Universidade de São Paulo
ICMC - Instituto de Ciências Matemáticas e Computação

SME0823 - Modelos de Regressão e Aprendizado Supervisionado II

Brenno Hissao Serikawa - 11296697
Caio Assumpção Rezzadori - 11810481
Vitor Beneti Martins - 11877635

Prof. Dr. Mário de Castro

Enunciado

Selecione um conjunto de dados de uma base de dados pública (Reddit, Kaggle, UCI Machine Learning, etc) relacionado a um problema de classificação.

Proponha um modelo de classificação e avalie sua capacidade preditiva. Variáveis explicativas podem ser selecionadas utilizando critérios como GAIC e IV.

1 Introdução

A base de dados escolhida foi retirada do site "Kaggle"[1] e traz dados de diagnóstico de câncer de mama do estado de Wisconsin (EUA), a qual será utilizada para propor o modelo de classificação.

A variável resposta, dada por "diagnosis" assume os valores "B" (tumor benigno) e "M" (tumor maligno) , as quais possuem 357 e 212 observações respectivamente, totalizando 569 observações. Seus valores categóricos foram convertidos para 0 e 1, respectivamente. Além disso, desconsiderando colunas vazias, há 31 variáveis explicativas, coletadas por meio de imagens médicas.

2 Metodologia

2.1 Calculo do valor da informação

A seleção de variáveis inicial para o modelo será feita por meio da métrica IV (valor de informação), dada por:

$$IV_x = \sum_{j}^{r} (p_j - q_j) WoE(j) \tag{1}$$

$$WoE_x(j) = \begin{cases} \ln(\frac{p_j}{q_j}), \text{ caso } p_j \neq 0 \text{ e } q_j \neq 0 \\ 0, \text{ caso contrário} \end{cases}$$

Em que:

- x é a variável numérica sendo analisada;
- r é a quantidade de divisões em intervalos igualmente espaçados dos valores que x pode assumir;
- j representa o j-ésimo intervalo que a variável numérica x está sendo dividida;
- p_j é a porcentagem de observações da categoria 1 cujo valor de x está no intervalo j em relação ao total de observações da categoria 1 (212 observações). Pode ser vista como uma probabilidade empírica da categoria 1 estar no intervalo j;
- q_j é a porcentagem de observações da categoria 0 cujo valor de x está no intervalo j em relação ao total de observações da categoria 0 (357 observações). Pode ser vista como uma probabilidade empírica da categoria 0 estar no intervalo j.

WoE é a abreviação de "Weight of Evidence", e a definição que está sendo feita aqui foi adaptada sobre as definições normalmente encontradas, pois os dados escolhidos possuem intervalos onde não há observações de uma classe ou outra, o que faz com que existam j tais que $p_j = 0$ ou $q_j = 0$, e como $\ln(0)$ e $\ln(p_j/0)$ não estão definidos, decidimos retirar tais intervalos da soma total atribuindo 0 à função WoE quando avaliada sobre estes j. Um outro jeito de contornar este problema é escolher r de tal forma que não existam intervalos sem observações de 0 e 1, mas dado o alto número de variáveis, julgamos necessário fazer esta adaptação para evitar possíveis problemas sobre o cálculo de IV.

A seleção de variáveis será feita seguindo a tabela abaixo.

IV	**Poder preditivo**
< 0.02	Desprezível
$0.02 - 0.1$	Fraco
$0.1 - 0.3$	Médio
$0.3 - 0.5$	Forte
> 0.5	Muito forte

Uma vez com as métricas IV calculadas, será feita uma seleção de variáveis iniciais para ajustar múltiplos modelos de regressão logística para a classificação binária. Após isso, tais modelos serão comparados pela métrica GAIC (Generalized Akaike information criterion) com o intuito de encontrar o conjunto de variáveis que reduz a perda de informação dos dados. Outros critérios serão utilizados para avaliar os modelos e serão apresentados em breve.

2.2 Regressão Logistica

A regressão logística é um método estatístico utilizado para modelar a relação entre uma variável dependente binária (que assume dois valores, como 0 ou 1) e uma ou mais variáveis independentes. Ela é amplamente utilizada em análises de dados onde a variável de interesse é categórica e as respostas não podem ser representadas adequadamente por um modelo de regressão linear.

Formulação

Isso posto, uma forma de desenvolver um modelo preditivo para respostas binárias é por meio da estimação dos parâmetros da função logística/curva sigmoide, dada por:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

onde $\boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^p$, sendo $\boldsymbol{x}$ o vetor de p variáveis preditoras e $\boldsymbol{\theta}$ o vetor dos p parâmetros que se deseja estimar.

Ao proceder com a estimação dos parâmetros da função logística para construir um modelo preditivo, é imperativo incorporar a consideração da independência das observações. Este requisito fundamental assegura que cada observação no conjunto de dados seja independente das demais, evitando potenciais vieses nas estimativas dos parâmetros. Outra consideração essencial é a ausência de multicolinearidade entre as variáveis preditoras. A multicolinearidade, indicada por elevada correlação entre variáveis independentes, pode impactar negativamente a interpretação dos coeficientes estimados.

Tal estimação fornecerá, na verdade, a função de densidade de probabilidade do evento ser a resposta 1, dadas as variáveis que se tem controle. Ou seja, será estimada $f(\boldsymbol{x}) = P(Y = 1 | X = \boldsymbol{x})$,

onde Y é a variável resposta (variável que será predita) e X ao vetor de variáveis preditoras (variáveis controláveis).

Uma vez com a função de probabilidade estimada, será utilizado o seguinte critério de classificação para novas predições:

$$C(x) = \begin{cases} 1, \text{ se } f_{\boldsymbol{\theta}}(x) > 0.5 \\ 0, \text{ caso contrário} \end{cases} \tag{2}$$

2.3 Métricas de avaliação

A avaliação adequada do desempenho de um classificador é fundamental para compreender sua eficácia em tarefas específicas. Neste contexto, várias métricas são empregadas para medir diferentes aspectos do desempenho de um modelo. Neste trabalho, nos concentraremos em quatro métricas essenciais: acurácia, precisão, recall e F1-score.

Divisão do Dataset

Para a construção de todos os modelos criados, será utilizada a divisão **70-30**, em que o *dataset* é dividido em **70%** para treino (ajuste dos modelos) e **30%** para teste (calcular as métricas de avaliação). Isso é necessário para que o modelo treinado realize os testes sobre dados desconhecidos.

Acurácia

A acurácia é uma métrica amplamente utilizada que mensura a proporção de predições corretas em relação ao total de predições. Representada como uma porcentagem, a acurácia oferece uma visão geral do quão bem o modelo está realizando suas previsões. Sua formulação é dada por:

$$Acc = \frac{TN + TP}{TN + TP + FN + FP}$$

Onde TP representa Verdadeiros Positivos, TN representa Verdadeiros Negativos, FP são Falsos Positivos e FN são Falsos Negativos.

Precisão

A precisão focaliza na qualidade das predições positivas do modelo. Sua formulação é dada por:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall

O recall, também conhecido como revocação, mede a capacidade do modelo em identificar todas as instâncias positivas presentes nos dados. Sua formulação é dada por:

$$Recall = \frac{TP}{TP + FN}$$

F1-score

O F1-score é uma métrica que combina precisão e recall em uma única medida, representanda a média harmônica entre essas duas métricas.

$$F1 = 2 \times \frac{P \times R}{P + R}$$

Onde P representa Precisão e R é o Recall.

3 Resultados

Ao calcular o valor da informação, obtemos o resultado disposto na Tabela 2. Percebe-se que a única variável que teve IV desprezível foi a variável de identificação "id", o que é de se esperar, uma vez que a identificação do paciente deve ser independente de seu diagnóstico. Todavia, nenhuma outra variável teve IV abaixo de 0.02 e ser considerado desprezível. Uma possível justificativa para isso é a baixa quantidade de observações da base de dados utilizada.

Modelo 1

O resultado do primeiro ajuste está disposto na Tabela 3. A acurácia do modelo foi de 94,15%. O modelo não retornou um valor para GAIC. Os valores para a **Precisão**, **Recall** e **F1-score** foram:

Classe	Precisão	Recall	F1-score
0	0.962	0.944	0.953
1	0.908	0.937	0.922

Claramente, o ajuste não foi bom. Percebe-se que tanto a estimação dos valores dos coeficientes quanto seus intervalos de confiança explodiram, exibindo a instabilidade do modelo.

Ao investigar as causas disso, encontrou-se a matriz de correlações disposta na Figura 1. Percebe-se que existe um problema de multicolinearidade no modelo, i.e, há muitos pares de variáveis explicativas fortemente correlacionadas, o que explica a imprecisão e instabilidade da estimação do modelo, por exemplo, as variáveis "radius_mean" com as variáveis "area_mean" e "perimeter_mean"com correlações de aproximadamente 99%.

Modelo 2: Modelo 1 sem variáveis fortemente correlacionadas

Uma vez com o problema identificado, geramos a Tabela 4 dos pares de variáveis com mais de 85% de correlação, juntamente com seus IV's. A tabela está ordenada pela coluna "Correlação" e depois pela coluna "IV Max", este último sendo o IV máximo entre as duas variáveis analisadas.

Assim, para cada par com fortes correlações listadas na Tabela 4, foram removidas as variáveis com menor IV: "perimeter_mean", "perimeter_worst", "radius_mean", "area_mean", "area_worst", "perimeter_se", "radius_se", "concave points_mean", "texture_worst", "compactness_worst", "concavity_worst"e "concave points_worst".

O ajuste do modelo é disposto na Tabela 5. A acurácia do modelo foi novamente de 94,15% e seu GAIC de 73.63. Os valores para a **Precisão**, **Recall** e **F1-score** foram:

Classe	Precisão	Recall	F1-score
0	0.962	0.944	0.953
1	0.908	0.937	0.922

Nota-se agora que os coeficientes tiveram muito mais estabilidade em suas estimações. Além disso, muitos deles foram estatisticamente significativos com p-valores abaixo de 5%, rejeitando a hipótese nula de que sejam iguais a zero.

Modelo 3: Modelo 2 com apenas variáveis significativas

Adaptando o modelo anterior e mantendo apenas as variáveis com significância estatística, obtemos o modelo disposto na Tabela 6.

Após tal remoção, a acurácia aumentou para 94,74% e GAIC de 157.21. Os valores para a **Precisão**, **Recall** e **F1-score** foram:

Classe	Precisão	Recall	F1-score
0	0.971	0.944	0.958
1	0.909	0.952	0.930

Novamente, foi encontrado um coeficiente sem significância estatística: "area_se".Todavia, ao ajustar o modelo sem esta variável, tanto as estimativas dos coeficientes quanto as métricas de acurácia e GAIC diferiram muito pouco do modelo com a variável. Isso posto, achamos que não seria interessante colocá-lo no trabalho.

Modelo 4: Modelo 1 com as variáveis que foram retiradas no Modelo 2

Com o intuito de analisar o conjunto de variáveis retiradas no Modelo 1.1, foi também ajustado o modelo com apenas tais preditoras. O resultado está disposto na Tabela 7.

A acurácia foi de 96,49% e GAIC 76.95. Os valores para a **Precisão**, **Recall** e **F1-score** foram:

Classe	Precisão	Recall	F1-score
0	0.964	0.981	0.972
1	0.967	0.937	0.952

Modelo 5: Modelo 4 com apenas variáveis significativas

Percebe-se que a acurácia do Modelo 4 foi superior às demais. Todavia, ao ajustar o modelo novamente apenas com as variáveis com significância estatística, obtemos o ajuste da Tabela 8, cuja acurácia foi de 88,3% e GAIC 326.74. Os valores para a **Precisão**, **Recall** e **F1-score** foram:

Classe	Precisão	Recall	F1-score
0	0.907	0.907	0.907
1	0.841	0.841	0.841

4 Discussão

A Tabela 1 resume os resultados de acurácia e GAIC, obtidos para cada um dos modelos.

Modelos	GAIC	Acurácia
1	-	94,15%
2	73,63	94,15%
3	157,21	94,74%
4	76,95	96,49%
5	326,74	88,3%

Tabela 1: Acurácia e GAIC separada por modelo

Nota-se que o ajuste com menor GAIC foi o do Modelo 2, o que indica que seu conjuto de variáveis explicativas foi o melhor para explicar a variabilidade da resposta e evitar perdas de informação.

Percebe-se, todavia, que o modelo com maior poder preditivo foi o 4, justamente o que foi ajustado com variáveis muito correlacionadas entre si. Uma possível explicação para tal acurácia foi a redundância de informação sobre as variáveis preditoras, o que pode ter acrescentado um "peso" sobre a estimação da probabilidade para certos valores, impactando na classificação. Nota-se porém que seu GAIC foi levemente maior que do Modelo 2, indicando que houve um pouco mais de perda de informação.

A maior diferença surge ao se retirarem as variáveis sem significância estatística destes modelos. Enquanto o Modelo 3 tem um súbito aumento de 0,59% em sua capacidade preditiva e um aumento de 105,36% em seu GAIC quando comparado com o Modelo 2, o Modelo 5 tem uma queda de aproximadamente 8% em sua acurácia e seu GAIC sofre um aumento de 324,61% comparado ao Modelo 4.

Retirar variáveis sem significância estatística é algo muito importante para a interpretabilidade do modelo. Isso posto, a análise feita aqui neste trabalho indica que os modelos 2 e 3 são superiores aos demais modelos quando procura-se um ajuste para propósitos explicativos. Para propósitos preditivos, contudo, o Modelo 4 indica ser o mais adequado.

Comparando o ajuste dos cinco modelos, é evidente que a seleção das váriaveis preditoras é fundamental. A persistência de variáveis não significativas, como "area_se"no Modelo 3 destaca a importância de considerar não apenas a estatística de significância, mas também o contexto do problema. Pode ser necessário explorar abordagens mais avançadas, como regularização, para lidar com variáveis redundantes.

Em resumo, a análise sistemática da multicolinearidade e a seleção cuidadosa de variáveis foram essenciais para a construção dos ajustes estáveis e eficazes. O comprometimento entre a precisão preditiva e a interpretabilidade do modelo deve ser considerado para obter resultados confiáveis na análise estatística dependendo do propósito buscado.

Tabelas e figuras

Variáveis	IV
concavity_mean	3.7192
concavity_worst	3.4629
concave points_worst	3.2601
radius_worst	3.0726
radius_mean	2.9331
area_worst	2.8103
perimeter_worst	2.5524
concave points_mean	2.5109
perimeter_mean	2.3499
area_mean	2.311
compactness_mean	2.0527
area_se	1.9285
compactness_worst	1.8023
radius_se	1.5861
perimeter_se	1.5432
concave points_se	1.2189
texture_mean	1.2098
texture_worst	1.0322
smoothness_worst	0.8697
smoothness_mean	0.6999
symmetry_worst	0.6525
compactness_se	0.6125
symmetry_mean	0.5195
fractal_dimension_worst	0.4924
concavity_se	0.3755
fractal_dimension_se	0.188
fractal_dimension_mean	0.162
symmetry_se	0.104
texture_se	0.0861
smoothness_se	0.0747
id	0.0029

Tabela 2: Resultados IV

Dep. Variable:	diagnosis	**No. Observations:**	569
Model:	GLM	**Df Residuals:**	539
Model Family:	Binomial	**Df Model:**	29
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	nan
Date:	Mon, 13 Nov 2023	**Deviance:**	1.0371e-08
Time:	03:43:01	**Pearson chi2:**	5.19e-09
No. Iterations:	34	**Pseudo R-squ. (CS):**	nan
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
radius_mean	-4198.5057	4.21e+06	-0.001	0.999	-8.25e+06	8.24e+06
texture_mean	90.9265	3.26e+05	0.000	1.000	-6.39e+05	6.39e+05
perimeter_mean	130.5049	3.56e+05	0.000	1.000	-6.97e+05	6.98e+05
area_mean	31.1840	2.27e+04	0.001	0.999	-4.45e+04	4.45e+04
smoothness_mean	3.136e+04	2.82e+07	0.001	0.999	-5.53e+07	5.54e+07
compactness_mean	-3.816e+04	2.39e+07	-0.002	0.999	-4.69e+07	4.68e+07
concavity_mean	2.063e+04	2.16e+07	0.001	0.999	-4.22e+07	4.23e+07
concave points_mean	2.033e+04	2.95e+07	0.001	0.999	-5.77e+07	5.78e+07
symmetry_mean	-1.461e+04	9.88e+06	-0.001	0.999	-1.94e+07	1.93e+07
fractal_dimension_mean	4.746e+04	2.88e+07	0.002	0.999	-5.65e+07	5.66e+07
radius_se	2429.3183	4.6e+06	0.001	1.000	-9.01e+06	9.02e+06
texture_se	-189.8907	3.41e+06	-5.56e-05	1.000	-6.69e+06	6.69e+06
perimeter_se	-808.9518	4.68e+05	-0.002	0.999	-9.19e+05	9.17e+05
area_se	72.5097	4.46e+04	0.002	0.999	-8.73e+04	8.74e+04
smoothness_se	-7.361e+04	3.32e+08	-0.000	1.000	-6.52e+08	6.52e+08
compactness_se	6.53e+04	3.41e+07	0.002	0.998	-6.67e+07	6.68e+07
concavity_se	-5.354e+04	2.12e+07	-0.003	0.998	-4.16e+07	4.15e+07
concave points_se	2.118e+05	2.88e+08	0.001	0.999	-5.63e+08	5.64e+08
symmetry_se	-6.71e+04	6.22e+07	-0.001	0.999	-1.22e+08	1.22e+08
fractal_dimension_se	-5.014e+05	4.41e+08	-0.001	0.999	-8.66e+08	8.65e+08
radius_worst	1101.4212	1.15e+06	0.001	0.999	-2.25e+06	2.25e+06
texture_worst	40.8056	3.46e+05	0.000	1.000	-6.77e+05	6.77e+05
perimeter_worst	39.0208	7.24e+04	0.001	1.000	-1.42e+05	1.42e+05
area_worst	-7.2847	1.06e+04	-0.001	0.999	-2.08e+04	2.08e+04
smoothness_worst	-3280.5012	2.95e+07	-0.000	1.000	-5.79e+07	5.79e+07
compactness_worst	-5159.2425	1.16e+07	-0.000	1.000	-2.28e+07	2.28e+07
concavity_worst	4222.7058	6.24e+06	0.001	0.999	-1.22e+07	1.22e+07
concave points_worst	4729.4242	3.95e+07	0.000	1.000	-7.74e+07	7.74e+07
symmetry_worst	1.472e+04	5.86e+06	0.003	0.998	-1.15e+07	1.15e+07
fractal_dimension_worst	3.674e+04	3.95e+07	0.001	0.999	-7.74e+07	7.74e+07

Tabela 3: Ajuste do Modelo 1

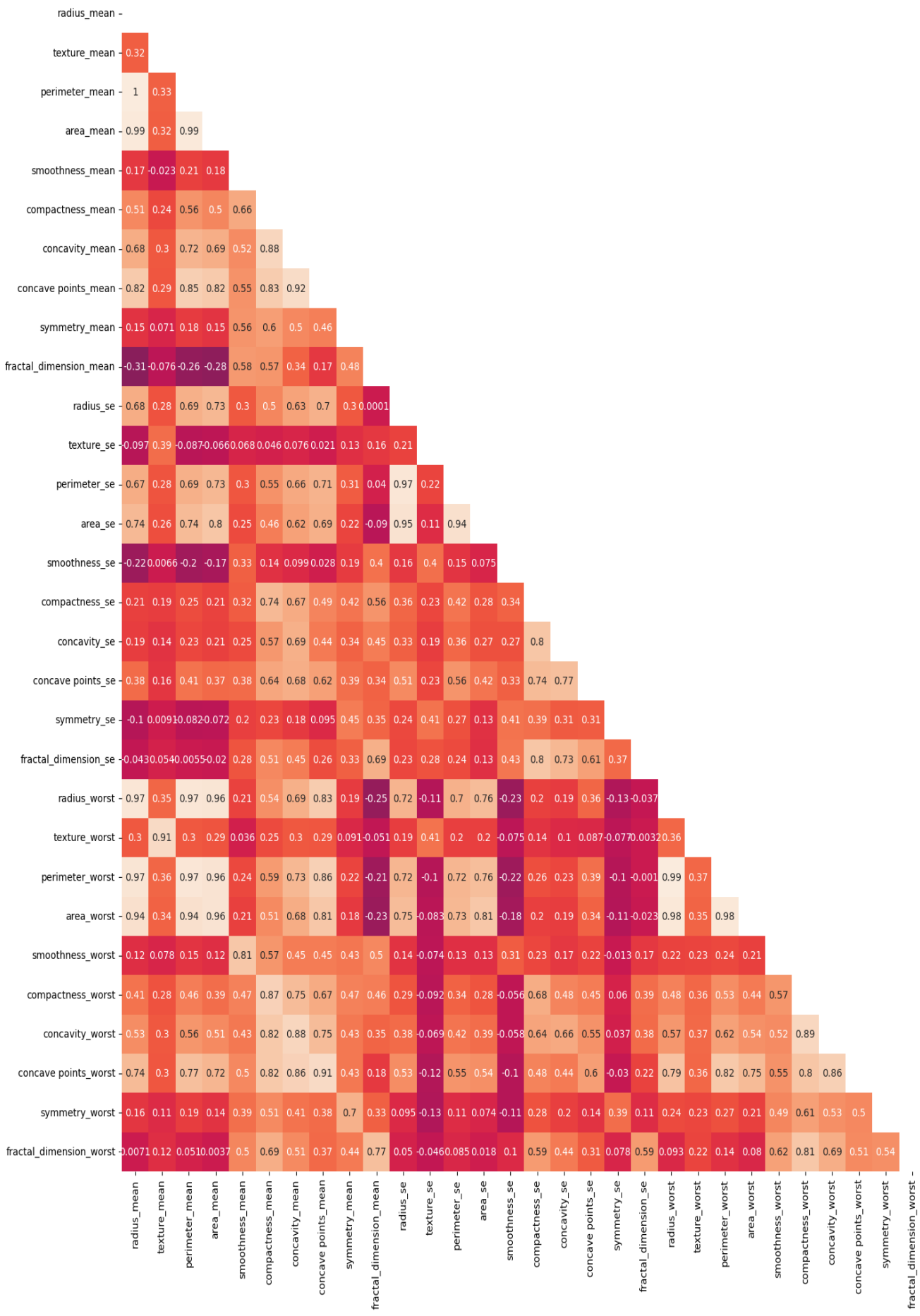

Figura 1: Correlação das variáveis

	Variavel 1	Variavel 2	Correlação	IV 1	IV 2	IV Max
0	radius_mean	perimeter_mean	1.00	2.93	2.35	2.93
24	radius_worst	perimeter_worst	0.99	3.07	2.55	3.07
1	radius_mean	area_mean	0.99	2.93	2.31	2.93
6	perimeter_mean	area_mean	0.99	2.35	2.31	2.35
25	radius_worst	area_worst	0.98	3.07	2.81	3.07
26	perimeter_worst	area_worst	0.98	2.55	2.81	2.81
21	radius_se	perimeter_se	0.97	1.59	1.54	1.59
9	perimeter_mean	perimeter_worst	0.97	2.35	2.55	2.55
2	radius_mean	radius_worst	0.97	2.93	3.07	3.07
8	perimeter_mean	radius_worst	0.97	2.35	3.07	3.07
3	radius_mean	perimeter_worst	0.97	2.93	2.55	2.93
11	area_mean	radius_worst	0.96	2.31	3.07	3.07
13	area_mean	area_worst	0.96	2.31	2.81	2.81
12	area_mean	perimeter_worst	0.96	2.31	2.55	2.55
22	radius_se	area_se	0.95	1.59	1.93	1.93
10	perimeter_mean	area_worst	0.94	2.35	2.81	2.81
4	radius_mean	area_worst	0.94	2.93	2.81	2.93
23	perimeter_se	area_se	0.94	1.54	1.93	1.93
16	concavity_mean	concave points_mean	0.92	3.72	2.51	3.72
5	texture_mean	texture_worst	0.91	1.21	1.03	1.21
20	concave points_mean	concave points_worst	0.91	2.51	3.26	3.26
27	compactness_worst	concavity_worst	0.89	1.80	3.46	3.46
17	concavity_mean	concavity_worst	0.88	3.72	3.46	3.72
14	compactness_mean	concavity_mean	0.88	2.05	3.72	3.72
15	compactness_mean	compactness_worst	0.87	2.05	1.80	2.05
18	concavity_mean	concave points_worst	0.86	3.72	3.26	3.72
19	concave points_mean	perimeter_worst	0.86	2.51	2.55	2.55
28	concavity_worst	concave points_worst	0.86	3.46	3.26	3.46
7	perimeter_mean	concave points_mean	0.85	2.35	2.51	2.51

Tabela 4: Pares de variáveis com correlações superiores à 85%

Dep. Variable:	diagnosis	**No. Observations:**	398
Model:	GLM	**Df Residuals:**	380
Model Family:	Binomial	**Df Model:**	17
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-18.815
Date:	Mon, 13 Nov 2023	**Deviance:**	37.630
Time:	18:16:30	**Pearson chi2:**	39.6
No. Iterations:	12	**Pseudo R-squ. (CS):**	0.7071
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
texture_mean	0.4847	0.182	2.660	0.008	0.128	0.842
smoothness_mean	-74.0803	126.943	-0.584	0.560	-322.885	174.724
compactness_mean	-0.8916	45.358	-0.020	0.984	-89.791	88.008
concavity_mean	151.6342	53.402	2.839	0.005	46.967	256.301
symmetry_mean	-7.0137	39.217	-0.179	0.858	-83.878	69.850
fractal_dimension_mean	-474.4851	236.308	-2.008	0.045	-937.639	-11.331
texture_se	-1.1385	1.733	-0.657	0.511	-4.534	2.257
area_se	0.4088	0.141	2.893	0.004	0.132	0.686
smoothness_se	-293.3957	548.559	-0.535	0.593	-1368.551	781.760
compactness_se	-297.3398	155.236	-1.915	0.055	-601.597	6.917
concavity_se	-76.5417	62.842	-1.218	0.223	-199.710	46.626
concave points_se	434.4955	243.599	1.784	0.074	-42.949	911.940
symmetry_se	-462.2580	281.336	-1.643	0.100	-1013.666	89.150
fractal_dimension_se	-494.2920	707.883	-0.698	0.485	-1881.717	893.133
radius_worst	-0.9179	0.458	-2.006	0.045	-1.815	-0.021
smoothness_worst	41.4971	82.708	0.502	0.616	-120.607	203.601
symmetry_worst	65.4475	34.895	1.876	0.061	-2.945	133.840
fractal_dimension_worst	138.6463	129.306	1.072	0.284	-114.789	392.082

Tabela 5: Ajuste do Modelo 2

Dep. Variable:	diagnosis	**No. Observations:**	398
Model:	GLM	**Df Residuals:**	393
Model Family:	Binomial	**Df Model:**	4
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-73.609
Date:	Mon, 13 Nov 2023	**Deviance:**	147.22
Time:	18:37:09	**Pearson chi2:**	355.
No. Iterations:	8	**Pseudo R-squ. (CS):**	0.6143
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
texture_mean	0.1428	0.050	2.884	0.004	0.046	0.240
concavity_mean	45.1542	6.159	7.331	0.000	33.083	57.226
fractal_dimension_mean	-229.1148	30.222	-7.581	0.000	-288.349	-169.881
area_se	0.0221	0.016	1.393	0.164	-0.009	0.053
radius_worst	0.4046	0.090	4.475	0.000	0.227	0.582

Tabela 6: Ajuste do Modelo 3

Dep. Variable:	diagnosis	**No. Observations:**	398
Model:	GLM	**Df Residuals:**	386
Model Family:	Binomial	**Df Model:**	11
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-26.478
Date:	Mon, 13 Nov 2023	**Deviance:**	52.956
Time:	19:34:06	**Pearson chi2:**	76.0
No. Iterations:	11	**Pseudo R-squ. (CS):**	0.6956
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
perimeter_mean	-0.1632	0.688	-0.237	0.812	-1.511	1.185
perimeter_worst	-0.1707	0.220	-0.775	0.438	-0.602	0.261
radius_mean	-2.3656	4.340	-0.545	0.586	-10.872	6.141
area_mean	0.0163	0.018	0.926	0.355	-0.018	0.051
area_worst	0.0397	0.018	2.267	0.023	0.005	0.074
perimeter_se	1.8185	1.122	1.620	0.105	-0.381	4.018
radius_se	-0.0193	6.465	-0.003	0.998	-12.691	12.653
concave points_mean	33.0427	51.265	0.645	0.519	-67.434	133.519
texture_worst	0.3835	0.104	3.682	0.000	0.179	0.588
compactness_worst	3.6488	9.217	0.396	0.692	-14.417	21.715
concavity_worst	-3.7208	3.686	-1.010	0.313	-10.945	3.503
concave points_worst	64.5548	27.492	2.348	0.019	10.672	118.437

Tabela 7: Ajuste do Modelo 4

Dep. Variable:	diagnosis	**No. Observations:**	398
Model:	GLM	**Df Residuals:**	395
Model Family:	Binomial	**Df Model:**	2
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-160.37
Date:	Mon, 13 Nov 2023	**Deviance:**	320.75
Time:	20:50:14	**Pearson chi2:**	370.
No. Iterations:	6	**Pseudo R-squ. (CS):**	0.4034
Covariance Type:	nonrobust		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
area_worst	0.0028	0.001	5.538	0.000	0.002	0.004
texture_worst	-0.2060	0.020	-10.095	0.000	-0.246	-0.166
concave points_worst	22.7077	3.694	6.147	0.000	15.468	29.948

Tabela 8: Ajuste do Modelo 5

Código

Para a confecção dos resultados do trabalho, foi utilizado a linguagem de programação *Python* no ambiente do **Google Colab**. Para manipulações numéricas e de dados gerais foram utilizadas essencialmente as bibliotecas *numpy* e *pandas*. Já para os ajustes dos modelos foram feitos com a biblioteca *statsmodels*. Para as métricas de desempenho e divisão dos dados em treino e teste, foi utilizada a biblioteca *sklearn*.

O código pode ser acessado em sua integra através do link **https://bit.ly/TrabRegressao2**

Implementação de IV

```
    def iv_woe(data, target, bins=10, show_woe=False):

    #Dataframes novos
    newDF,woeDF = pd.DataFrame(), pd.DataFrame()

    #Colunas dos dados
    cols = data.columns

    #Calculando WoE e IV para todas as variaveis preditoras
    for ivars in cols[~cols.isin([target])]:
        if (data[ivars].dtype.kind in 'bifc') and (len(np.unique(data[ivars]))>10):
            binned_x = pd.cut(data[ivars], bins,  duplicates='drop') # Divisao em
    subintervalos
            d0 = pd.DataFrame({'x': binned_x, 'y': data[target]})
        else:
            d0 = pd.DataFrame({'x': data[ivars], 'y': data[target]})
        d = d0.groupby("x", as_index=False).agg({"y": ["count", "sum"]})
        d.columns = ['Cutoff', 'N', 'Events']
        d['% of Events'] = d['Events'] / d['Events'].sum() # p_j
        d['Non-Events'] = d['N'] - d['Events']
        d['% of Non-Events'] = d['Non-Events'] / d['Non-Events'].sum() # q_j
        d = d.loc[(d['Events'] > 0) & (d['Non-Events'] > 0)] # Retirando intervalos
    sem observcoes das categorias
        d['WoE'] = np.log(d['% of Events']/d['% of Non-Events'])
        d['IV'] = d['WoE'] * (d['% of Events'] - d['% of Non-Events'])
        d.insert(loc=0, column='Variable', value=ivars)

        temp =pd.DataFrame({"Variable" : [ivars], "IV" : [d['IV'].sum()]}, columns =
     ["Variable", "IV"])
        newDF=pd.concat([newDF,temp], axis=0)
        woeDF=pd.concat([woeDF,d], axis=0)

        # Exibindo tabela de WoE
        if show_woe == True:
            print(d)
    return newDF, woeDF
```

Implementação da métrica de acurácia

```
def acuracia(X, Y, modelo):
  df = pd.concat([modelo.predict(X), Y], axis = 1).rename({0: 'probabilidade'}, axis
     = 1)
  df['predicao'] = 0
```

```
    df.loc[df.probabilidade > 0.5, 'predicao'] = 1
    acuracia = df[df['diagnosis'] == df['predicao']].shape[0]/df.shape[0]
    return acuracia, df
```

.

Referências

[1] Kaggle. Breast cancer wisconsin (diagnostic) data set https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data. 2016.