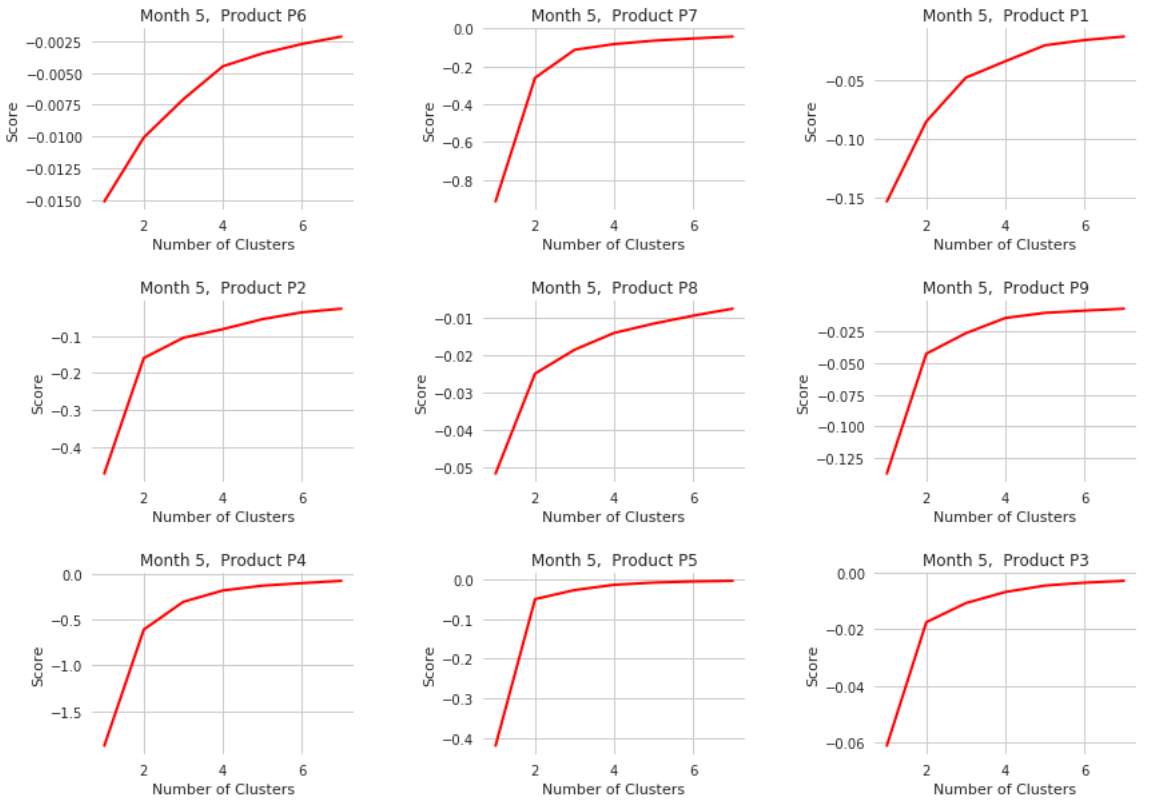


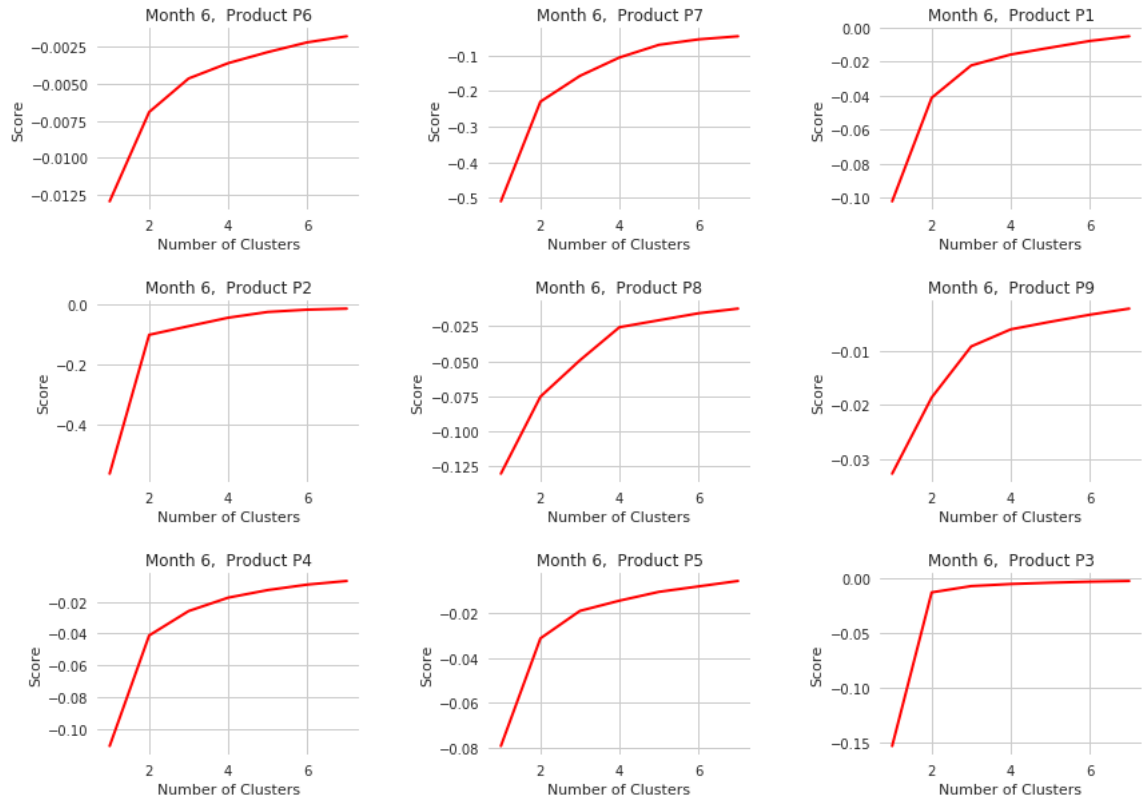



Month 2, Product P3
Score
Number of Clusters

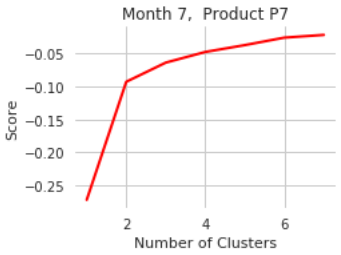

Month 3, Product P6
Month 3, Product P7
Month 3, Product P1
Month 3, Product P2
Month 3, Product P8
Month 3, Product P9
Month 3, Product P4
Month 3, Product P3
Score
Number of Clusters

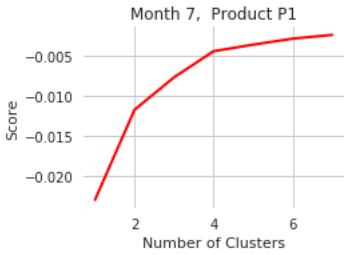

Month 4, Product P6
Month 4, Product P7
Month 4, Product P1
Month 4, Product P2
Month 4, Product P8
Month 4, Product P9
Month 4, Product P4
Month 4, Product P5
Month 4, Product P3
Score
Number of Clusters


Month 5, Product P6
Month 5, Product P7
Month 5, Product P1
Month 5, Product P2
Month 5, Product P8
Month 5, Product P9
Month 5, Product P4
Month 5, Product P5
Month 5, Product P3
Score
Number of Clusters

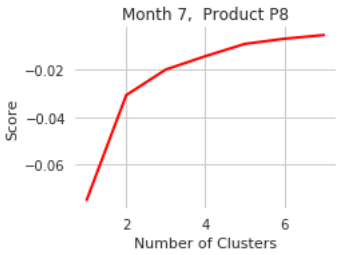

Month 6, Product P6
Month 6, Product P7
Month 6, Product P1
Month 6, Product P2
Month 6, Product P8
Month 6, Product P9
Month 6, Product P4
Month 6, Product P5
Month 6, Product P3
Score
Number of Clusters

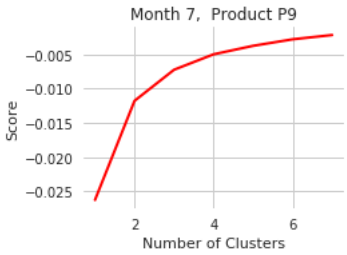

Month 7, Product P6
Score
Number of Clusters

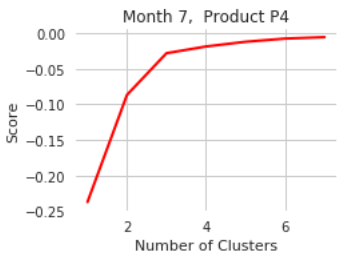

Month 7, Product P7
Score
Number of Clusters


Month 7, Product P1
Score
Number of Clusters

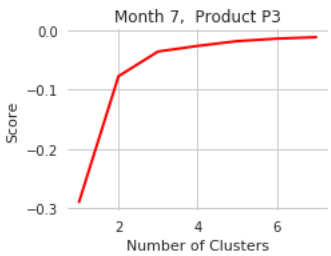

Month 7, Product P2
Score
Number of Clusters


Month 7, Product P8
Score
Number of Clusters

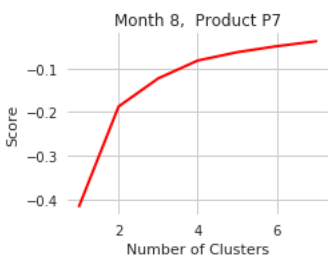

Month 7, Product P9
Score
Number of Clusters


Month 7, Product P4
Score
Number of Clusters

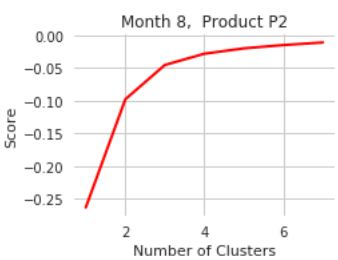

Month 7, Product P5
Score
Number of Clusters

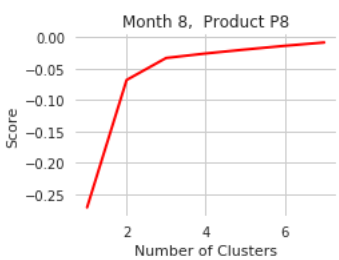

Month 7, Product P3
Score
Number of Clusters

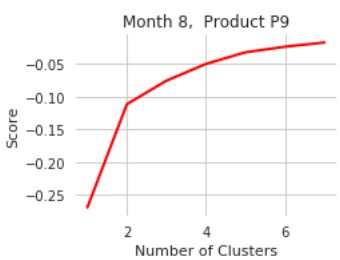

Month 8, Product P6
Score
Number of Clusters

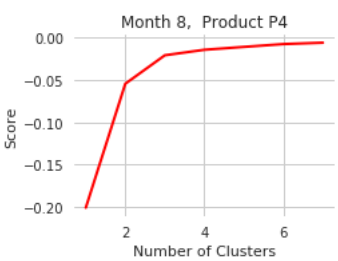

Month 8, Product P7
Score
Number of Clusters


Month 8, Product P1
Score
Number of Clusters


Month 8, Product P2
Score
Number of Clusters


Month 8, Product P8
Score
Number of Clusters

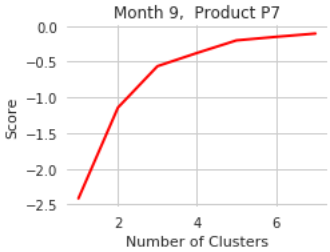

Month 8, Product P9
Score
Number of Clusters


Month 8, Product P4
Score
Number of Clusters

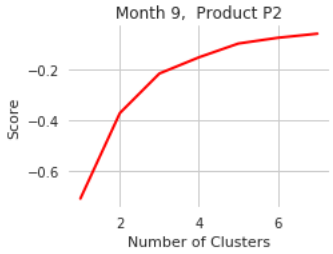

Month 8, Product P5
Score
Number of Clusters

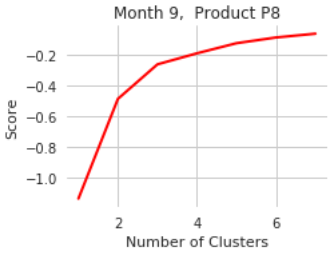

Month 8, Product P3
Score
Number of Clusters

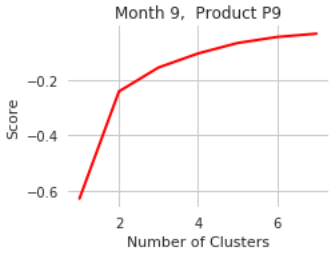

Month 9, Product P6
Score
−0.02
−0.04
−0.06
−0.08
2
4
6
Number of Clusters

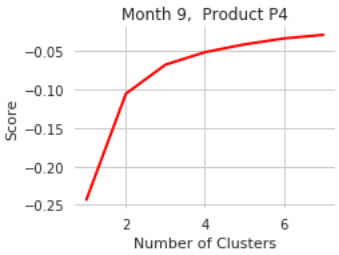

Month 9, Product P7
Score
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
2
4
6
Number of Clusters

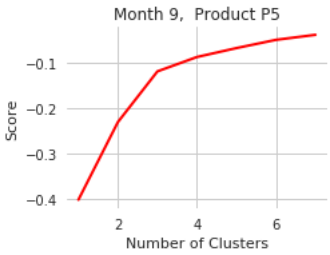

Month 9, Product P1
Score
−0.01
−0.02
−0.03
−0.04
2
4
6
Number of Clusters


Month 9, Product P2
Score
−0.2
−0.4
−0.6
2
4
6
Number of Clusters


Month 9, Product P8
Score
−0.2
−0.4
−0.6
−0.8
−1.0
2
4
6
Number of Clusters

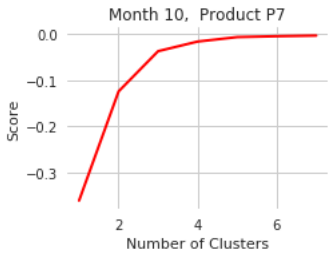

Month 9, Product P9
Score
−0.2
−0.4
−0.6
2
4
6
Number of Clusters

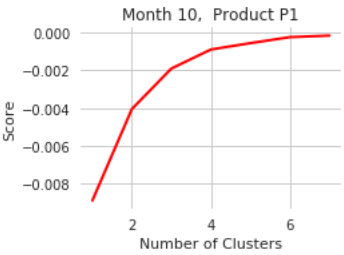

Month 9, Product P4
Score
−0.05
−0.10
−0.15
−0.20
−0.25
2
4
6
Number of Clusters


Month 9, Product P5
Score
−0.1
−0.2
−0.3
−0.4
2
4
6
Number of Clusters

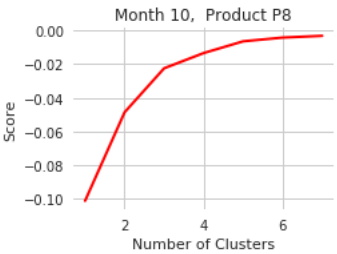

Month 9, Product P3
Score
−0.01
−0.02
−0.03
−0.04
2
4
6
Number of Clusters

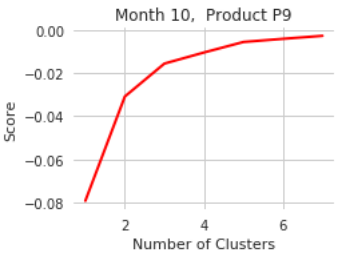

Month 10, Product P6
Score
0.00
−0.01
−0.02
−0.03
−0.04
2
4
6
Number of Clusters


Month 10, Product P7
Score
0.0
−0.1
−0.2
−0.3
2
4
6
Number of Clusters


Month 10, Product P1
Score
0.000
−0.002
−0.004
−0.006
−0.008
2
4
6
Number of Clusters


Month 10, Product P2
Score
0.00
−0.02
−0.04
−0.06
−0.08
2
4
6
Number of Clusters


Month 10, Product P8
Score
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
2
4
6
Number of Clusters


Month 10, Product P9
Score
0.00
−0.02
−0.04
−0.06
−0.08
2
4
6
Number of Clusters


Month 10, Product P4
Score
0.000
−0.005
−0.010
−0.015
−0.020
2
4
6
Number of Clusters

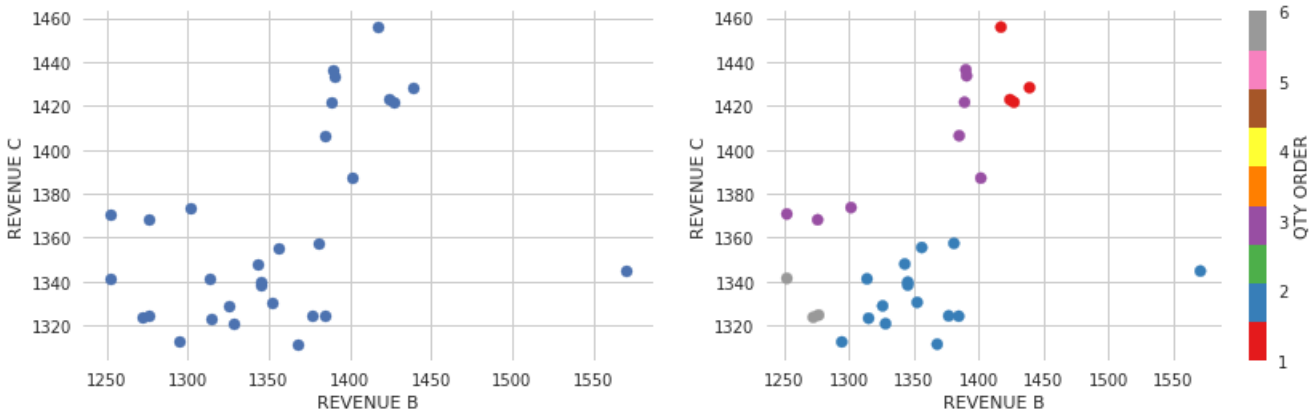

Month 10, Product P5
Score
0.00
−0.02
−0.04
−0.06
2
4
6
Number of Clusters


Month 10, Product P3
Score
0.00
−0.02
−0.04
2
4
6
Number of Clusters

4.4 Predição

Dado o mês, o produto e os dois preços, verificamos qual o centróide mais perto da coordenadas dos preços. Este centróide está relacionado a uma certa quantidade que foi obtida durante o treinamento. Esta será a quantidade que usaremos para a predição.

Apenas para efeito de ilustração, vamos olhar o caso do mês 4 e produto P3. Lembre que cada amostra é um vetor tridimensional com as informações [QTY ORDER, REVENUE B, REVENUE C]. Iremos fazer um scatter plot no plano, onde o eixo x é REVENUE B e o eixo y é REVENUE C. A quantidade será monitorada através da cor dos pontos, que foram obtidos durante o treinamento.



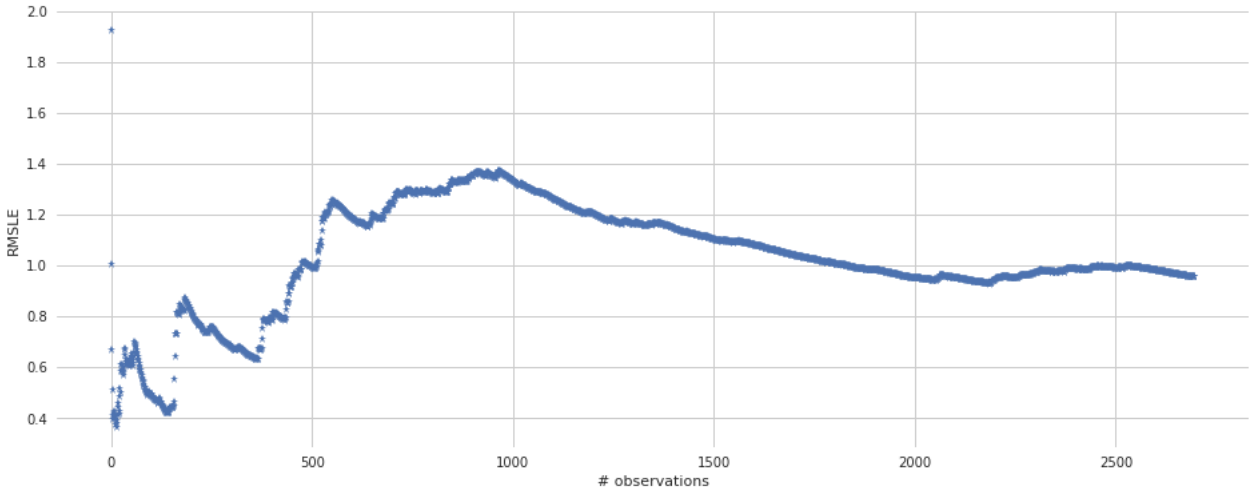
4.5 Acurácia do modelo

Agora que vimos que o algoritmo está separando bem os clusters, vamos como está a acurácia dele. Para isso, faremos previsões para todos os dados do dataset original, após o tratamento.

Sejam $y^{(1)}, \ldots, y^{(n)}$ aos targets originais do dataset, onde $y^{(i)}$ é o elemento da linha i e coluna QTY ORDER, e sejam $\tilde{y}^{(1)}, \ldots, \tilde{y}^{(n)}$ as previsões dadas pelo nosso modelo. Uma maneira de se medir a acurácia do nosso modelo é usando a métrica RMLSE (Root Mean Squared Logarithmic Error). Esta métrica geralmente é usada quando não queremos penalizar grandes diferenças nos valores previstos e os reais quando os valores previstos e verdadeiros são grandes. Ela é definida por

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log(1+\tilde{y}^{(i)}) - \log(1+y^{(i)})\right)^2}.$$

A grosso modo, esta métrica nos dá um fator de proporcionalidade entre o valor previsto e o real. Quanto maior for o valor, pior é a previsão, e quanto mais perto de zero, melhor é a previsão. Abaixo mostramos o plot da evolução do RMSLE. Podemos ver que o modelo tende a melhorar conforme mais dados são inseridos nele.


RMSLE
observations
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
500
1000
1500
2000
2500