Trabalho 1

INF-0615 – Aprendizado Supervisionado I

Prof. Anderson Rocha

IDENTIFICAÇÃO

Membros do grupo:

- Daniele Montenegro da Silva Barros
- Rodrigo Dantas da Silva
- Thiago Bruschi Martins

QUESTÃO 1: INSPEÇÃO DOS DADOS

Iniciamos a inspeção dos dados fazendo um comparativo entre os dados de treino e de validação. Comparamos o número de linhas para verificar a proporção da divisão dos dados, e verificamos se há dados faltantes ou repetidos entre os dois conjuntos com a função *merge*. O resultado destes comandos é apresentado na figura a seguir.

```
> print(paste('Numero de observacoes para treino:', nrow(train_set)))
[1] "Numero de observacoes para treino: 244582"
> print(paste('Numero de observacoes para validacao:', nrow(val_set)))
[1] "Numero de observacoes para validacao: 61147"
> merge(train_set, val_set)
 [1] No      year    month   day     hour    PM2.5   PM10    SO2     NO2     O3      TEMP    PRES    DEWP    RAIN    wd
 WSPM    target
<0 linhas> (ou row.names de comprimento 0)
```

Figura 1. Número de linhas de cada conjunto

Assim, podemos ver que a proporção entre os dados de treino e teste é de 61147 / (244582+61147) = 0.20, ou seja, 20% dos dados são usados para validação e não há repetição entre os dois conjuntos. A seguir utilizamos a função summary para ter uma visão geral do conjunto de treino.

```
       No             year          month             day              hour          PM2.5             PM10       
 Min.   :    1   Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   : 0.00   Min.   :  2.00   Min.   :  2.0  
 1st Qu.: 9589   1st Qu.:2014   1st Qu.: 3.000   1st Qu.: 8.00   1st Qu.: 6.00   1st Qu.: 20.00   1st Qu.: 36.0  
 Median :18057   Median :2015   Median : 6.000   Median :16.00   Median :12.00   Median : 55.00   Median : 82.0  
 Mean   :17935   Mean   :2015   Mean   : 6.469   Mean   :15.73   Mean   :11.57   Mean   : 79.42   Mean   :104.5  
 3rd Qu.:26489   3rd Qu.:2016   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:18.00   3rd Qu.:111.00   3rd Qu.:145.0  
 Max.   :35064   Max.   :2017   Max.   :12.000   Max.   :31.00   Max.   :23.00   Max.   :844.00   Max.   :999.0  

      SO2               NO2                O3               TEMP             PRES             DEWP             RAIN          
 Min.   :  0.2856   Min.   :  2.00   Min.   :   0.2142   Min.   :-19.90   Min.   : 982.7   Min.   :-36.0   Min.   : 0.00000  
 1st Qu.:  2.0000   1st Qu.: 23.00   1st Qu.:  10.4958   1st Qu.:  3.10   1st Qu.:1002.4   1st Qu.: -9.0   1st Qu.: 0.00000  
 Median :  7.0000   Median : 43.00   Median :  45.0000   Median : 14.40   Median :1010.4   Median :  2.9   Median : 0.00000  
 Mean   : 15.6896   Mean   : 50.63   Mean   :  57.2427   Mean   : 13.51   Mean   :1010.8   Mean   :  2.4   Mean   : 0.06359  
 3rd Qu.: 19.0000   3rd Qu.: 71.00   3rd Qu.:  82.0000   3rd Qu.: 23.20   3rd Qu.:1019.0   3rd Qu.: 15.0   3rd Qu.: 0.00000  
 Max.   :500.0000   Max.   :290.00   Max.   :1071.0000   Max.   : 41.60   Max.   :1042.8   Max.   : 29.1   Max.   :52.10000  

      wd              WSPM           target       
 NE      : 25096   Min.   : 0.000   Min.   :  100  
 ENE     : 20029   1st Qu.: 0.900   1st Qu.:  500  
 NW      : 19211   Median : 1.400   Median :  900  
 N       : 17987   Mean   : 1.738   Mean   : 1229  
 E       : 17371   3rd Qu.: 2.200   3rd Qu.: 1500  
 SW      : 16929   Max.   :12.900   Max.   :10000  
 (Other):127959                                    
```

Figura 2. Resumo (summary) dos dados de treinamento

Na Figura 2 podemos verificar que o conjunto de features categóricas a princípio consiste apenas na coluna wd, que indica a direção do vento no momento da leitura dos dados. No entanto, decidimos tratar o mês, o dia e a hora como categóricos também, pois acreditamos que o valor dessas variáveis não devem ser usados para se calcular distância entre si. Ao utilizá-los como categóricos e aplicarmos o processo de *One Hot Encode*, cada possível valor dessas variáveis se tornam uma nova dimensão do nosso conjunto de dados. Desta forma, se tornam vetores ortogonais entre si no nosso espaço amostral, impedindo que se calcule a distância entre os dias e meses, por exemplo. No caso do ano deixamos ele como variável numérica pois acreditamos que ele dá uma noção de tempo muito mais precisa, visto que os anos não se repetem. E por isso pode ser usado no cálculo de distância entre uma observação e outra.

Antes de prosseguir, verificamos que todas as features estão contidas nas anotações apresentadas no enunciado do exercício. Caso houvesse alguma feature sem anotação, teríamos que nos atentar ao tipo da variável, pois o R poderia errar o tipo da variável na hora de importar os features com outliers ou dados faltantes. Depois disso, o tratamento prosseguiria igualmente com a normalização das features numéricas ou processo de *One-Hot Encode* para features categóricas.

QUESTÃO 2: NORMALIZAÇÃO

Nosso grupo optou por utilizar a Z-Norma como técnica de normalização para as variáveis numéricas. Os cuidados tomados aqui foram em relação às variáveis categóricas e a coluna *target*, pois como visto em aula, nenhuma delas deve ser normalizada. Além disso, vale lembrar que a normalização dos dados de validação e dados de teste deve ser (e foi) feita com a média e desvio padrão dos dados de treino.

QUESTÃO 3: BASELINE

O Baseline, que serve como chute inicial e referência para os modelos futuros, foi criado utilizando todas as features disponíveis, incluindo as features categóricas, e sem combinação alguma entre elas. Devido a grande quantidade de valores para dia, mês e hora, chegamos a 96 features no total, o resultado de treino, validação e teste é apresentado na tabela a seguir.

Tabela 1. Resultados do modelo Baseline

Modelo	MAE Treino	MAE Validação	MAE Teste
Baseline	365.05	367.14	367.14

QUESTÃO 4 COMBINAÇÃO DE FEATURES EXISTENTES

Antes de combinarmos as features existentes, nós optamos por analisar a matriz de correlação em busca de insights de como seria a melhor maneira de combinar estas features. A matriz de correlação gerada pela função cor() foi exportada para uma planilha onde pudemos formatá-la condicionalmente, de forma que as maiores correlações (em módulo) fossem destacadas.

Tabela 2: Tabela de correlação entre as principais variáveis numéricas no conjunto de treinamento

	PM2.5	PM10	SO2	NO2	O3	TEMP	PRES	DEWP	RAIN	WSPM
PM2.5	1.00	0.88	0.48	0.67	-0.15	-0.13	0.01	0.12	-0.02	-0.28
PM10	0.88	1.00	0.47	0.65	-0.11	-0.09	-0.02	0.07	-0.03	-0.19
SO2	0.48	0.47	1.00	0.50	-0.17	-0.32	0.22	-0.27	-0.04	-0.11
NO2	0.67	0.65	0.50	1.00	-0.48	-0.28	0.17	-0.03	-0.04	-0.40
O3	-0.15	-0.11	-0.17	-0.48	1.00	0.60	-0.45	0.31	0.02	0.30
TEMP	-0.13	-0.09	-0.32	-0.28	0.60	1.00	-0.81	0.82	0.04	0.03
PRES	0.01	-0.02	0.22	0.17	-0.45	-0.81	1.00	-0.75	-0.06	0.07
DEWP	0.12	0.07	-0.27	-0.03	0.31	0.82	-0.75	1.00	0.09	-0.30
RAIN	-0.02	-0.03	-0.04	-0.04	0.02	0.04	-0.06	0.09	1.00	0.02
WSPM	-0.28	-0.19	-0.11	-0.40	0.30	0.03	0.07	-0.30	0.02	1.00
target	0.79	0.71	0.54	0.71	-0.31	-0.32	0.18	-0.05	-0.01	-0.30

Ao observarmos a Tabela 2, fica fácil encontrar as principais correlações entre as variáveis, que estão destacadas com cores mais quentes. As cores mais frias indicam uma correlação mais baixa. A partir desses resultados nós pensamos em duas maneiras de combinar as features:

1. Começar com um número reduzido de features e ir aumentando esse número gradativamente, enquanto aumentamos o número de combinações entre as features ao mesmo tempo
2. Testar com um número fixo de features mas ir aumentando o número de combinações

Em ambos os casos começamos com uma complexidade mais baixa e fomos aumentando gradativamente. No modelo número 1, optamos por começar com as features que apresentavam maior correlação com o target e menor correlação entre si. E por último deixamos as features com uma correlação maior com alguma outra feature. O desempenho no treino e na validação das duas abordagens pode ser visto no gráfico a seguir. Para facilitar a identificação dos modelos, chamamos o modelo em que vamos adicionando features de Modelo Crescente e o modelo em que o número de features é fixo chamamos de Modelo Combinatório. Através do gráfico da Figura 3, podemos observar que os dois modelos começam a dar overfitting em momentos distintos. O Modelo Combinatório começa a dar indícios de overfitting no nível de complexidade 4, enquanto o Modelo Crescente faz isso apenas no nível de complexidade 9. Como devemos escolher um modelo para testar no conjunto de teste baseado apenas no conjunto de validação, podemos observar no gráfico que o menor erro de validação se encontra no nível 5 para o Modelo Combinatório. Neste ponto o desempenho no conjunto de treino já está com uma distância considerável do desempenho no conjunto de validação, o que traz indícios de overfitting, mas como o erro de validação ainda não começou a subir, permanecemos com esta escolha. O resultado deste modelo no conjunto de teste é apresentado na tabela 3, que contém também os resultados do baseline para fins de comparação.

Tabela 3. Resultados do modelo Baseline e do melhor modelo de combinação de features

Modelo	MAE Treino	MAE Validação	MAE Teste
Baseline	365.05	367.14	367.14
Modelo Combinatório	283.97	286.97	288.11

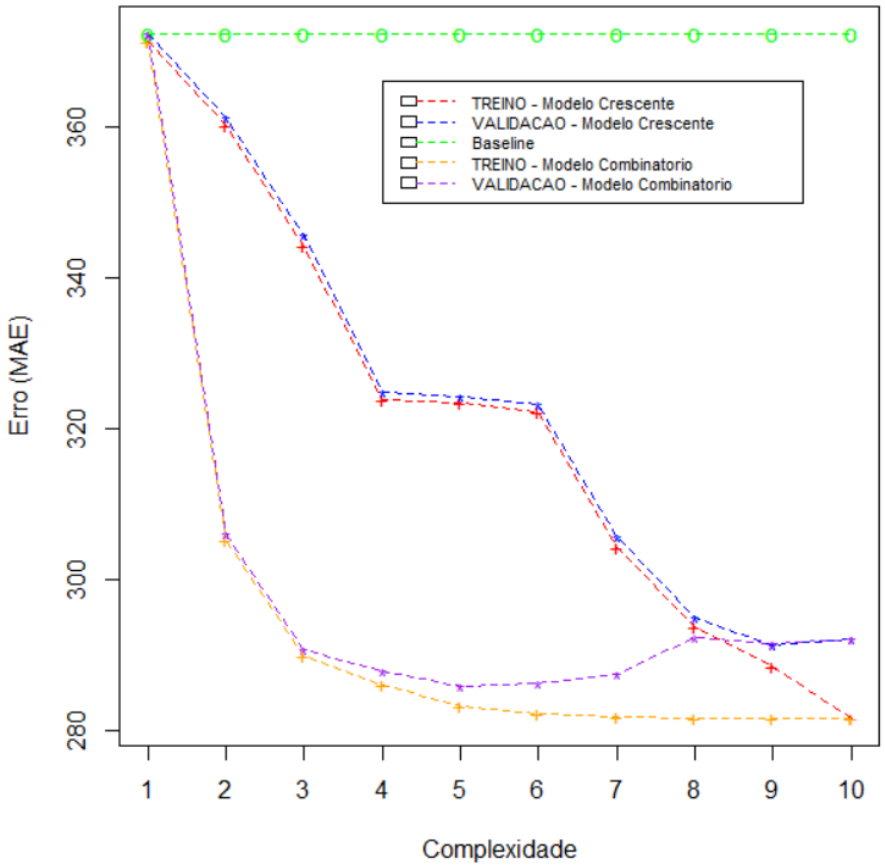

Figura 3. Gráfico do Erro versus Complexidade de dois modelos de combinação de features

QUESTÃO 5) MODELO POLINOMIAL

Para os modelos polinomiais também testamos duas abordagens diferentes.

1. Testar todos os expoentes de 1 a 10 em todas as features numéricas e utilizar as features categóricas em todos os modelos
2. Testar todos os expoentes de 1 a 10 apenas em algumas features numéricas, sem utilizar features categóricas

O resultado das duas abordagens pode ser visto na Figura 4, que contém um gráfico com o desempenho de todos os modelos polinomiais testados, junto com o baseline e a melhor estratégia de combinação da questão anterior, o que chamamos de Modelo Combinatório (cores laranja e azul).Para a abordagem número um, demos o nome de Modelo Exponencial (cores cinza e preto). Já na abordagem número dois, chamamos de Features Selecionadas (cores rosa e marrom). Ainda nesta figura identificamos as regiões de *Underfitting* e *Overfitting* dos modelos exponenciais. A região entre esses dois extremos é onde o erro de treino e o erro de validação são parecidos e permanecem próximos a uma região de mínimo local, que seria o ponto ótimo indicado no gráfico.

Mais para o lado direito do gráfico, na região de alta complexidade, temos a região que indica o overfitting do modelo. Ela é caracterizada pela diferença de comportamento entre a curva de validação e a curva de treino. Enquanto a curva de validação tende a subir, a curva de treino continua descendo. A região de *overfitting* para os dois modelos é praticamente a mesma, pois embora tenhamos adotado estratégias diferentes na seleção de features, o comportamento dos dois modelos neste gráfico é parecido.

A região de *underfitting* também coincide para as duas abordagens, mais à esquerda no gráfico na região de baixa complexidade. Podemos ver que ela corresponde a uma região onde a complexidade é baixa e os erros estão mais elevados do que na região próxima ao ponto ótimo. O ponto ótimo foi escolhido como o ponto onde o erro de validação é menor, e está indicado na cor roxa na Figura 4.

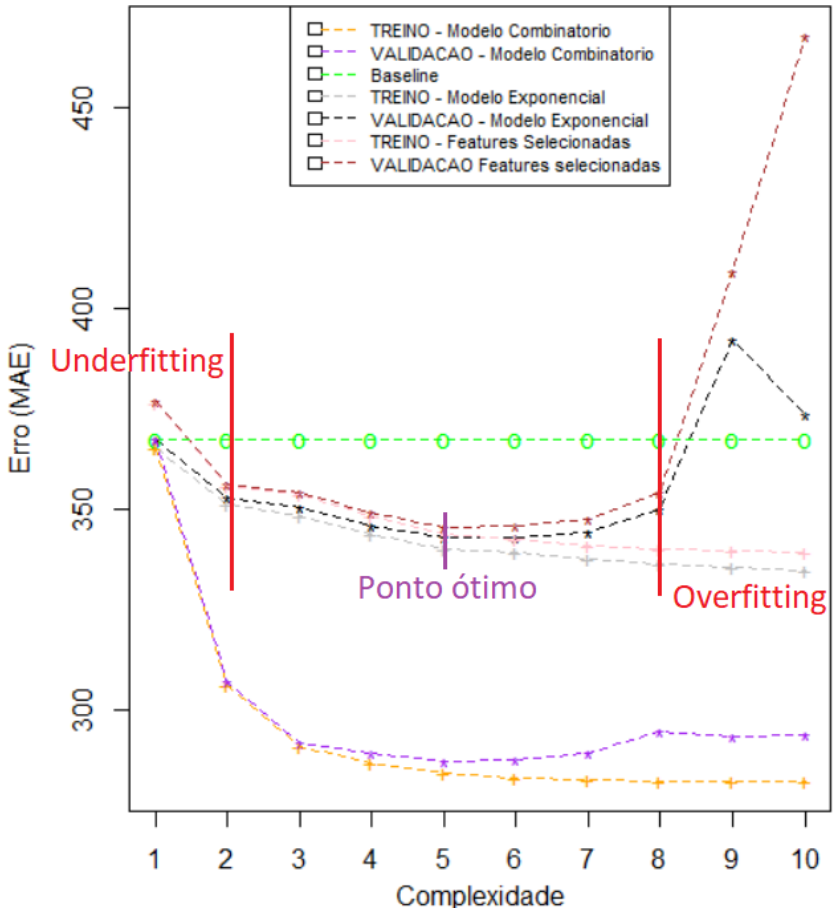

Figura 4. Gráfico com o Erro versus Complexidade dos modelos polinomiais

O modelo correspondente ao ponto ótimo do gráfico da Figura 4 foi utilizado para prever os dados do conjunto de teste. A Tabela 4 compara o resultado deste modelo com os obtidos anteriormente.

Tabela 4. Resultados do modelo Baseline e do melhor modelo de combinação de features

Modelo	MAE Treino	MAE Validação	MAE Teste
Baseline	365.05	367.14	367.14
Modelo Combinatório	283.97	286.97	288.11
Modelo Polinomial	340.11	342.82	343.1

CONCLUSÃO

Com base no gráfico da Figura 4 e na Tabela 4, verificamos que o modelo que apresentou melhor resultado foi o Modelo Combinatório de complexidade 5. A vantagem que ele apresenta em relação ao modelo polinomial, é a combinação de features. O modelo polinomial poderia ter utilizado também a combinação de features, mas como as combinações de features e expoentes eram muitas possíveis, decidimos focar em uma estratégia de cada vez. Sendo assim, o modelo combinatório combinou diversas features de diversas maneiras e obteve um resultado muito melhor na validação e no teste quando comparado ao modelo polinomial que apenas elevou cada uma das variáveis numéricas a diversos expoentes. Podemos notar também que próximo a região do ponto ótimo o modelo polinomial foi um pouco melhor que o baseline. Por fim, o uso das features categóricas não fez muita diferença.