Aprendizagem Estatística em Altas Dimensões

[MAE0501/MAE5904/IBI5904]

Departamento de Estatística (MAE) - IME/USP

Profª Associada Florencia Graciela Leonardi

Relatório Final

Ícaro Maia Santos de Castro (Nº USP 11866921)

✉ icaromsc@usp.br

Rayssa de Carvalho Roberto (Nº USP 10940828)

✉ rayscarvalho@usp.br

Rodrigo Aoyama Nakahara (Nº USP 3510922)

✉ nakahara@usp.br

Rodrigo Marcel Araujo Oliveira (Nº USP 9299208)

✉ rodrigo.marcel.oliveira@usp.br

Vitor Hugo Vieira de Lima (Nº USP 10263886)

✉ vitorhugo@usp.br

2 de Dezembro de 2020

Sumário

Página

Introdução

A diabetes se tornou uma doença muito difundida e muitas pesquisas tem sido feitas nesse sentido para tentar reduzir o crescimento desse problema. Diabetes é um grupo de doenças metabólicas caracterizadas por hiperglicemia crônica resultante de defeitos na secreção de insulina, na ação da insulina ou em ambas. Níveis baixos de insulina para atingir uma resposta adequada e/ou resistência à insulina dos tecidos-alvo são responsáveis por essas anormalidades metabólicas [9]. Existe certa dificuldade no diagnóstico da diabetes, por conta de ser controlada por diversos fatores, tanto ambientais como genéticos. Dessa forma um mecanismo que possa prever se determinado paciente possui a doença ou não, poderia auxiliar muito no diagnóstico dos médicos.

Assim, o objetivo geral do problema foi prever se a pessoa possui ou não diabetes *mellitus*. Com base em uma série de variáveis preditoras, algumas das quais com dados faltantes, o desafio foi posto declaradamente como um problema de aprendizagem estatística com *missings*. Para isso, foi feita uma análise preliminar desses dados faltantes para se decidir sobre duas estratégias para lidar com o problema.

Através de modelos de análise discriminante, regressão logística, florestas aleatórias e máquinas de suporte de vetores foram avaliados os conjuntos de dados de treinamento segundo algumas métricas de desempenho. Ao fim, são apresentados os modelos que apresentaram as melhores performances para se avaliar os conjuntos de teste.

Os códigos comentados em `R` *markdown*, bem como o conjunto de dados utilizados, podem ser acessados no repositório do `GitHub` dedicado a esse projeto:

https://github.com/Diabetes-Database-MAE0501-IME-USP/diabetes-database-mae0501

Partição dos Dados

De maneira a evitar o *data snooping*, o conjunto de dados foi dividido em conjunto de treinamento, conjunto de validação e conjunto de teste. Conforme será explicado, foram utilizados dois bancos de dados: um com dados imputados para os valores faltantes e outro com a eliminação de todas as observações com dados *missings*. Dessa maneira, a análise foi feita sobre seis conjuntos de dados:

[—] **Partição 1: Imputação dos Dados**

Total de 768 Observações (100%)

- Treinamento: 537 (70%)
- Validação: 161 (21%)
- Teste (*out-of-sample*): 70 (9%)

[—] **Partição 2: Eliminação de *Missings***

Total de 382 Observações (100%)

- Treinamento: 266 (70%)
- Validação: 89 (23%)
- Teste (*out-of-sample*): 27 (7%)

Toda a análise descritiva e análise dos *missings* foi feita sobre o conjunto de treinamento da primeira Partição. Os conjuntos de validação foram utilizados apenas para se obter as métricas de performance por cada modelo. Já os conjuntos de teste foram utilizados somente ao fim, após a eleição dos melhores modelos, sem qualquer influência sobre a escolha dos modelos.

Descrição do Banco de Dados e Análise Descritiva

O banco de dados possui todas suas observações constituídas de pessoas do gênero feminino, com idade superior a 21 anos, de ascendência do povo Pima (grupo de nativos norte-americanos). Os dados, que são atualmente abertos e gratuitos para download (CC0 1.0), são provenientes do Instituto Nacional de Diabetes e Distúrbios Digestivos e do Rim (NIDDK) dos EUA e mantidos pela UC Irvine Machine Learning Repository (University of California-Irvine).

O banco conta com 8 variáveis preditoras, e a variável resposta "`Diabetes`" é categórica (dicotômica, 0 ou 1, indicando a presença ou ausência da diabetes) e apresentou 268 indivíduos como diabéticos no total da amostra de tamanho 768. Além disso as observações contam com diversos valores faltantes, e esses valores faltantes foram tratados como zero. Em variáveis como "`BloodPressure`", "`Glucose`", "`SkinThickness`", "`Insulin`" e "`BMI`" onde é biologicamente impossível observar esses valores, um método será proposto para superar essa deficiência. Valores de 0 na variável "`Pregnancies`" não podem ser considerados como dados faltantes, pois é possível que algumas das mulheres da pesquisa realmente não tenham tido gravides ao longo da vida, dessa forma valores de 0 para essa variável, não foram tratado como dados faltantes, de acordo com o trabalho realizado por [2] que reforça a nossa decisão.

Tabela 1: Variáveis do Banco de Dados

Variável	Explicação
`Diabetes`	Variável resposta categórica (1 se diabético, 0 se não diabético)
`Pregnancies`	Quantidade de gestações
`Glucose`	Concentração de glicose no plasma após 2 horas em um teste oral de tolerância a glicose
`BloodPressure`	Pressão arterial diastólica (mm Hg)
`SkinThickness`	Espessura da dobra da pele do tríceps (mm)
`Insulin`	Insulina sérica de 2-horas (mu U/ml)
`BMI`	Índice de massa corporal (peso em kg/(altura em m)^2)
`DiabetesPedigreeFunction`	Função "pedigree" de diabetes
`Age`	Idade (anos)

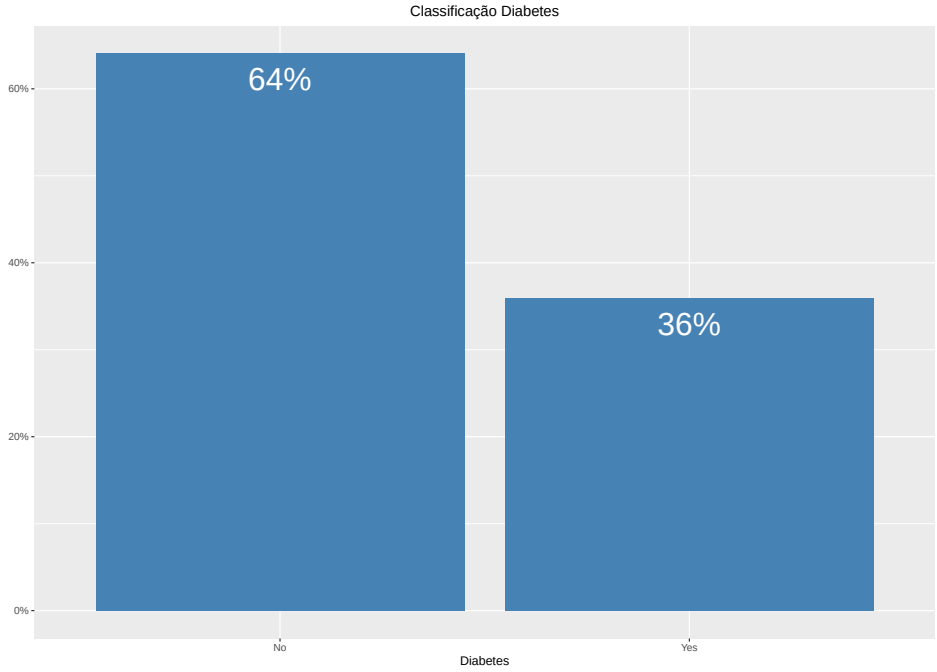
O estudo pioneiro de SMITH *et alii* (1988)[1] foi um dos primeiros a selecionar as variáveis desse banco de dados. Por esse motivo, esse estudo descreve em maiores detalhes a constituição de cada uma dessas variáveis[2].

[1]Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). *Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus.* In: Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261–265). IEEE Computer Society Press.

[2]Por exemplo, a função "pedigree" de diabetes refere-se a uma função de linhagem que avalia a probabilidade de diabetes com base no histórico familiar. Sua metodologia de cálculo é devidamente detalhada no mencionado *paper*.

Partindo-se do conjunto de dados de treinamento (com dados imputados, como será oportunamente explicado) para evitar o *data snooping*[3], a variável resposta `Diabetes` é categórica (dicotômica, 0 ou 1, indicando a presença ou ausência da diabetes), como já mencionado.

Figura 1: Variável Resposta no Conjunto de Treinamento

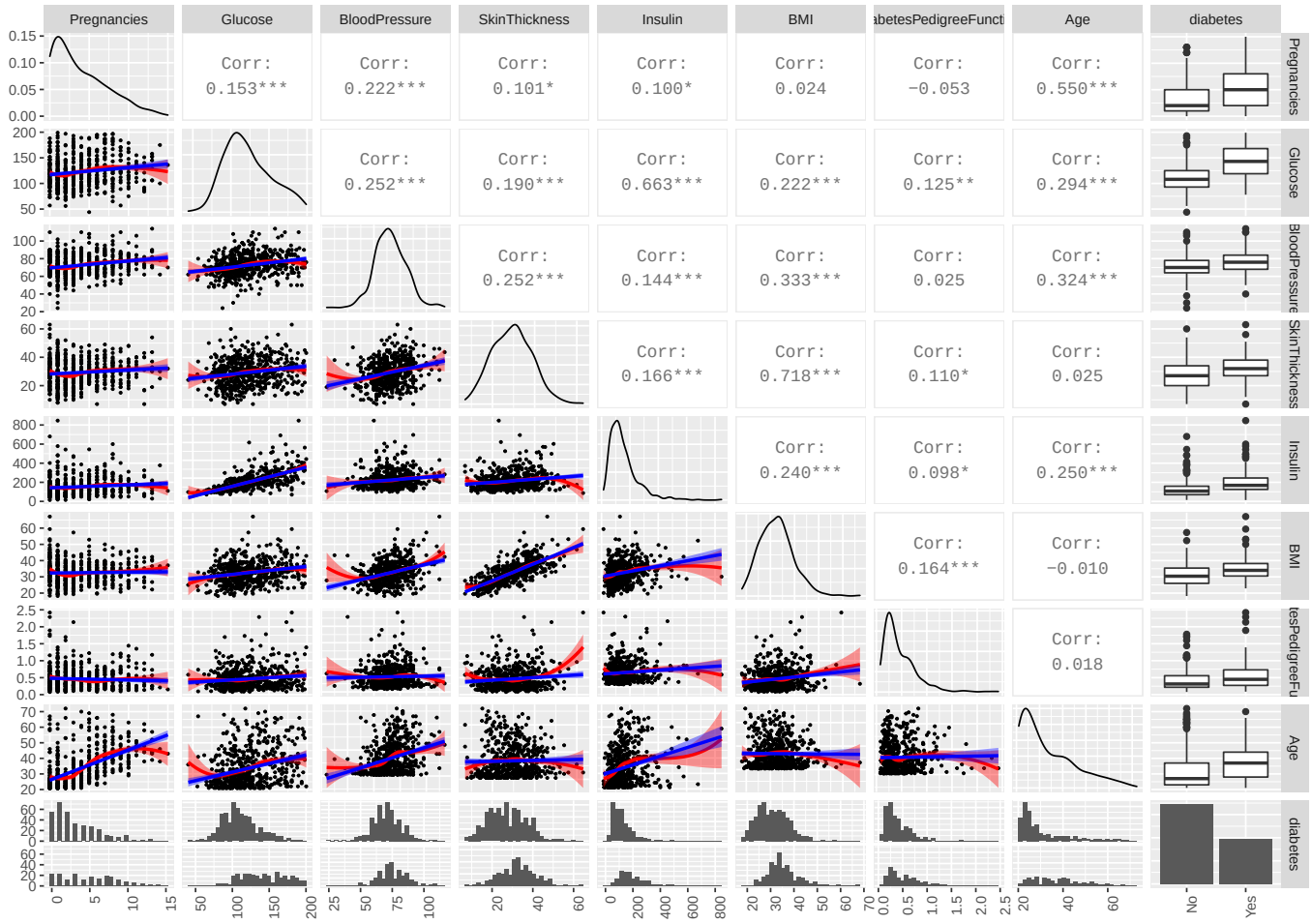


Conforme se observa, existe um desbalanceamento considerável entre as categorias da variável resposta. Enquanto cerca de 64% da base de dados não apresenta diabetes, apenas 36% apresenta a doença. Isso pode gerar problemas nos modelos preditivos que tenderão a favorecer a classe com maior frequência. Por esse motivo, como será explicado oportunamente, conjuntamente com a métrica de desempenho usual de acurácia será utilizada uma métrica balanceada para corrigir por esse desbalanceamento.

Na página seguinte, a Figura 2 traz uma série de informações. Na margem direita do gráfico estão os *boxplots* pelas categorias da variável resposta. Na margem inferior estão os histogramas e densidades marginais alisadas de cada variável. No triângulo superior da matriz, estão as correlações lineares de Pearson com seus respectivos níveis de significância, ou seja: $0\ |{-}{*}{*}{*}{-}|\ 0.001\ |{-}{*}{*}{-}|\ 0.01\ |{-}{*}{-}|\ 0.05\ |{-}.{-}|\ 0.1\ |{-}{-}|\ 1$. No triângulo inferior da matriz estão os gráficos de dispersão entre cada par de variáveis quantitativas, assim como uma reta em azul estimada por mínimos quadrados ordinários (regressão linear simples) e uma curva em vermelho estimada por LOESS. Ambas apresentam uma banda de confiança de 95%.

[3]A análise descritiva sobre todo o conjunto de dados - não somente sobre o conjunto de treinamento - pode induzir o analista a adotar determinados tipos de modelos em detrimento de outros. Esse seria um sutil viés decorrente do *data snooping*.

Figura 2: Correlações, *Boxplots*, Densidades Marginais e Gráficos de Dispersão com Ajustes



Pode-se observar que algumas das variáveis explicativas relacionam-se de maneira aproximadamente linear, o que pode gerar potenciais problemas de colinearidade. No entanto, a maioria delas apresenta uma correlação de Pearson baixa.

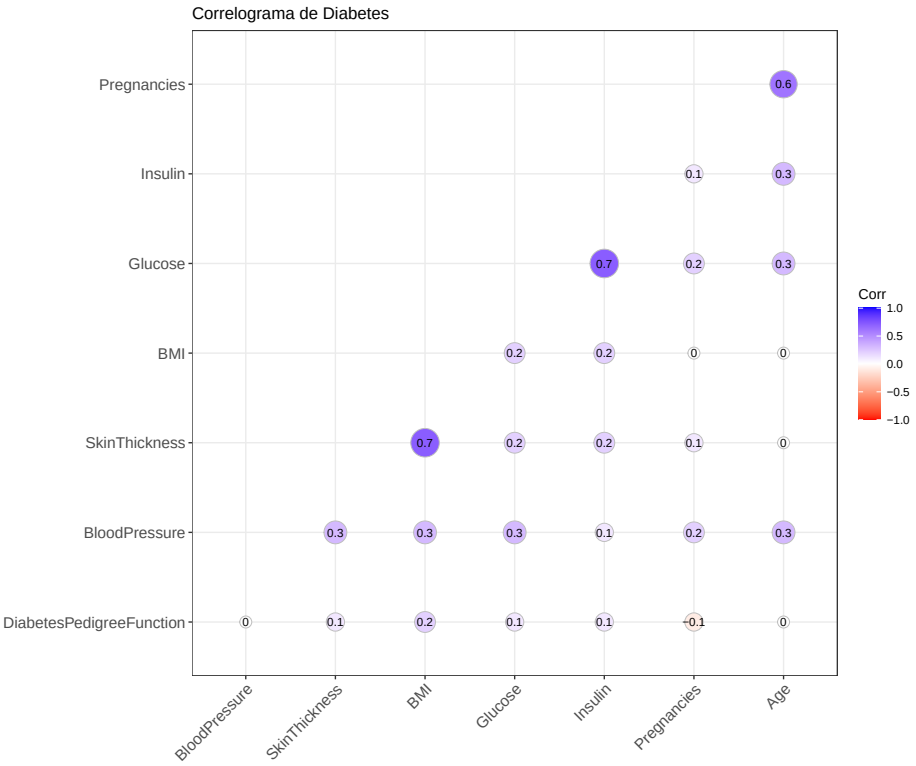
Além disso, percebe-se que uma grande quantidade de variáveis apresenta uma assimetria à direita, o que pode favorecer o método de imputação utilizado, conforme será explicado mais adiante.

O grande destaque é nos *boxplots* por categorias da variável `Diabetes`. Como se observa em todas as variáveis explicativas, a distribuição pela categoria de presença de diabetes é sempre mais elevada que a distribuição pela categoria de ausência da diabetes.

Como o gráfico da Figura 2 contém muitas informações de maneira agregada, são retomadas as mesmas informações de maneira separada nos próximos gráficos de modo a apresentar mais detalhes e com outras visualizações.

O gráfico da Figura 3 mostra o correlograma entre todas as variáveis com uma escala de mapa de calor. Em azul mais escuro são as correlações positivas mais fortes e em vermelho mais escuro são as correlações negativas mais fortes. As quantidades indicadas dentro do gráfico são as correlações lineares de Pearson.

Figura 3: Correlograma



Como se observa, a grande maioria da associações lineares são relativamente fracas. Destacam-se algumas poucas exceções como é o caso da associação entre `Age` e `Pregnancies`, entre `Insulin` e `Glucose`, e entre `BMI` e `SkinThickness`. Todas essas associações são biologicamente esperadas que sejam elevadas. Portanto, não há nenhuma associação linear que foge do que seria esperado.

Os gráficos da Figura 4 na página seguinte trazem as densidades alisadas de cada variável explicativa por categoria da variável resposta. As curvas em azul representam a ausência da diabetes e as curvas em vermelho representam a presença da diabetes. Como já destacado na Figura 2, verifica-se um deslocamento à direita em todas as variáveis para a distribuição pela categoria da presença da diabetes.

Por fim, os *boxplots* da Figura 5 por variáveis explicativas também destacam esse deslocamento. Nesses gráficos, os pontos dentro das caixas representam as medianas das respectivas distribuições. Como se verifica, existe uma assimetria à direita em quase todas as distribuições.

Figura 4: Densidades Marginais: *Features* por Categoria de Diabetes

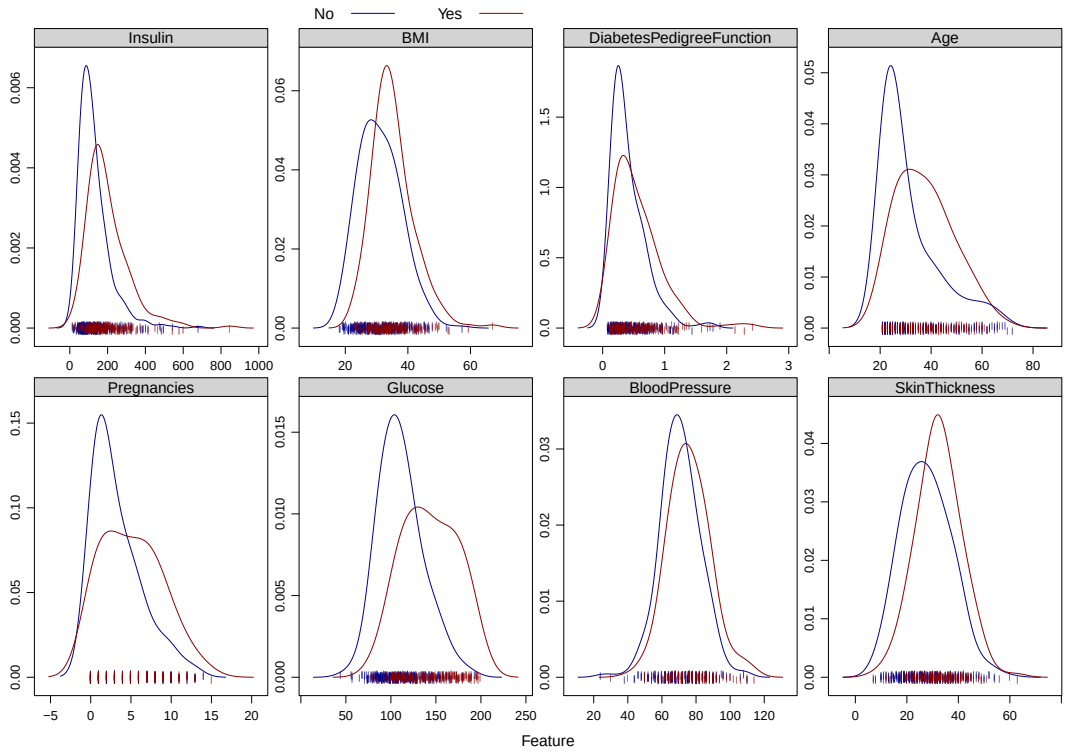
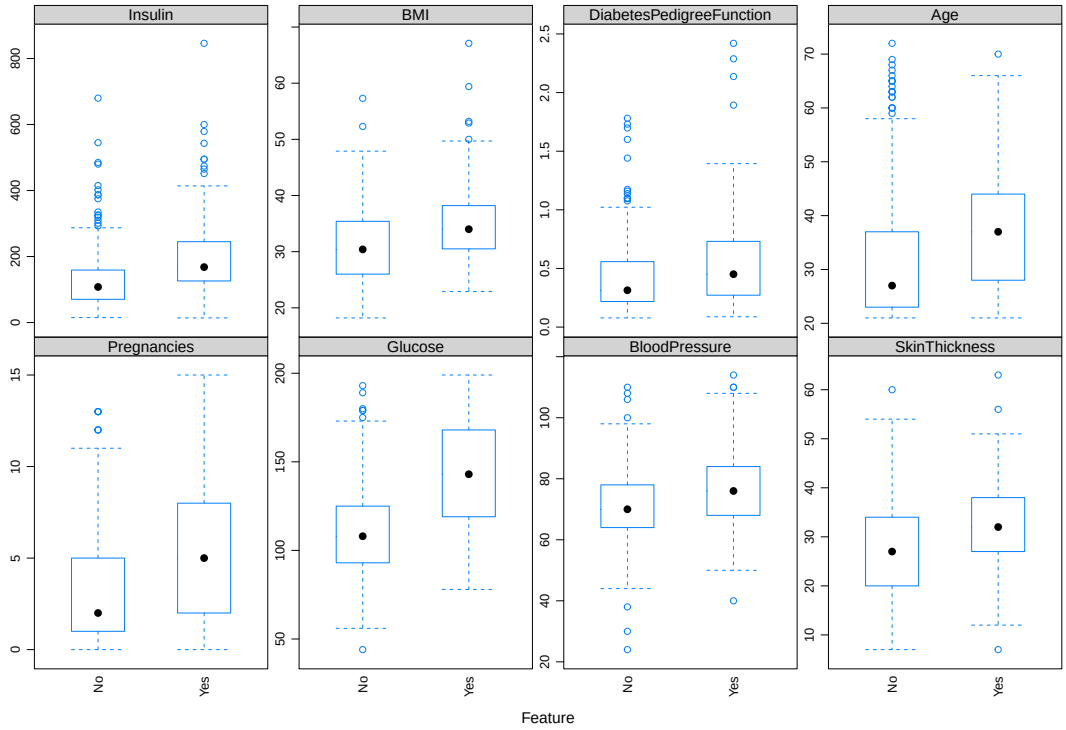


Figura 5: *Boxplots*: *Features* por Categoria de Diabetes

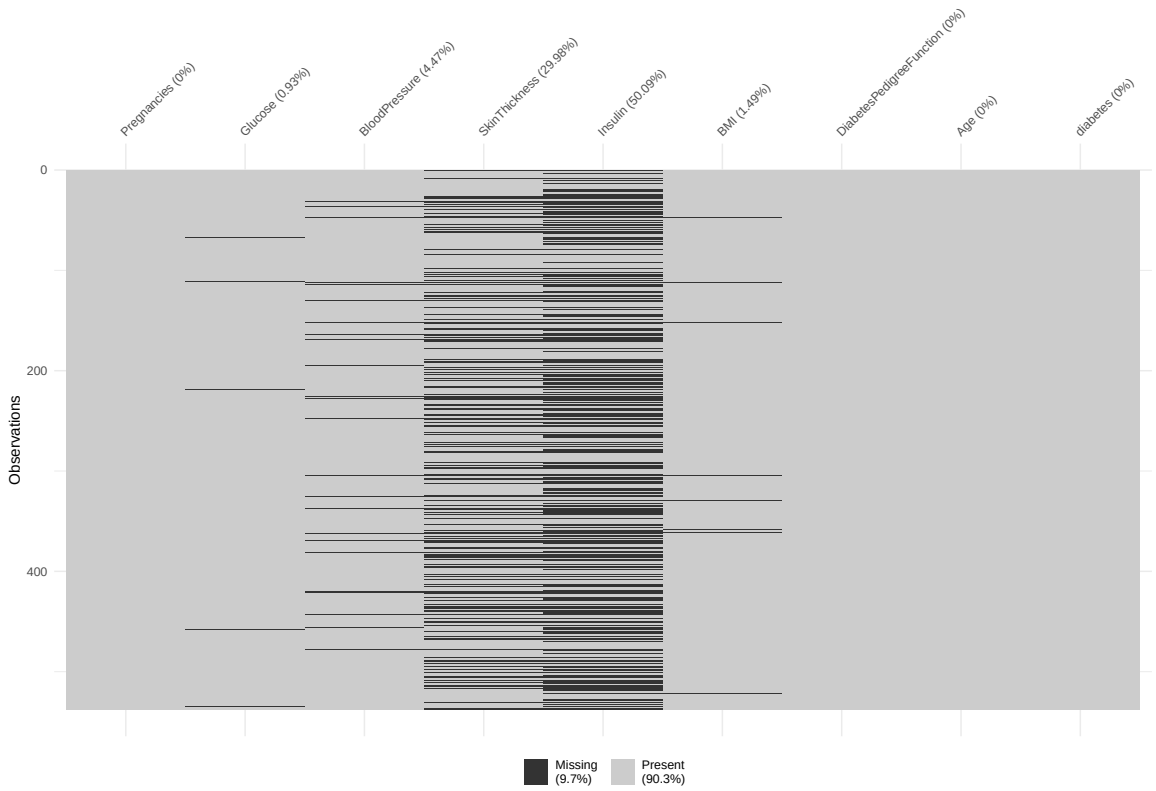


Análise de Dados Faltantes e Estratégia de Imputação

Como explicado, foram considerados como dados ausentes os valores registrados como zero na base de dados para as variáveis as quais são biologicamente impossíveis de apresentarem esse valor. Assim, foi feita uma análise preliminar dos dados faltantes (*missings*) para se verificar a possibilidade de imputação de dados, de maneira a não reduzir expressivamente a quantidade de observações originais com a eliminação dos registros para as análises.

Para todas as variáveis foi feito um mapeamento dos *missings* para se investigar a existência de padrões. A imputação de dados para *missings* que apresentam padrões ou regularidades implica a introdução de vieses não desejados aos dados originais[4]. Por isso é importante que sejam MAR (*missing at random*) ou MCAR (*missing completely at random*).

Figura 6: Mapeamento dos Dados Faltantes nas *Features*

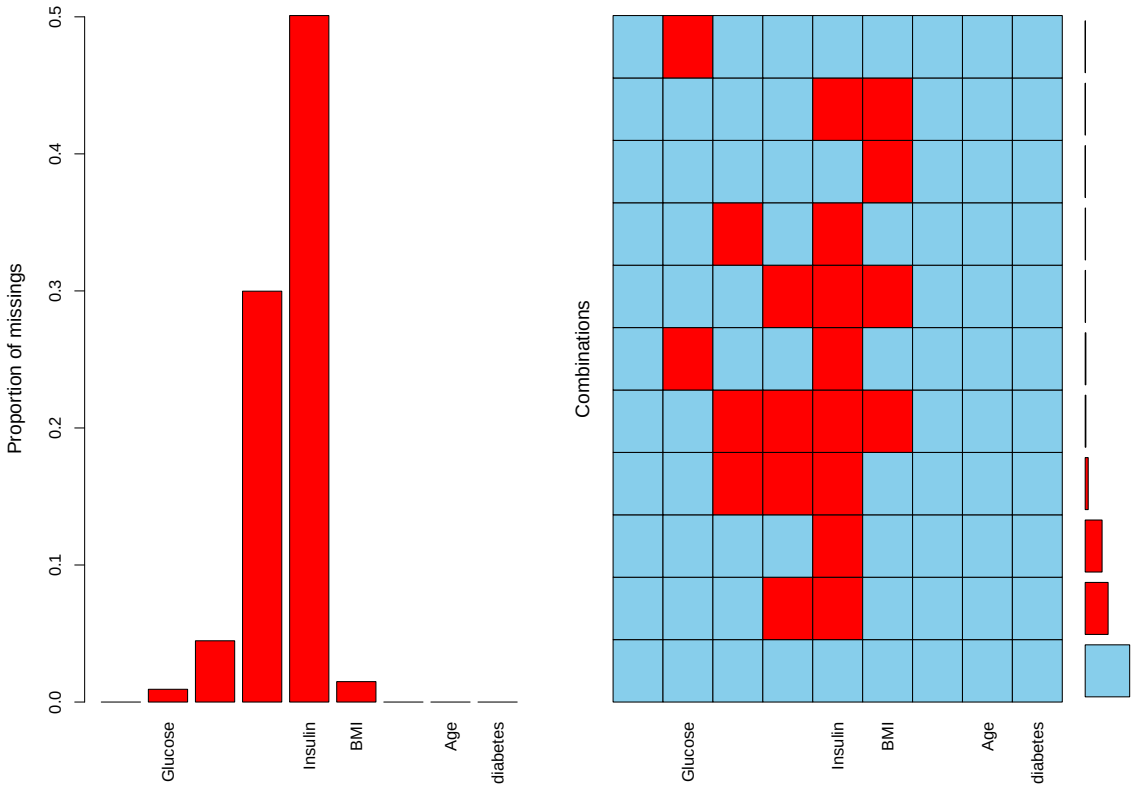


[4]Nesse caso, foi feito uma análise superficial dos *missings*, sem adentrar em técnicas específicas para esse tipo de análise. Técnicas mais avançadas contemplam inclusive a modelagem de *missings*. Mas, como não era esse o objetivo dessa análise, foi feita apenas a verificação de padrões dos dados faltantes.

A Figura 6 mostra um total de 9.7% de *missings* espalhados pelas variáveis `Glucose`, `BloodPressure`, `SkinThickness`, `Insulin` e `BMI`. Em especial, as duas variáveis explicativas que mais apresentam *missings* são `Insulin` e `SkinThickness`. Caso os 9.7% estivessem concentrados nas mesmas observações, não haveria grande perda com a eliminação dessas observações. Mas, como se encontram espalhados por várias observações, a estratégia de imputação dos dados passou a ser uma possibilidade de remediação do problema.

Para analisar com mais detalhes, a Figura 7 traz dois gráficos. O primeiro é a proporção de *missings* por variável em relação ao total dos dados. O segundo considera combinações aleatórias de observações e verifica as frequências de *missings* por variável. Os destaques em vermelho são os *missings* e os em azul são os não-*missing*. As barras no canto direito do gráfico indicam as frequências de *missing*, exceto a última linha que não apresentou dados faltantes.

Figura 7: Proporções de *Missings* nas *Features* e Combinações de Blocos de Observações



Como se observa, não parece haver um padrão ou uma regularidade explícita para os *missings* em relação aos não-*missings* ou em relação às variáveis em si. Isso significa que a imputação dos dados não seria tão prejudicial ao conjunto de dados original. Ou seja, eventuais tendências não seriam decorrentes da imputação por si, mas muito provavelmente seriam decorrentes das próprias características originais dos dados.

Por fim, a Figura 8 traz os gráfico de dispersão entre as variáveis com as maiores quantidades de *missings*, `Insulin` e `SkinThickness`. Nela se pode observar a a amplitude e variabilidade dos *missings* em relação aos não-*missings* para as variáveis.

Figura 8: Dispersão entre `Insulin` e `SkinThickness` Para *Missings* e Não-*Missings*

Como se observa, os *missings* não parecem apresentar tendência ou se concentrar de maneira desproporcional. Pelo contrário, parecem acompanhar o suporte amostral da distribuição dos não-*missings*. Essa é outra evidência em favor da imputação dos dados.

Considerando-se toda a análise dos *missings*, decidiu-se pela imputação dos dados por meio do uso do pacote `mice`[5] para `R`. Em termos gerais, o pacote gera uma imputação multivariada por equações encadeadas (*Multivariate Imputation by Chained Equations*).

Dentre as várias opções de métodos de imputação disponíveis pelas *chained equations*, decidiu-se pelo método `pmm` (*predictive mean matching*)[6]. O algoritmo basicamente faz um *pool* de valores próximos ao *missing* (valores candidatos) considerando todas as variáveis para se selecionar aleatoriamente um

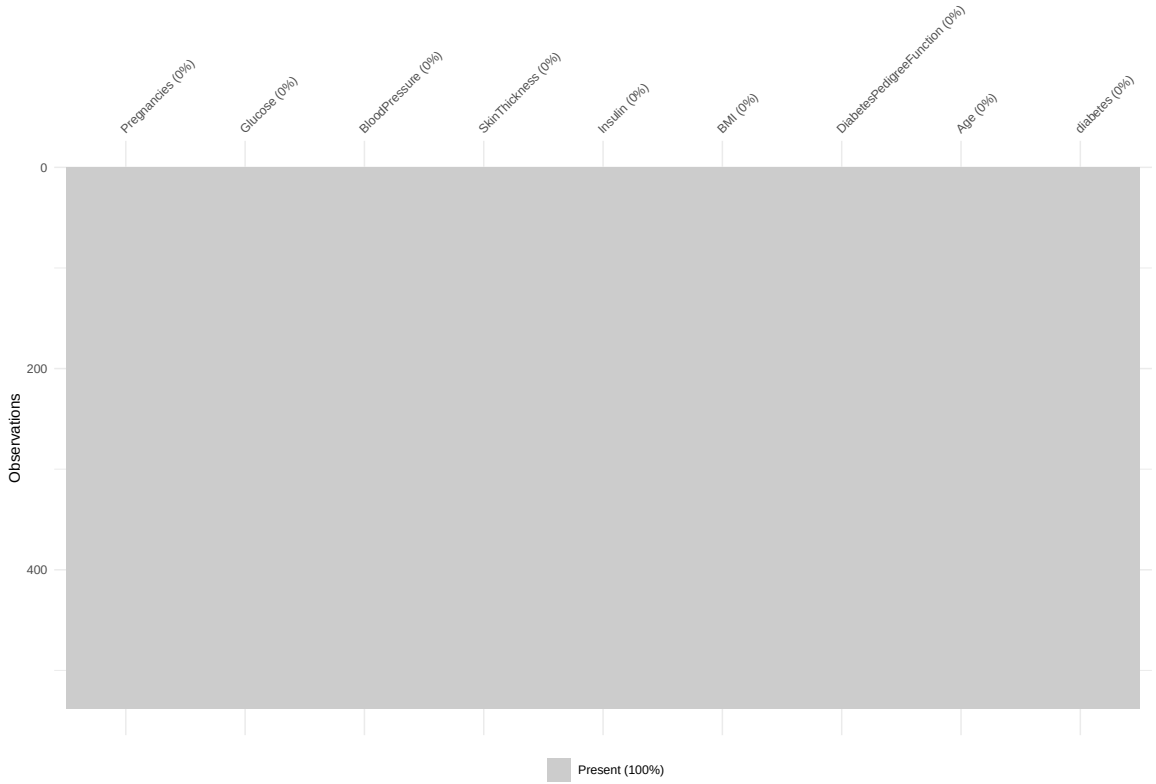
[5]Para maiores detalhes, cf. a documentação do pacote, disponível em https://cran.r-project.org/web/packages/mice/mice.pdf.

[6]Para maiores informações e exemplos, cf. a documentação do pacote ou o endereço https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html.

desses valores e imputar no lugar do dado ausente. A vantagem dessa imputação sobre uma imputação puramente estocástica é a impossibilidade de se imputar dados absurdos ou incoerentes com a natureza da variável, como, por exemplo, imputar dados negativos a variáveis estritamente positivas. Além disso, segundo a documentação, o método `pmm` é recomendado para dados assimétricos, ou multimodais, ou com valores inteiros.

O resultado da imputação dos dados faltantes, conforme a explicação, é ilustrado na Figura 9.

Figura 9: Dados Faltantes: Resultado do Mapeamento Após Imputação



Como se observa, 100% dos dados passam a estar presentes no banco de dados, diferentemente do que foi visto na Figura 6. Além disso, a Figura 10 na página seguinte mostra que não há mais dados faltantes para nenhuma variável e que quaisquer blocos de combinações de observações apresentam todos os dados como não-*missing*. A Figura 10 mostra o resultado da imputação, em contraste com o que foi mostrado na Figura 7.

Por fim, os gráficos da Figura 11 na página seguinte mostram as densidades originais das variáveis (em azul) e as densidades das simulações (em vermelho).

Para cada variável com *missings* foram gerados cinco banco de dados simulados pelo `mice` e se obteve

a média dos cinco para se imputar os dados faltantes.

Figura 10: Dados Faltantes: Resultado das Proporções e Combinações Após Imputação

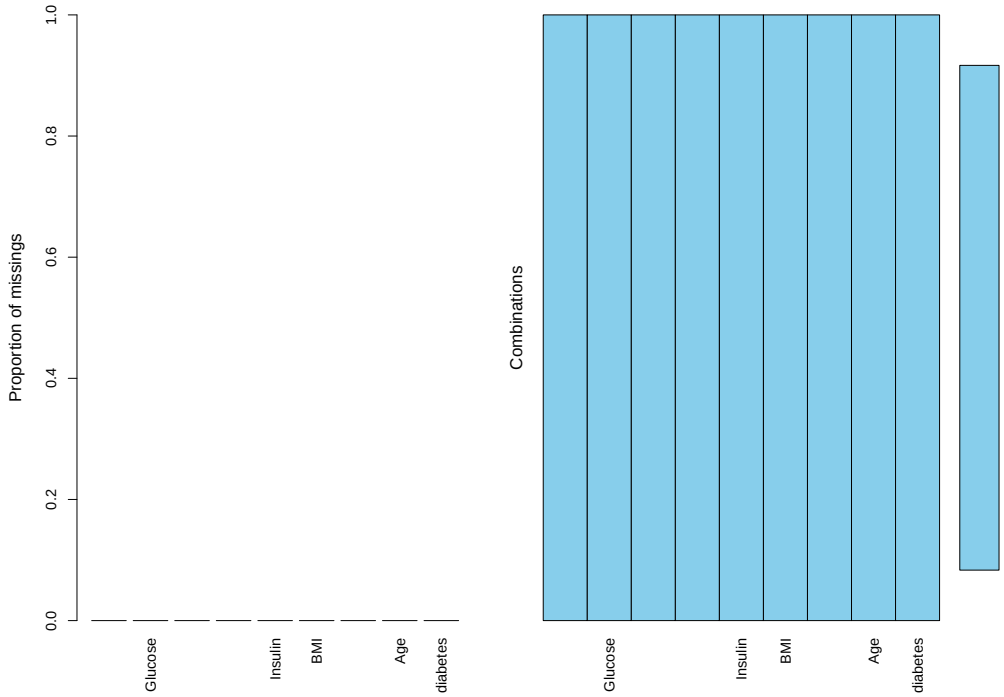
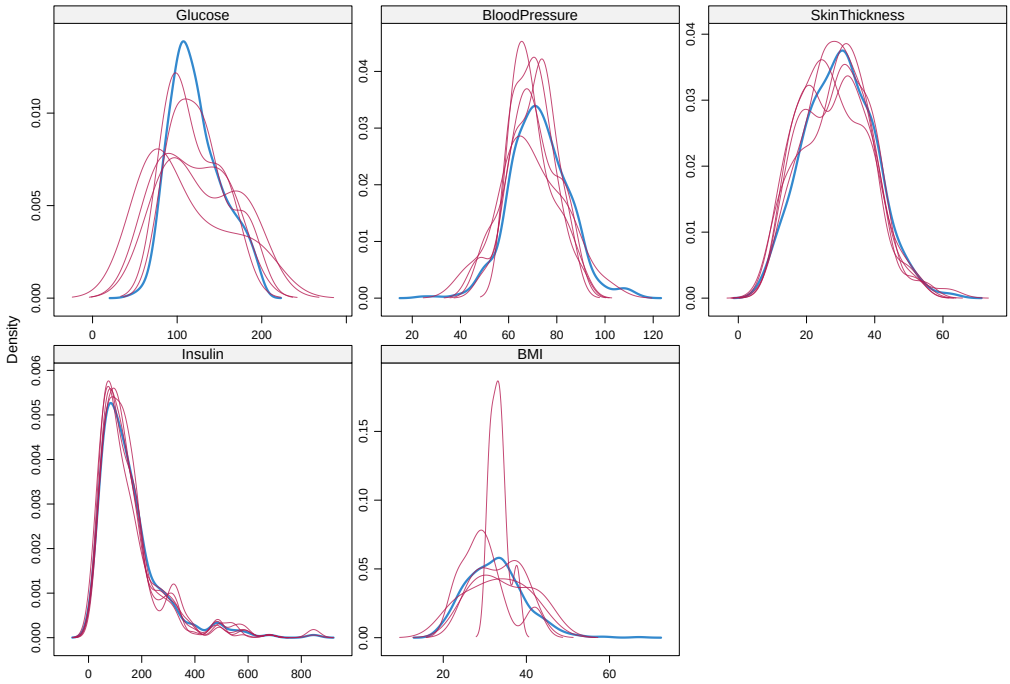


Figura 11: Dados Faltantes: Densidades Marginais Originais e Simulados das *Features*



Critérios de Desempenho dos Modelos

Para avaliarmos o desempenho dos modelos ajustados e posteriormente elegermos os melhores, nos baseamos principalmente em métricas derivadas da matriz de confusão, a seguir explicamos o objetivo dessas métricas e mostramos suas formulas de cálculo.

Métricas Baseadas na Matriz de Confusão

A matriz de confusão consiste em uma matriz (2x2) onde as respostas reais e preditas são alocadas de forma que possamos avaliar o desempenho de um modelo de classificação, nesta matriz as linhas correspondem a classe real do indivíduo e as colunas a classe predita. O nome matriz de confusão deriva do fato de tornar fácil verificar se o sistema (predito x real) esta se confundindo, a seguir temos uma ilustração de uma matriz de confusão genérica.

Tabela 2: Matriz de Confusão

		Valor Predito	
		Positivo	**Negativo**
Valor Real	**Positivo**	n_{11}	n_{12}
	Negativo	n_{21}	n_{22}

a partir dessa matriz podemos definir algumas métricas utilizadas na avaliação de modelos de classificação [7].

- **Sensibilidade (SEN):** Calcula a proporção de observações positivas classificadas corretamente pelo modelo, quanto maior a sensibilidade de um modelo melhor será a capacidade de ele classificar as observações positivas corretamente.

$$SEN = \frac{n_{11}}{n_{11} + n_{12}}$$

[7] Como exemplo da utilização dessas métricas em modelos de crédito, cf. [4] que compara diferentes modelos de regressão logística de *credit scoring* em um caso brasileiro.

- **Especificidade (ESP):** Semelhante a sensibilidade essa medida verifica a proporção de negativos classificados corretamente pelo modelo, em geral um modelo bom tem alta sensibilidade e alta especificidade.

$$ESP = \frac{n_{22}}{n_{21} + n_{22}}$$

- **Acurácia (ACU):** Essa medida mede a proporção de predições corretas do modelo, quanto maior seu valor maior a taxa de acerto do modelo [14].

$$ACU = \frac{n_{11} + n_{22}}{n_{11} + n_{12} + n_{21} + n_{22}}$$

- **Acurácia Balanceada (ACUB):** Essa medida é semelhante a acurácia mas leva em conta o desbalanceamento dos dados.

$$ACUB = \frac{SEN + ESP}{2}$$

- **Prevalência (PRE):** É a proporção de individuos positivos, serve para avaliarmos o desbalanceamento dos dados.

$$PRE = \frac{n_{11} + n_{21}}{n_{11} + n_{12} + n_{21} + n_{22}}$$

- **Valor Predito Positivo (VPP):** É a probabilidade que individuos classificados como positivo sejam de fato positivo.

$$VPP = \frac{n_{11}}{n_{11} + n_{12}}$$

- **Valor Predito Negativo (VPN):** É a probabilidade que individuos classificados como negativo sejam de fato negativo.

$$VPN = \frac{n_{22}}{n_{21} + n_{22}}$$

- **Taxa de detecção (TD):** É a proporção de individuos positivos detectados pelo modelo dentre todos os individuos.

$$TD = \frac{n_{11}}{n_{11} + n_{12} + n_{21} + n_{22}}$$

- **prevalência da detecção (PD):** É a proporção de individuos preditos positivos dentre todos os individuos preditos.

$$PD = \frac{n_{11} + n_{12}}{n_{11} + n_{12} + n_{21} + n_{22}}$$

- **Estatistica Kappa (Kappa):** Esta estatística verifica a proporção da concordância entre as categorias preditas e as categorias verdadeiras.

$$Kappa = \frac{\theta_o - \theta_e}{1 - \theta_e}$$

onde, $\theta_o = n_{11} + n_{11}$ é a concordância total observada e θ_e é a concordância total esperada, obtida pela formula a seguir:

$$\theta_e = \sum_{i=1}^{2} \theta_i,\ \theta_i = \frac{1}{(n_{11} + n_{12} + n_{21} + n_{22})^2} \sum_{j=1}^{2} n_{ji} \sum_{j=1}^{2} n_{ij}$$

Outras Métricas

Além das medidas derivadas da matriz de confusão, existem também medidas que avaliam a qualidade do ajuste como AIC, BIC e log-verossimilhança, essas medidas foram utilizadas em alguns modelos para regulariza-los, a seguir falamos de forma sucinta sobre essas métricas.

- **AIC, BIC, Log-verossimilhança:** Essas métricas são utilizadas em modelos de classificação, auxiliando na escolha das variáveis relevantes para o modelo e na estimativa dos parâmetros.
  - **log-verossimilhança:** O log da verossimilhança mede o quão bem o modelo se ajusta aos dados, como essa medida varia entre $[-\infty, 0]$ quanto mais próximo de 0 melhor o modelo se ajusta.
  - **AIC, BIC:** Essas duas medidas seguem o mesmo princípio do log da verossimilhança, no entanto aplicam penalidades de acordo com o número de parâmetros. A formula de cálculo para essas métricas podem aparecer de diversas formas dependendo das suposições adotadas na modelagem, a seguir apresentamos a forma de calculo mais genérica adota por [1]

$$AIC = -2l + 2p$$

$$BIC = -2l + log(n)p + C$$

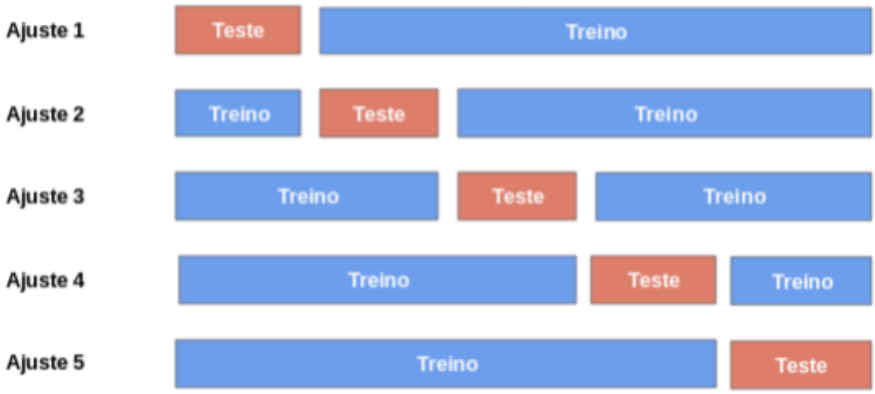
onde, l é o logaritmo da verossimilhança, p é o número de parâmetros, n é o tamanho amostral e C é uma constante que depende somente da amostra.

Validação Cruzada

A situação ideal para avaliação da capacidade preditiva de um modelo ocorre quando existem dois conjuntos de dados, o primeiro, chamado conjunto de treinamento, que será usado para ajustar o modelo e o segundo, chamado conjunto de teste, onde serão realizadas as predições do modelo e posterior cálculo das medidas de qualidade das predições. No entanto, esse cenário quase nunca ocorre, o que faz com que modelos sejam ajustados e testados no mesmo conjunto de dados, tornando as medidas de avaliação mais otimistas do que deveriam ser.

Devido a isso, se fazem necessárias as técnicas de validação cruzada, que dividem o conjunto de dados em subconjuntos a serem usados como dados de treinamento e de teste, para os quais o mesmo modelo é treinado e testado gerando estimativas melhores para o desempenho do modelo e tornando as medidas da capacidade do modelo mais condizentes com seu verdadeiro desempenho. A seguir apresentamos algumas dessas técnicas de validação cruzada, para um leitor que tenha interesse em se aprofundar no tema recomendamos [3], que ao final apresenta uma recomendação de como escolher a melhor técnica de validação cruzada

- ***Hold-out*:** Essa técnica consiste simplesmente na divisão do conjunto de dados de forma aleatória em dois grupos, um de treinamento e outro de teste. É comum que o grupo de treinamento seja maior que o de teste, geralmente o tamanho do grupo de treinamento é 2/3 do banco de dados.
- ***K-fold*:** Nesta técnica o conjunto de dados é dividido em k partes, em cada ajuste uma dessas partes é considerada como conjunto de testes e as demais como conjunto de treinamento. A cada ajuste são coletadas as estimativas dos parâmetros e as medidas de desempenho, ao final do processo são calculadas suas médias que serão as estimativas e medidas finais. A seguir temos uma imagem ilustrativa do processo retirada de [12].



- ***Leave-One-Out Cross-Validation* (LOOCV):** Essa técnica é um caso extremo do *k-fold* quando k é igual ao tamanho do banco de dado, ou seja, todo banco de dados será usado tanto para treinar quanto para testar o modelo. Uma desvantagem dessa técnica é seu alto custo computacional para conjuntos grandes de dados.

Bootstrap

Essa técnica é utilizada nos modelos de árvores e tem por objetivo gerar diversos conjuntos para treinamento, através de uma amostragem com reposição do banco de dados, esses conjuntos tornam mais precisas as estimativas dos parâmetros do modelo e facilitam a obtenção dos erros das estimativas sem a necessidade de uma teoria mais complicada. A seguir apresentamos o algoritmo para o cálculo do erro padrão de um estimador $\hat{\theta} = t(x)$ descrito por [12].

1. Selecione B amostras independentes $x_1^*, \ldots, x_B^*$ cada uma consistindo de n (tamanho do conjunto de dados) valores do conjunto de dados. Tome $B \approx 200$
2. Para cada amostra *bootstrap* x_b^* calcule a réplica *bootstrap* do estimador

$$\widehat{\theta}^*(b) = t(x_b^*), \ b = 1, \ldots, B$$

3. o erro padrão de $\hat{\theta}$ é estimado pelo desvio padrão das B réplicas.

$$\widehat{e.p.}_B = \left[\frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}^*(b) - \hat{\theta}^*(.))^2 \right]^{\frac{1}{2}}$$

com

$$\widehat{\theta}^*(.) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b)$$

Modelos e Análise de Desempenho

Nesta seção são apresentados cada um dos modelos utilizados para avaliar a capacidade de predizer se a pessoa apresenta ou não diabetes. Em cada modelo foram utilizados os conjuntos de treinamento tanto da Partição 1, quanto da Partição 2. Sobre os conjuntos de treinamento foi utilizada uma validação cruzada k-fold com 10 ou 15 partes a depender do modelo. Em seguida, foram utilizados os conjuntos de validação - da Partição 1 e da Partição 2 - para se obter as métricas já explicadas.

Foram utilizados os seguintes conjuntos de modelos:

[–] Análise Discriminante (AD)

- Análise Discriminante Linear
- Análise Discriminante Flexível
- Análise Discriminante Quadrática

[–] Regressão Logística (RL)

- Regressão Logística Simples
- Regressão Logística Regularizada

[–] Métodos Baseados em Árvores

- Florestas Aleatórias

[–] Máquinas de Suporte Vetorial (SVM)

- Máquinas de Vetores de Suporte com *Kernel* Linear
- Máquinas de Vetores de Suporte com *Kernel* Não-Linear

Análise Discriminante

Conforme explicam Johnson & Wichern (2007), a análise discriminante insere-se na classe de métodos para discriminação e classificação dos dados. Enquanto a discriminação tem como objetivo a separação de objetos (ou observações) distintos, a classificação preocupa-se com a alocação de objetos (ou observações) a grupos previamente definidos. Mas, conforme explicam, na análise dos dados reais essa distinção conceitual não é tão clara, pois ambos objetivos frequentemente se sobrepõem. Além disso, ao contrário das técnicas de agrupamento em que não há um conhecimento prévio da alocação dos objetos aos grupos (por serem técnicas não supervisionadas), na análise discriminante tal conhecimento existe para os objetos da amostra (por serem técnicas supervisionadas), justamente pelo fato de a variável resposta ser conhecida. Dentre os muitos tipos, Witten, Hastie, James e Tibshirani (2009) elencam alguns tipos de análise discriminante, tais como a regularizada, as mistas e as generalizações baseadas na separação em hiperplanos e as baseadas em *kernels*.

Em linhas gerais, o objetivo da análise discriminante é encontrar funções que melhor separem os grupos da variável resposta com base em combinações das variáveis explicativas. Por esse motivo, tais funções podem ser desde estritamente lineares até altamente flexíveis.

Para os conjuntos de dados da Partição 1 e 2 foram utilizadas a análise discriminante linear, a flexível (que permite funções mais amplas, tais como não-lineares, baseadas em *kernel* e aquelas que utilizam *splines*) e a análise discriminante quadrática.

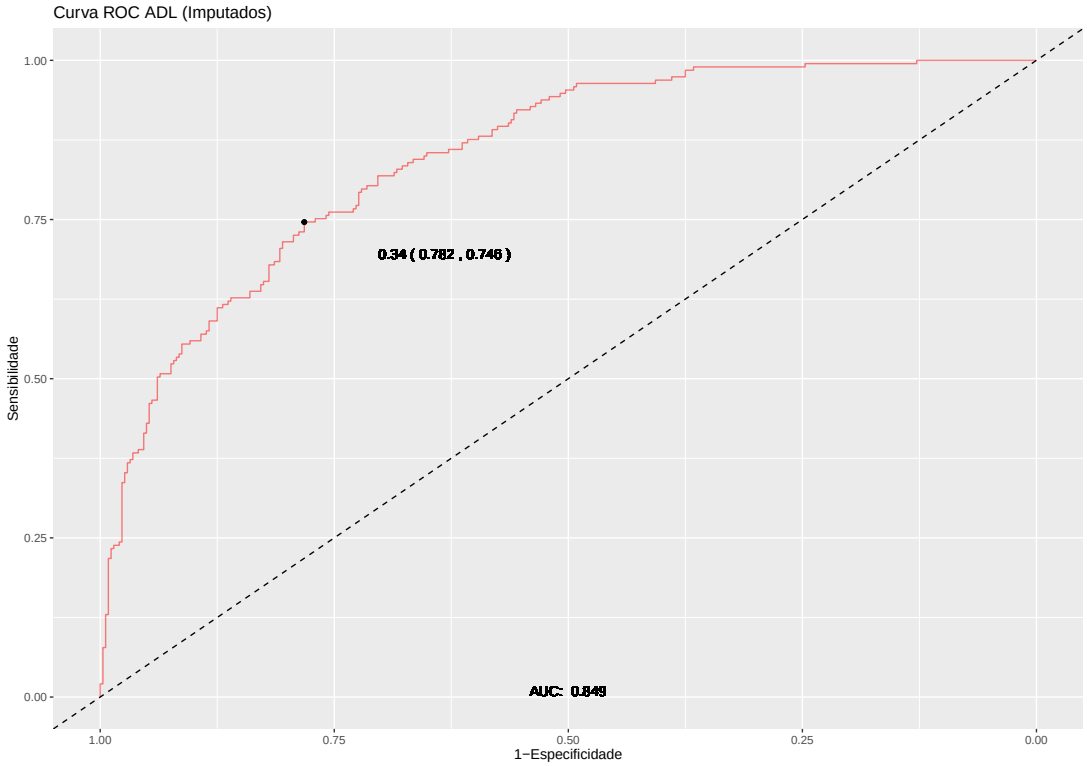
Análise Discriminante Linear

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	99	28
Yes	9	25

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.849.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

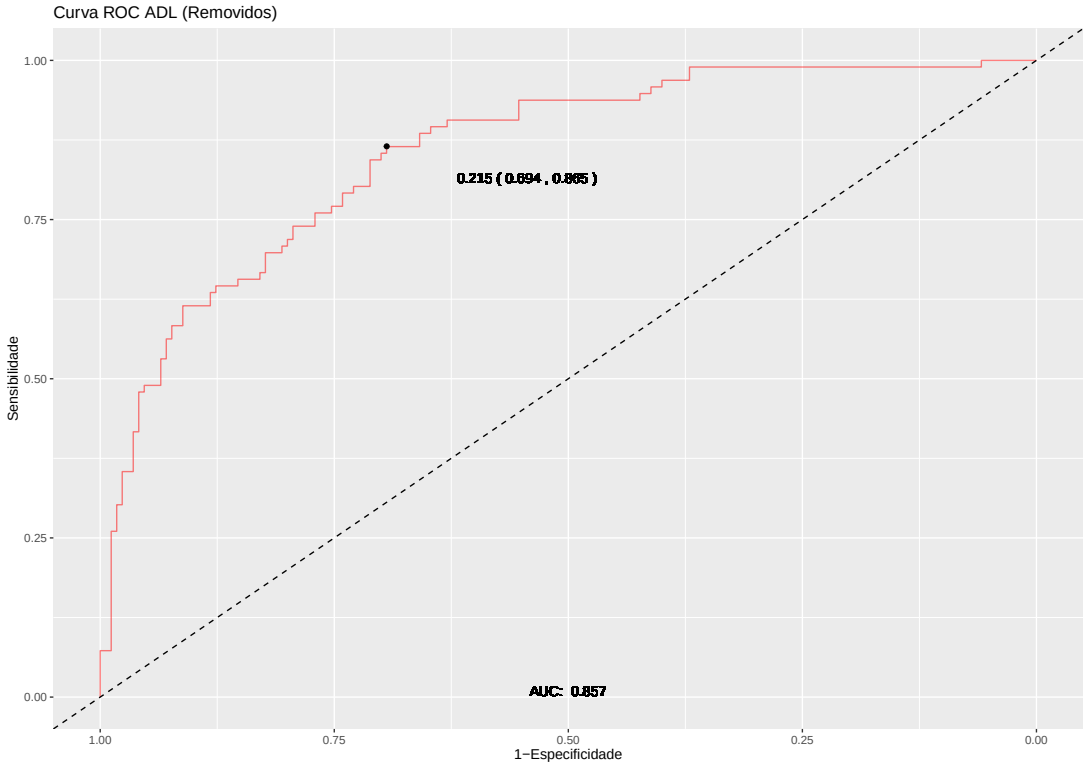
Desempenho do Modelo			
Accuracy	0.7702	**Sensitivity**	0.9167
95% CI	(0.6974, 0.8327)	**Specificity**	0.4717
No Information Rate	0.6708	**Pos Pred Value**	0.7795
P-Value [Acc >NIR]	0.003815	**Neg Pred Value**	0.7353
		Prevalence	0.6708
Kappa	0.4274	**Detection Rate**	0.6149
		Detection Prevalence	0.7888
Mcnemar's Test P-Value	0.003085	**Balanced Accuracy**	0.6942

(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	64	12
Yes	4	9

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.857.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.8202	**Sensitivity**	0.9412
95% CI	(0.7245, 0.8936)	**Specificity**	0.4286
No Information Rate	0.764	**Pos Pred Value**	0.8421
P-Value [Acc >NIR]	0.12910	**Neg Pred Value**	0.6923
		Prevalence	0.7640
Kappa	0.4258	**Detection Rate**	0.7191
		Detection Prevalence	0.8539
Mcnemar's Test P-Value	0.08012	**Balanced Accuracy**	0.6849

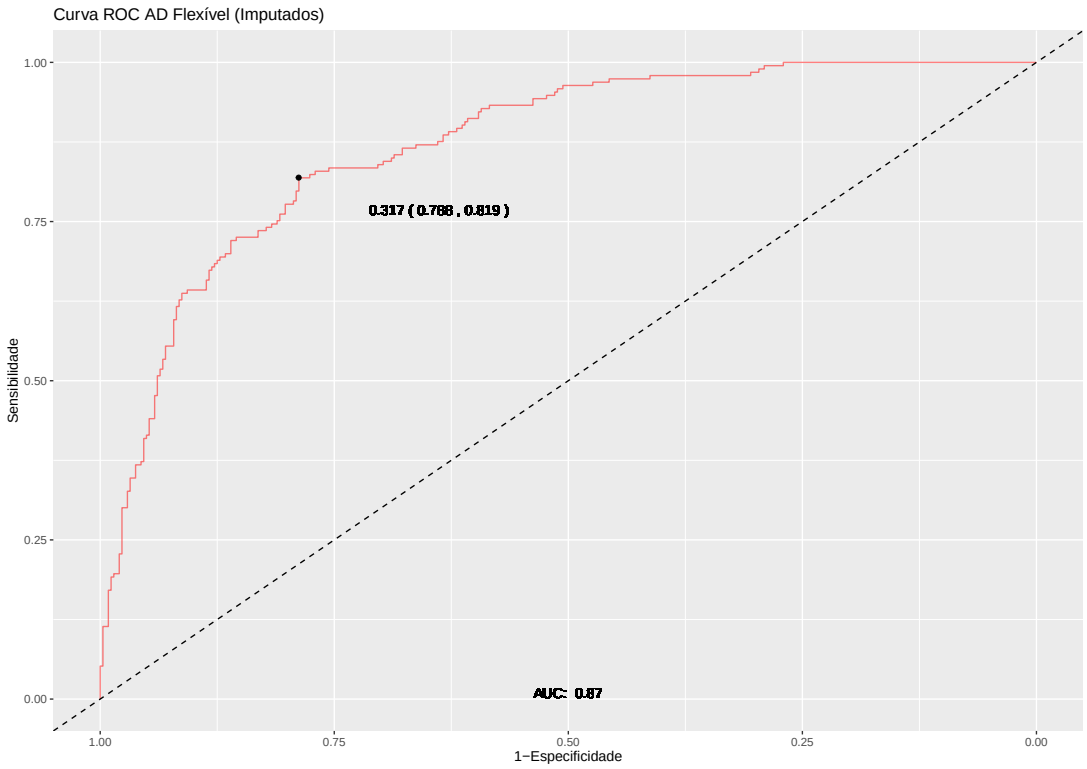
Análise Discriminante Flexível

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	95	25
Yes	13	28

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.857.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

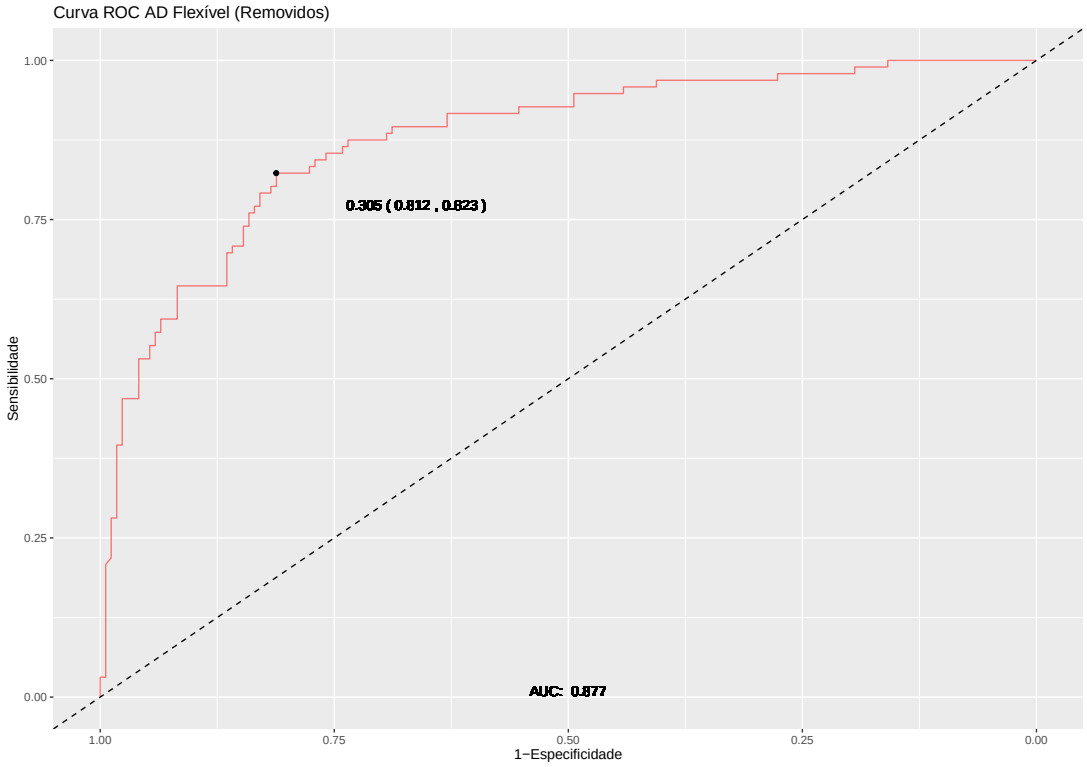
Desempenho do Modelo			
Accuracy	0.764	**Sensitivity**	0.8796
95% CI	(0.6907, 0.8272)	**Specificity**	0.5283
No Information Rate	0.6708	**Pos Pred Value**	0.7917
P-Value [Acc >NIR]	0.006397	**Neg Pred Value**	0.6829
		Prevalence	0.6708
Kappa	0.4329	**Detection Rate**	0.5901
		Detection Prevalence	0.7453
Mcnemar's Test P-Value	0.074353	**Balanced Accuracy**	0.7040

(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	61	11
Yes	7	10

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.877.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.7978	**Sensitivity**	0.8971
95% CI	(0.6993, 0.8755)	**Specificity**	0.4762
No Information Rate	0.764	**Pos Pred Value**	0.8472
P-Value [Acc >NIR]	0.2709	**Neg Pred Value**	0.5882
		Prevalence	0.7640
Kappa	0.3996	**Detection Rate**	0.6854
		Detection Prevalence	0.8090
Mcnemar's Test P-Value	0.4795	**Balanced Accuracy**	0.6866

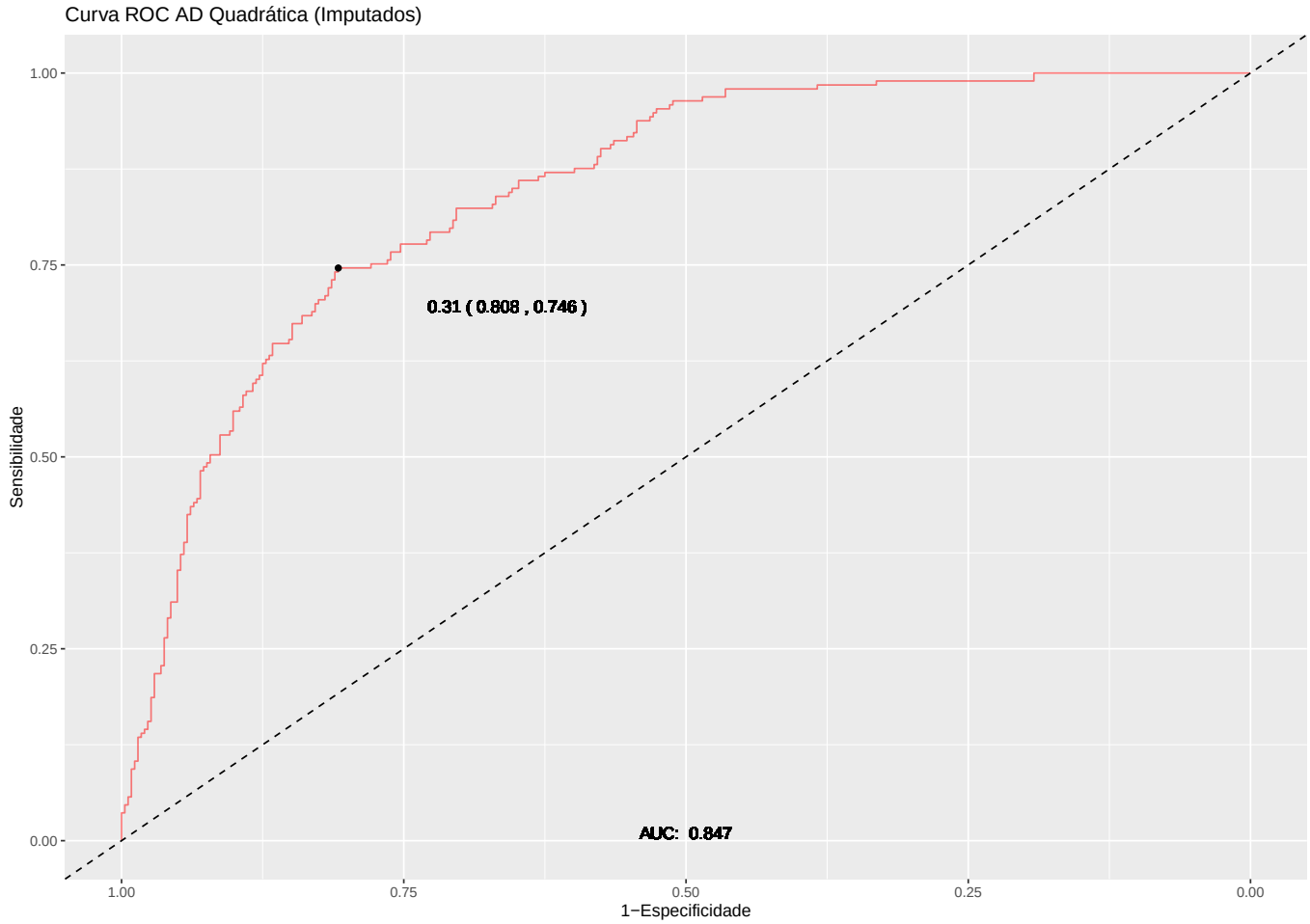
Análise Discriminante Quadrática

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	96	27
Yes	12	26

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.847.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

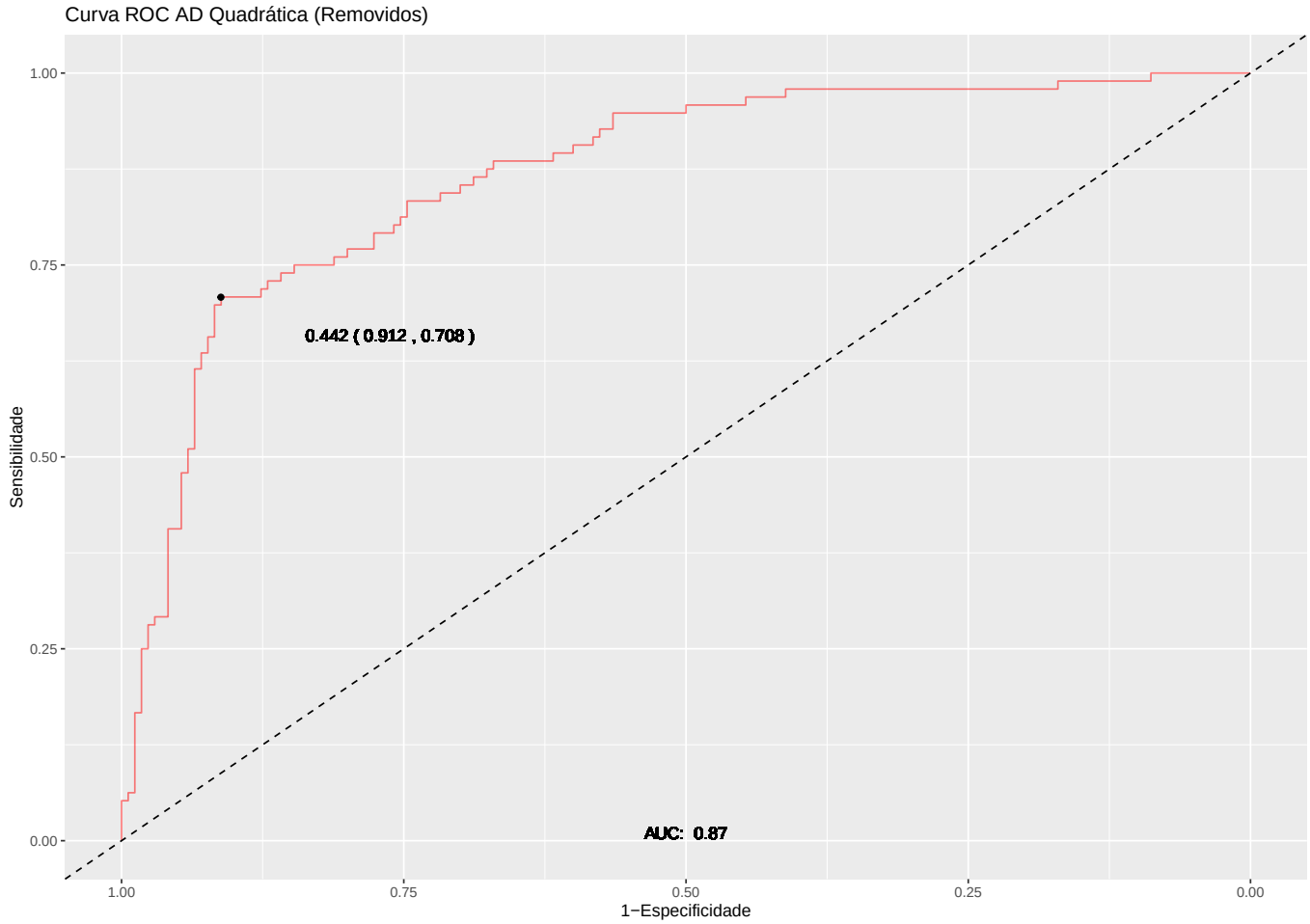
Desempenho do Modelo			
Accuracy	0.7578	**Sensitivity**	0.8889
95% CI	(0.6841, 0.8217)	**Specificity**	0.4906
No Information Rate	0.6708	**Pos Pred Value**	0.7805
P-Value [Acc >NIR]	0.01039	**Neg Pred Value**	0.6842
		Prevalence	0.6708
Kappa	0.4089	**Detection Rate**	0.5963
		Detection Prevalence	0.7640
Mcnemar's Test P-Value	0.02497	**Balanced Accuracy**	0.6897

(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	60	6
Yes	8	15

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.87.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.8427	**Sensitivity**	0.8824
95% CI	(0.7502, 0.9112)	**Specificity**	0.7143
No Information Rate	0.764	**Pos Pred Value**	0.9091
P-Value [Acc >NIR]	0.04778	**Neg Pred Value**	0.6522
		Prevalence	0.7640
Kappa	0.5776	**Detection Rate**	0.6742
		Detection Prevalence	0.7416
Mcnemar's Test P-Value	0.78927	**Balanced Accuracy**	0.7983

Regressão Logística

Em termos de regressão, quando se deseja modelar probabilidades p_i, a estimação pela abordagem da regressão linear normal naturalmente apresenta uma inconsistência. Enquanto a estimação de $w_0 + w_1\, x_{i1} + w_2\, x_{i2} + \cdots + w_p\, x_{ip}$ potencialmente possui um suporte infinito, por outro lado, p_i possui suporte finito no intervalo $[0, 1]$ pois se trata de uma probabilidade.

De maneira a resolver essa aparente inconsistência, uma simples transformação na regressão linear permite que a função tenha suporte compatível em ambos os lados. Essa transformação é a função `logit` que naturalmente conduz a um modelo de regressão logística:

$$\text{logit}[p_i] = \log\left(\frac{p_i}{1-p_i}\right) = w_0 + w_1\, x_{i1} + w_2\, x_{i2} + \cdots + w_p\, x_{ip} = \boldsymbol{w} \cdot \boldsymbol{x}^\intercal \tag{1}$$

Cabe mencionar que o argumento $p_i/(1-p_i)$ nada mais é do que uma chance (*odds*) e assume valores entre 0 e $+\infty$. Consequentemente, o $\log(p_i/[1-p_i])$ passa a assumir valores entre $-\infty$ e $+\infty$.

Assim, a função consiste na estimação do log da chance (*log-odds*), o que possui interpretação relativamente simples e intuitiva, além de ser um modelo linear generalizado[8], por isso o êxito na adoção desse método nos modelos de probabilidades preditivas até os dias atuais.

A Equação 1 também pode ser reescrita como:

$$p_i = \frac{e^{w_0 + w_1\, x_{i1} + w_2\, x_{i2} + \cdots + w_p\, x_{ip}}}{1 + e^{w_0 + w_1\, x_{i1} + w_2\, x_{i2} + \cdots + w_p\, x_{ip}}} = \frac{e^{\boldsymbol{w} \cdot \boldsymbol{x}^\intercal}}{1 + e^{\boldsymbol{w} \cdot \boldsymbol{x}^\intercal}}$$

Assim, recupera-se facilmente a probabilidade. O gráfico dessa função, com p_i no eixo das ordenadas e x_i no eixo da abcissas, é tipicamente representado pelo conhecido formato em "S".

[8] Nos modelos lineares generalizados, trata-se de um modelo de distribuição binomial de ligação do tipo `logit` e que é facilmente expansível a outros modelos com outras funções de ligação compatíveis, tais como o `probit`, o `Log-Log Complementar` e o `Log-Log Negativo`.

Além disso, exponenciando-se ambos os lados da Equação 1:

$$\frac{p_i}{1-p_i} = \exp\{w_0 + w_1\, x_{i1} + w_2\, x_{i2} + \cdots + w_p\, x_{ip}\} = \exp\{w_0\} \cdot \exp\{w_1\, x_{i1}\} \cdots \exp\{w_p\, x_{ip}\} \quad (2)$$

Vale dizer, a chance pode ser expressa como uma multiplicação das exponenciais de cada parcela $w_i\, x_i$, considerando-se também a parcela w_o correspondente ao intercepto do modelo linearizado.

Por fornecer probabilidades preditas, a regressão logística também passa a ser naturalmente considerada no contexto de classificação. Isto é, cria-se uma regra de decisão a partir da qual se classifica de uma determinada maneira se a probabilidade predita for superior à regra e se classifica de outra maneira caso contrário. Independentemente do critério que se utilize para se fixar o ponto de corte, o fato é que a regressão logística se mostra como um dos modelos mais simples e mais utilizados no contexto de classificação.

Como uma grande família de modelagem, a regressão logística permite desde o modelo mais simples até o modelo mais sofisticado com multiclasses (multinomial) e modelagem não-paramétrica.

Para os dados da Partição 1 e 2 foram utilizadas a regressão logística simples e a regressão logística regularizada (com estimação por LASSO, RIDGE e *Elastic-Net*).

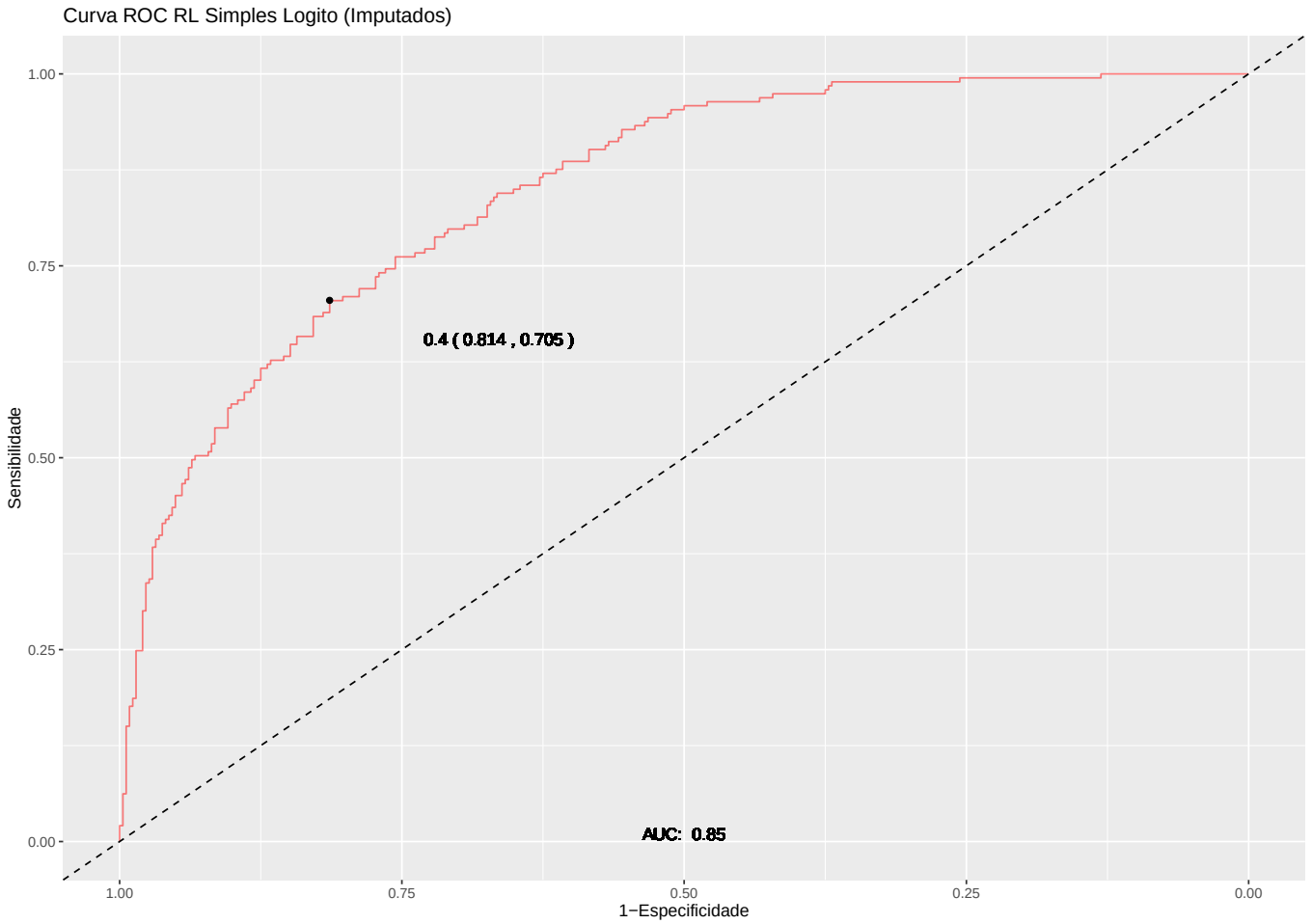
Regressão Logística Simples

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	99	27
Yes	9	26

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.85.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

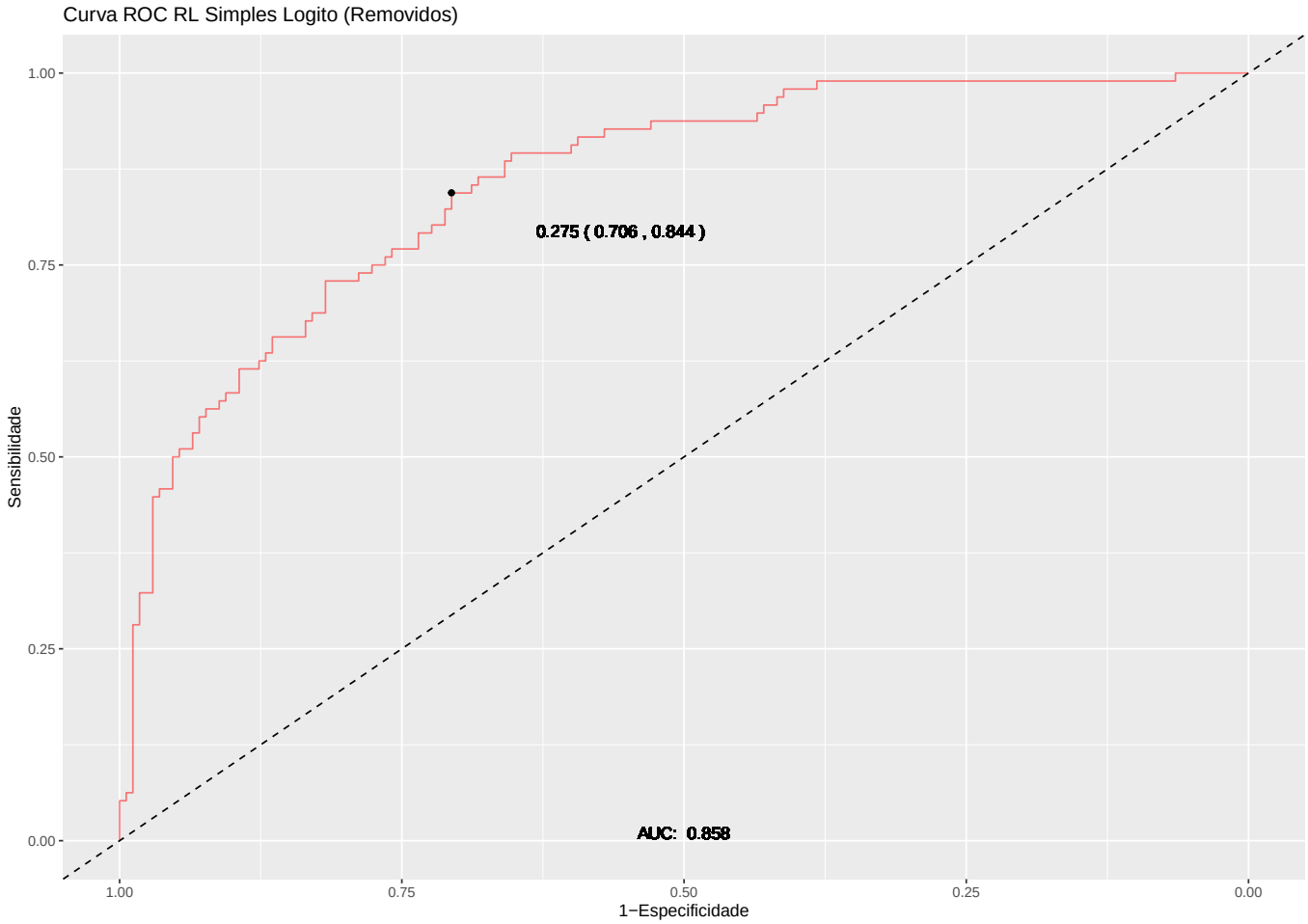
Desempenho do Modelo			
Accuracy	0.7764	**Sensitivity**	0.9167
95% CI	(0.7041, 0.8382)	**Specificity**	0.4906
No Information Rate	0.6708	**Pos Pred Value**	0.7857
P-Value [Acc >NIR]	0.002203	**Neg Pred Value**	0.7429
		Prevalence	0.6708
Kappa	0.4458	**Detection Rate**	0.6149
		Detection Prevalence	0.7826
Mcnemar's Test P-Value	0.004607	**Balanced Accuracy**	0.7036

(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	64	11
Yes	4	10

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.859.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.8315	**Sensitivity**	0.9412
95% CI	(0.7373, 0.9025)	**Specificity**	0.4762
No Information Rate	0.764	**Pos Pred Value**	0.8533
P-Value [Acc >NIR]	0.08127	**Neg Pred Value**	0.7143
		Prevalence	0.7640
Kappa	0.4717	**Detection Rate**	0.7191
		Detection Prevalence	0.8427
Mcnemar's Test P-Value	0.12134	**Balanced Accuracy**	0.7087

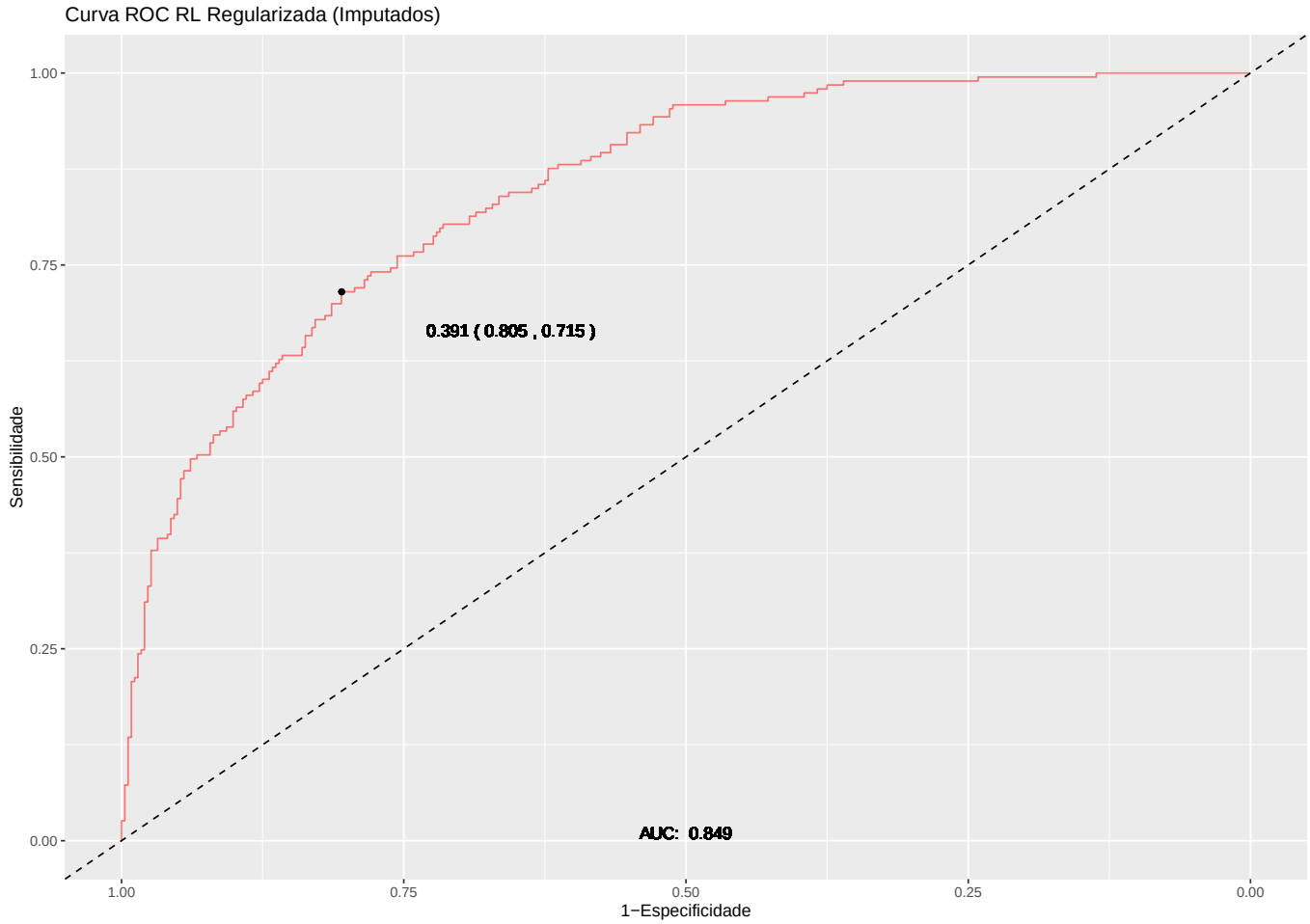
Regressão Logística Regularizada

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	99	28
Yes	9	25

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.849.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

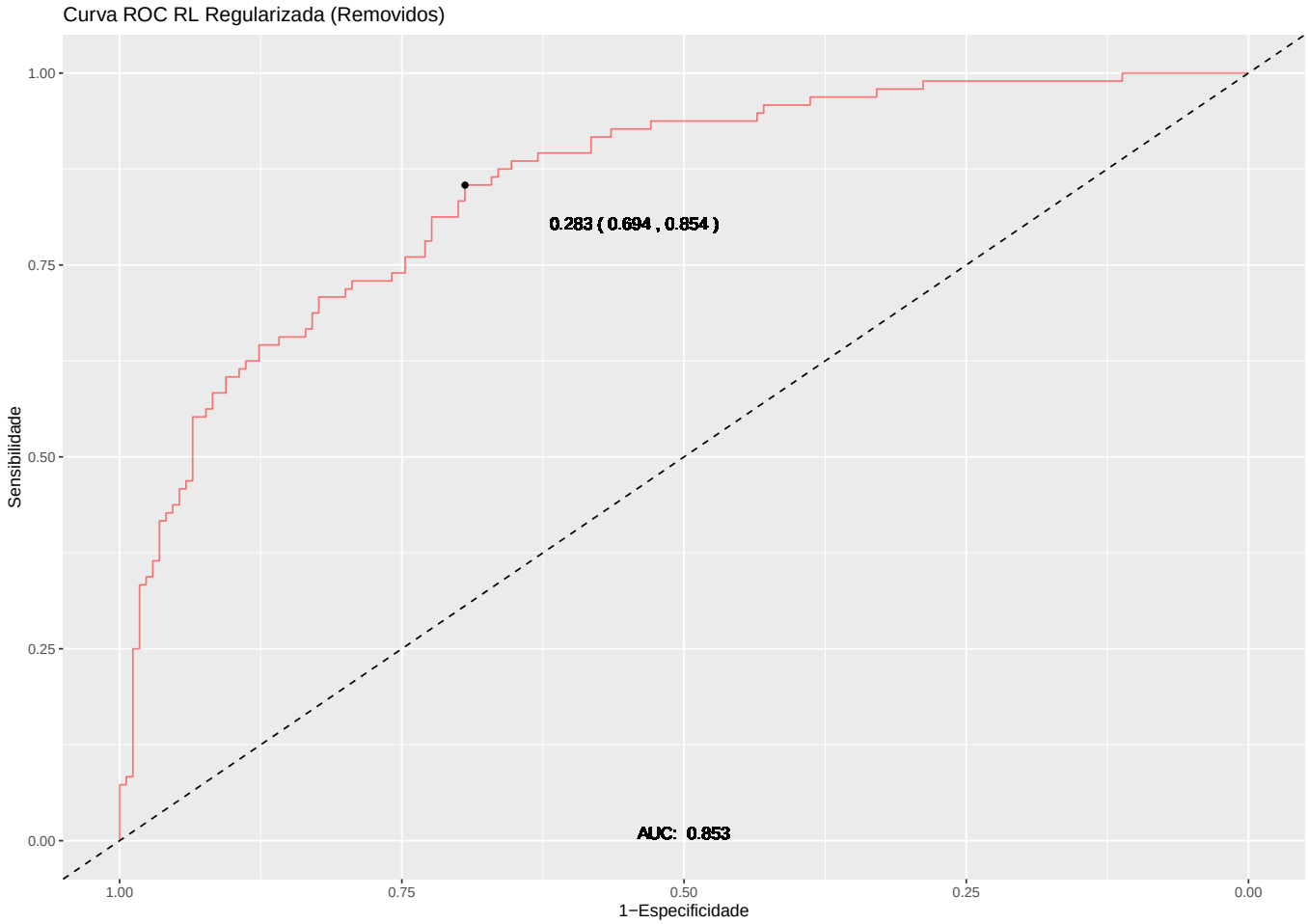
Desempenho do Modelo			
Accuracy	0.7702	**Sensitivity**	0.9167
95% CI	(0.6974, 0.8327)	**Specificity**	0.4717
No Information Rate	0.6708	**Pos Pred Value**	0.7795
P-Value [Acc >NIR]	0.003815	**Neg Pred Value**	0.7353
		Prevalence	0.6708
Kappa	0.4274	**Detection Rate**	0.6149
		Detection Prevalence	0.7888
Mcnemar's Test P-Value	0.003085	**Balanced Accuracy**	0.6942

(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	64	11
Yes	4	10

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.853.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.8315	**Sensitivity**	0.9412
95% CI	(0.7373, 0.9025)	**Specificity**	0.4762
No Information Rate	0.764	**Pos Pred Value**	0.8533
P-Value [Acc >NIR]	0.08127	**Neg Pred Value**	0.7143
		Prevalence	0.7640
Kappa	0.4717	**Detection Rate**	0.7191
		Detection Prevalence	0.8427
Mcnemar's Test P-Value	0.12134	**Balanced Accuracy**	0.7087

Florestas Aleatórias

Os métodos de árvores são simples e mais fáceis de interpretar, normalmente eles não são tão competitivos em termos de precisão de predição, mas a combinação de um grande número de árvores pode resultar em melhorias na precisão, porém com alguma perda em questão de interpretação. Esses métodos podem ser usados para classificação ou regressão [13]

Árvores de decisão são modelos de aprendizado supervisionado que representam regras de decisão baseadas nos valores dos atributos. Uma árvore de decisão utiliza uma estratégia de decompor um problema maior em subproblemas mais simples. A construção da árvore de decisão começa com a escolha de uma variável para compor o primeiro nó (pai/raiz), que dará origem aos primeiros ramos, os nós subsequentes da ávore são os nós descendentes (filho), Todos os nós que não possuem filhos são chamados de nós terminais ou folhas [10].

Em termos gerais, dois passo são necessário para construir uma árvore de Regressão, sendo o primeiro: selecionar possíveis espaços para as variáveis preditoras X_1, X_2, X_3, . . ., X_P, em regiões distintas R_1, R_2, R_3, . . .,R_j e que não se sobrepõe, por exemplo, preditora X_1 (anos de estudo), região $R_1(< 8$ anos) e região $R_2(\geq$ anos). E o segundo passo: para todas as observações que caírem dentro de uma região, a predição será a mesma, e a região pode simplesmente ser definida como a média dos valores de resposta para as observações de treinamento dentro da região R_j. As regiões são definidas com o objetivo de formar no espaço das preditoras retângulos de alta dimensão, ou caixas, para facilitar a interpretação do modelo preditivo resultante. O objetivo é encontrar as caixas ou regiõrs R_1, . . ., R_j que minimizem o erro do modelo [13].

Como é computacionalmente inviável dividir os espaços em j regiões para considerar todas as possíveis partições, o método de divisão binária recursiva é utilizado. Essa abordagem começa no topo da árvore e vai dividindo o espaço do preditor, cada divisão é representada por dois novos ramos mais abaixo na árvore, em cada etapa do processo de construção da árvore a melhor divisão é feita. Então o método não olha para passos subsequentes com o objetivo de obter uma árvore melhor em etapas futuras [13].

Na divisão binária recursiva o primeiro passo é selecionar uma variável preditora X_j, e um ponto de corte (S), de um modo que dividir o espaço do preditor nas regiões $(X \mid X_j < s)$ e $(X \mid X_j \geq s)$ leve ao menor erro possível. Então todas as variáveis preditoras são consideradas e todos os valores possíveis de ponto de corte também. O próximo passo seria selecionar uma variável preditora e seu respectivo

ponto de corte que resultem na menor soma dos resíduos quadrados (*residual sum of squares - RSS*) [13].

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}R_1)^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}R_2)^2$$

Onde $\hat{y}R_1$ é a média das respostas para as observações de treinamento na região R_1(j, s), e $\hat{y}R_2$ é a médias das respostas para as observações de treinamento na região R_2(j, s). E o processo é repetido dividindo as regiões que foram selecionadas anteriormente, até encontrar os pontos de corte que minimizem o RSS. Esse processo continua até encontrar um ponto de parada, um possível ponto de parada pode ser continuar o processo até que nenhuma região contenha mais do que um número mínimo observações [13].

Após a criação das regiões, a resposta para uma determinada observação do conjunto de teste, é dada através da média das observações de treino na região que a observação de teste pertence. Seguindo o exemplo anterior se o empregado possuir menos do que 8 anos de estudo o salário dele será predito como a média das observações de treinamento que se encontram naquela região (R_1) e caso tenha 8 anos ou mais de estudo, o salário dele será predito de acordo com a média das observações de treinamento da região R_2 [13].

A árvore pronta pode ser podada ao invés de utilizar um critério de parada, de forma que a árvore tenha o menor número possível de ramos que explique bem a predição. E uma abordagem semelhante ao Lasso pode ser utilizada, onde modelos mais complexos são penalizados, e então uma penalidade é adicionado a função objetivo, sendo um α multiplicado pelo número de ramos da árvore. Para encontrar um bom valor de α, pode-se utilizar uma validação cruzada.

Construir uma árvore de classificação é muito semelhante a construir uma árvore de regressão, a divisão binária recursiva também pode ser utilizada, a diferença é que a árvore de classificação é utilizada para predizer uma resposta qualitativa e não quantitativa. No entanto, o RSS não pode ser usado como um critério. Então, dois outros métodos são preferíveis, sendo eles o índice Gini e a entropia [13].

$$G = \sum_{k=1}^{K} \hat{p}_{mk}\,(1 - \hat{p}_{mk}) \qquad D = \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}$$

Gini Entropia

O índice Gini é uma medida de variância total ao longo das classes, ele soma para cada uma das classes dentro de cada região a estimativa da variância. Os valores resultantes são próximos de 0

e 1, onde um pequeno valor indica que o nó contém predominantemente observações de uma única classe. A entropia tem resultados muito parecidos com o índice de Gini, está sempre próximo de 0 e 1, e pode ser interpretado da mesma forma. Esses dois métodos também podem ser utilizados para avaliar o erro no momento de podar a árvore [13].

As árvores de decisão sofrem de alta variância, ou seja, se os dados de treinamento forem divididos em duas partes aleatórias e duas árvores de decisão forem ajustadas, os resultados obtidos podem ser bem diferentes. E o *bagging* tem o objetivo de reduzir a variância através da predição de vários modelos separados usando diferentes sets dos dados de treinamento, para depois calcular a média dos resultados das predições. Ou seja, calcular $\hat{f}_1(x)$, $\hat{f}_2(x)$, ..., $\hat{f}_{\mathrm{B}}(x)$ usando B conjuntos de treinamento separados e posteriormente calcular a média deles, para obter um único modelo de aprendizagem estatística de baixa variância. Para isso o método de *bootstrap* é utilizado, onde várias amostras de um único conjunto de treinamento são obtidas [13].

As árvores do *bagging* geralmente são construídas com alta profundidade e não precisam de poda, já que as podas tem a intenção de reduzir a variância e o *over-fitting*, e o próprio *bagging* já está lidando com esses problemas.

$$\hat{f}_{\mathrm{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

Para um problema de classificação tem que se obter uma votação majoritária entre as classes previstas por cada árvore.

Random forest é uma modificação de *bagging*, ele constrói uma grande coleção de árvores não correlacionadas. Isso porque se houver uma variável preditora forte, pelo método de *bagging* essa variável sempre será escolhida, e todas as árvores serão sempre muito semelhantes umas às outras, sendo altamente correlacionadas. A média de valores altamente correlacionados, não leva a uma redução tão grande na variância quanto calcular a média de valores não correlacionados [13].

No *random forest* uma nova amostra de m preditoras é tomada em cada divisão e, normalmente, o número de variáveis preditoras consideradas em cada divisão é $\approx \sqrt{p}$. Se o *random forest* for feito usando um $m = p$, ele será simplesmente um *bagging*, utilizando todas as variáveis preditoras para a construção da árvore [13].

Embora a coleção de árvores seja mais difícil de interpretar do que uma única árvore, um resumo geral da importância de cada variável preditora pode ser obtido usando o RSS (para árvores de regressão) ou o índice de Gini (para árvores de classificação). Para a árvore de regressão, isso é

obtido através do registro da diminuição do erro ao adicionar uma etapa na criação da árvore, ou seja, a cada divisão binária e adição de uma variável preditora. Um valor alto de diminuição do erro indica um preditor importante [13].

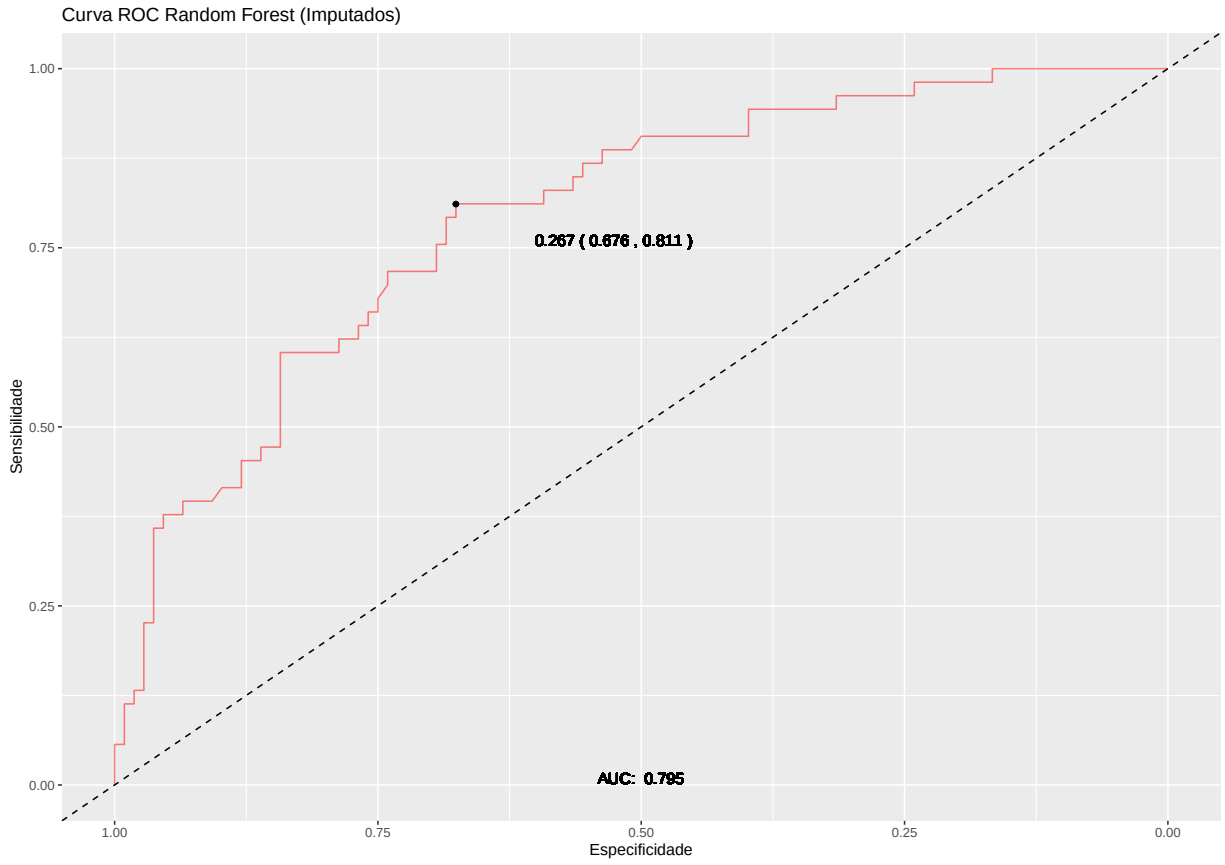
O *Random Forest* foi performado para os dados da Partição 1 e 2.

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	93	28
Yes	15	25

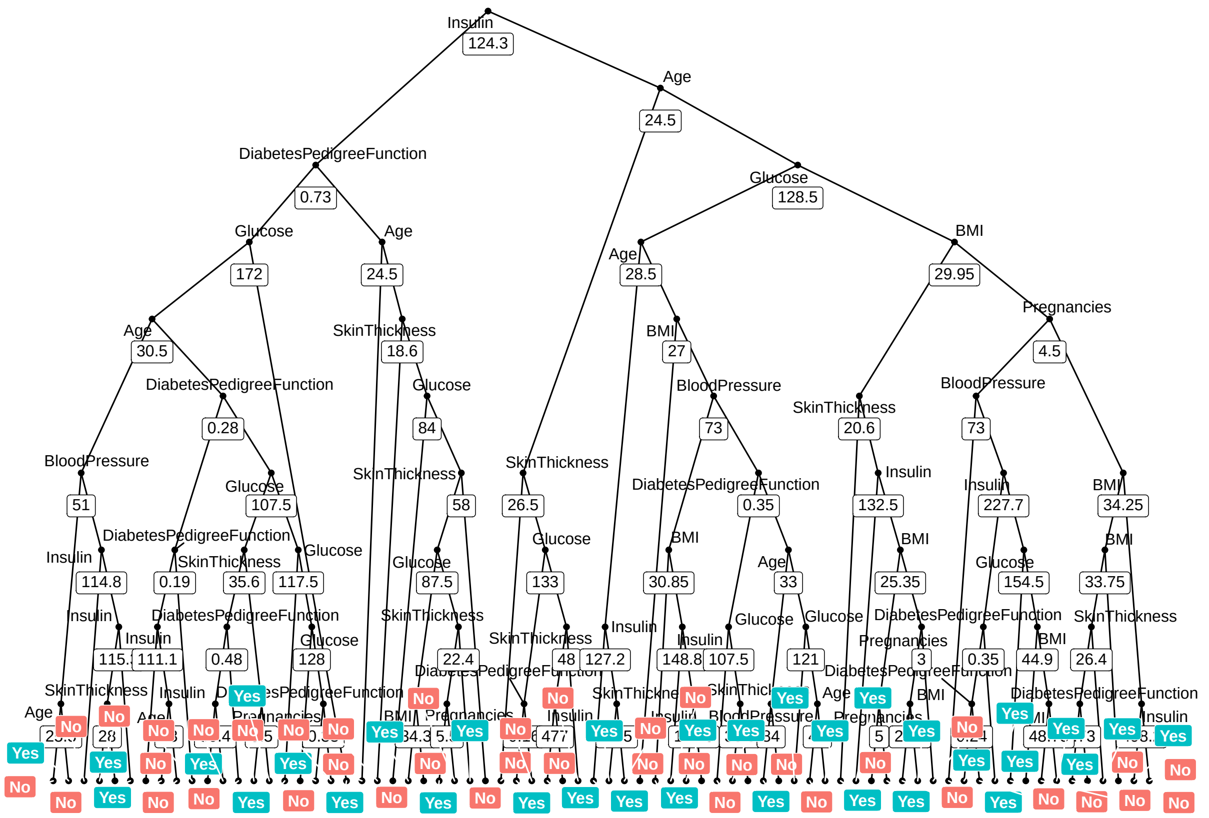
O desempenho na curva ROC é mostrado no gráfico seguinte. E o AUC foi de 0.795.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.7329	Sensitivity	0.8611
95% CI	(0.6576, 0.7995)	Specificity	0.4717
No Information Rate	0.6708	Pos Pred Value	0.7686
P-Value [Acc >NIR]	0.05368	Neg Pred Value	0.6250
		Prevalence	0.6708
Kappa	0.355	Detection Rate	0.5776
		Detection Prevalence	0.7516
Mcnemar's Test P-Value	0.06725	Balanced Accuracy	0.6664

A seguir, temos uma ilustração da menor árvore de decisão gerada pelo modelo utilizando os dados com imputação para valores *missing*. Por meio dessa árvore verifica-se um grande número de nós gerados a partir do conjunto de dados imputados.

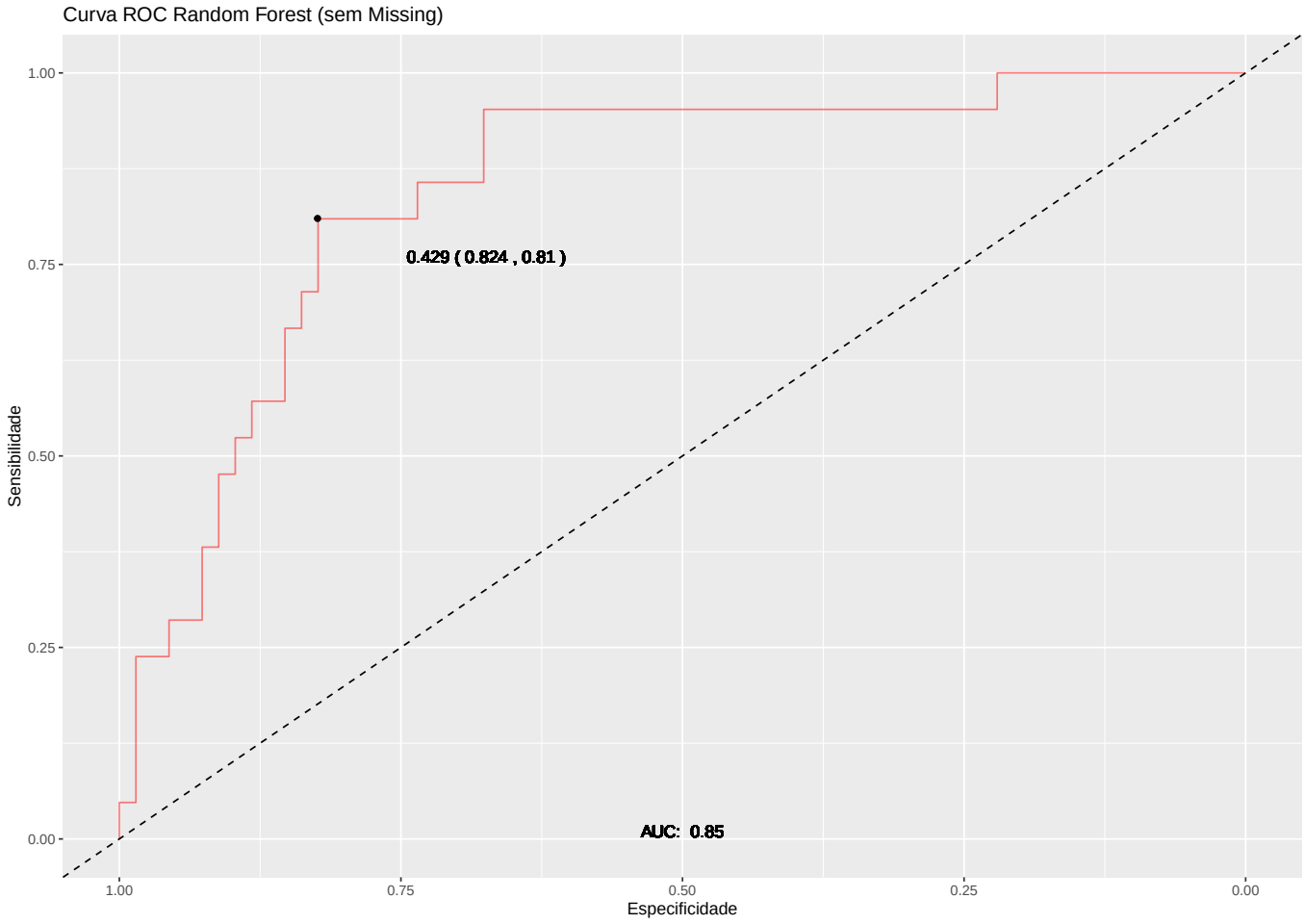


(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	59	9
Yes	9	12

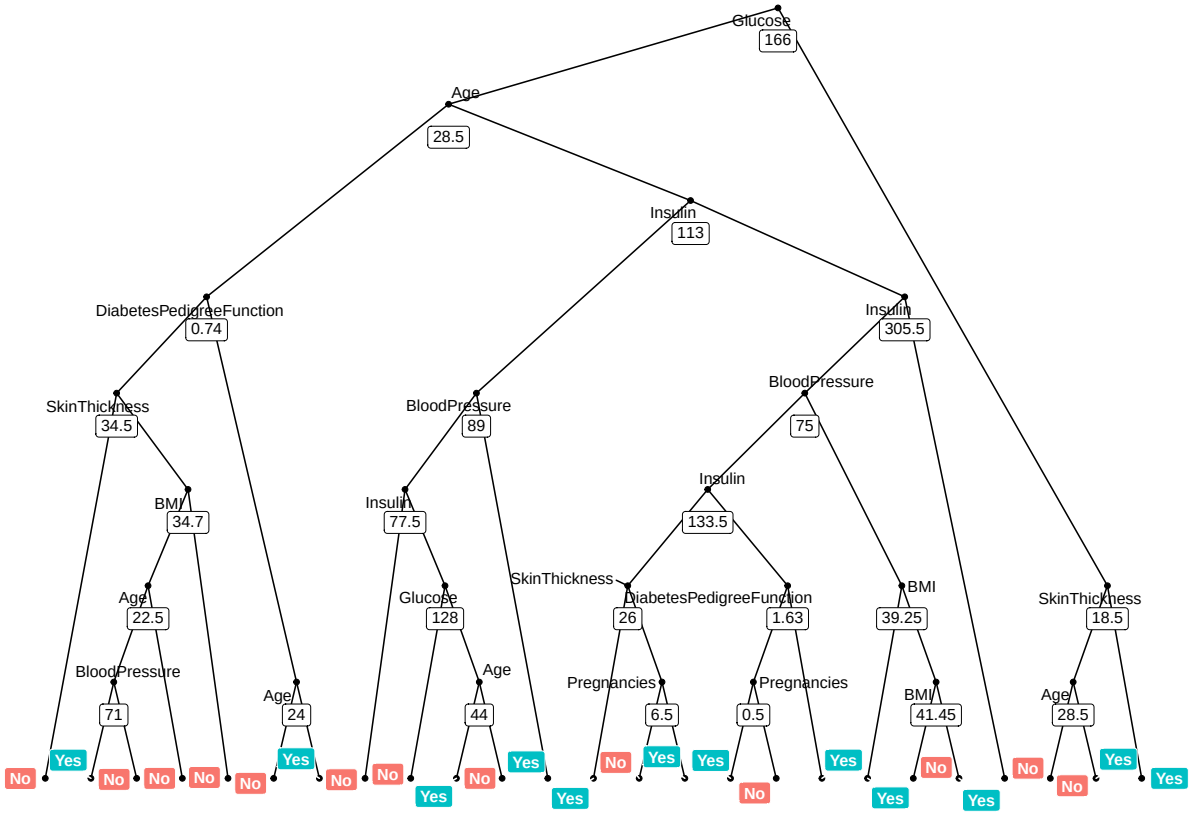
O desempenho na curva ROC é mostrado no gráfico seguinte. E o AUC foi de 0.85.



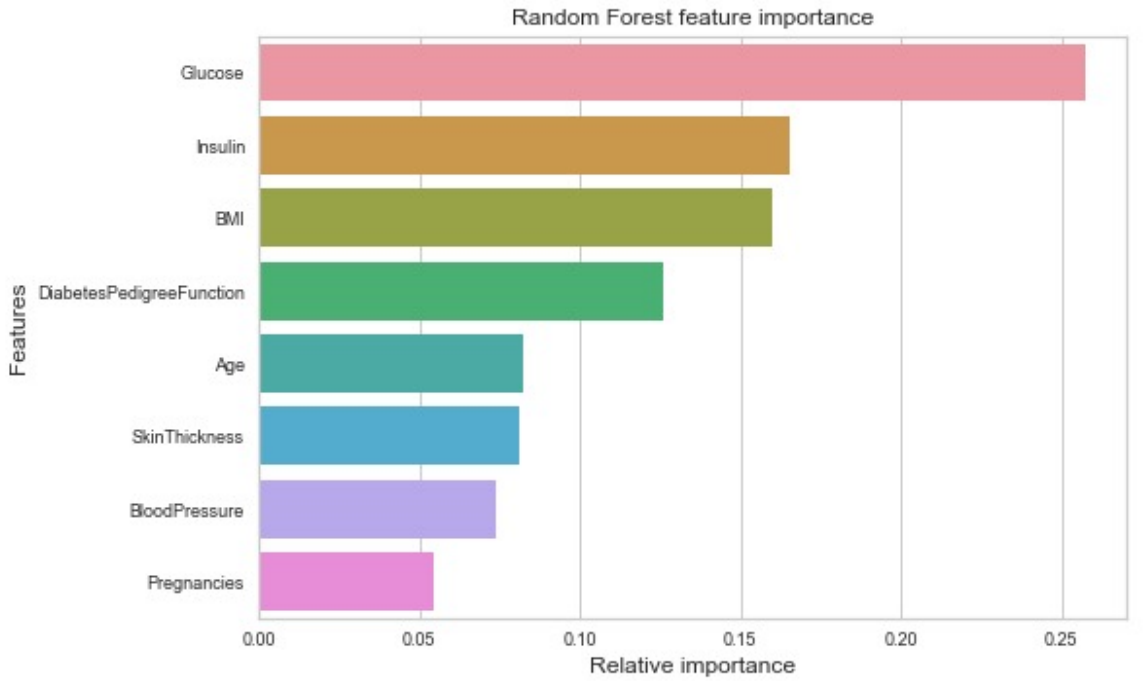
Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.7978	Sensitivity	0.8676
95% CI	(0.6993, 0.8755)	Specificity	0.5714
No Information Rate	0.764	Pos Pred Value	0.8676
P-Value [Acc >NIR]	0.2709	Neg Pred Value	0.5714
		Prevalence	0.7640
Kappa	0.4391	Detection Rate	0.6629
		Detection Prevalence	0.7640
Mcnemar's Test P-Value	1.0000	Balanced Accuracy	0.7195

A seguir, temos uma ilustração da menor árvore de decisão gerada pelo modelo utilizando os dados sem imputação. Em comparação com a menor árvore gerada pelo conjunto de dados imputados, foi possível identificar um menor número de nós da árvore no conjunto com os *missings* removidos.



Adicionalmente, é possível verificar a contribuição de cada uma das variáveis preditoras para a classificação de pacientes com diabetes. De acordo com a imagem abaixo, os níveis de glicose, insulina e índice de massa corporal são as variáveis que mais contribuem para discriminar a classe de pacientes que possuem diabetes.

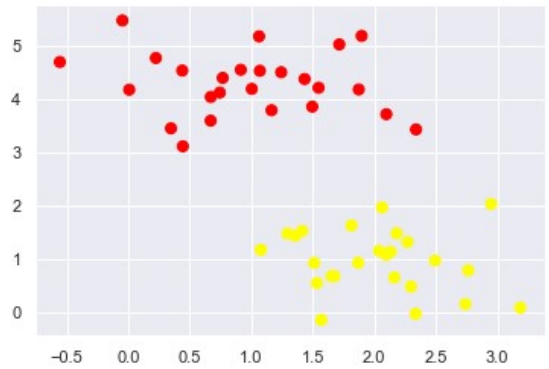


Maquinas de Vetores de Suporte

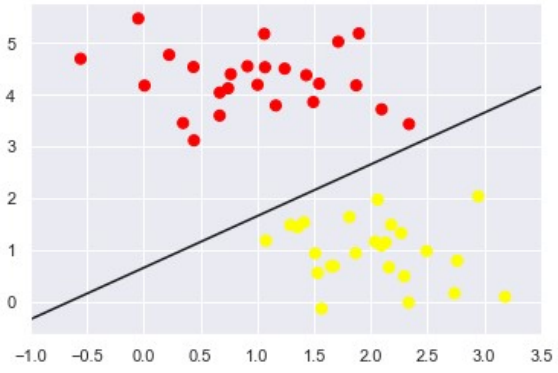
Máquinas de vetores de suporte (SVMs) são uma classe poderosa e flexível de algoritmos supervisionados para classificação e regressão. Nesse trabalho, vamos discutir a intuição por trás de seu uso em problemas de classificação [7].

Um SVM (Support Vector Machine) é um classificador discriminativo formalmente definido por um hiperplano de separação. Em outras palavras, se temos dados de treinamento rotulados (aprendizado supervisionado), o algoritmo gera um hiperplano ideal que categoriza novos dados inéditos. No espaço bidimensional, esse hiperplano é uma linha que divide um plano em duas partes, onde cada classe fica em cada lado.

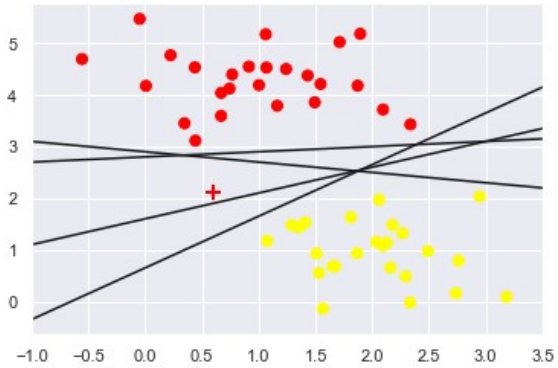
Suponha que tenhamos um gráfico de duas classes de etiquetas no gráfico, como mostra a imagem abaixo. É possível decidir uma linha de separação para as classes?



Qualquer ponto que fica à esquerda da linha cai na classe círculo preto e à direita cai na classe quadrado azul. Separação de classes, é isso que o SVM faz, ele descobre uma linha / hiperplano (no espaço multidimensional que separa as classes).

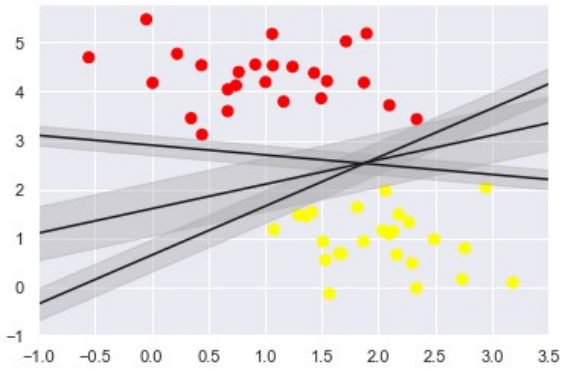


Mas podemos ter várias linhas, como mostrado abaixo, que podem separar os pontos de dados. Como escolhemos a melhor linha que separa o conjunto de dados em duas classes?

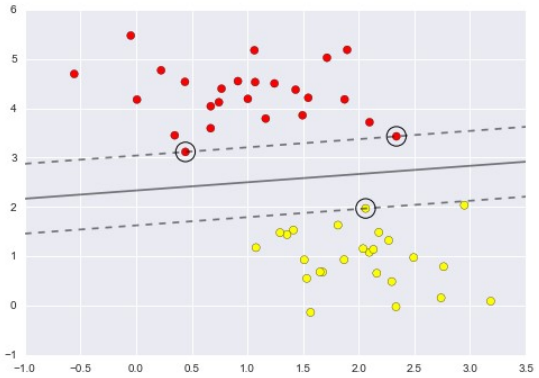


Temos quatro separadores muito diferentes que, no entanto, discriminam perfeitamente essas amostras. Dependendo da sua escolha, um novo ponto de dados (por exemplo, o marcado pelo "+"neste gráfico) receberá uma etiqueta diferente! Evidentemente, nossa simples intuição de "traçar uma linha entre as classes"não é suficiente.

Além disso, o SVM localiza o hiperplano para maximizar a *margem* entre os vetores de suporte das duas classes. Hiperplano são limites de decisão que classificam o conjunto de dados enquanto maximizam a margem. A intuição é a seguinte: em vez de simplesmente desenhar uma linha de largura zero entre as classes, podemos desenhar em torno de cada linha uma margem de alguma largura, até o ponto mais próximo.



Nas máquinas de vetores de suporte, a linha que maximiza essa margem é a que escolheremos como o modelo ideal. Máquinas de vetores de suporte são um exemplo desse estimador de margem máxima. Mas o que são vetores de suporte?



Os vetores de suporte são os pontos de dados no conjunto de dados mais próximos ao hiperplano, a remoção deles alterará o hiperplano que separa duas classes, portanto eles são elementos críticos do conjunto de dados, à medida que o SVM é construído sobre eles. A máquina de vetores de suporte tem dois objetivos principais, encontrar um hiperplano (linha) que separa linearmente os pontos de dados em duas classes e maximizar a margem entre vetores de suporte das duas classes.

Um conceito fundamental na construção de um SVM é a ideia do *Kernel*, aqui a álgebra linear desempenha um papel fundamental, nossos problemas estará associado as transformações de espaços vetoriais, etc. O algoritmo é implementado na prática usando um kernel. O aprendizado do hiperplano no SVM linear é feito transformando o problema usando alguma álgebra linear.

Assim o kernel é responsável por transformar os dados de entrada no formato necessário. Alguns dos kernels usados no SVM são lineares, polinomiais e radiais (função de base). Para criar um hiperplano não linear, usamos as funções RBF e Polinomial. Para aplicações complexas, deve-se usar kernels mais avançados para separar classes de natureza não linear. Com essa transformação, é possível obter classificadores precisos.

Segue alguns exemplos de Kernels SVM:

Kernel polinomial:

$k(x_i, x_j) = (x_i.x_j + 1)^d$

onde d é o grau do polinômio

Kernel gaussiano:

$$k(x_i, x_j) = e^{\frac{-\|(x-y)\|^2}{2\sigma^2}}$$

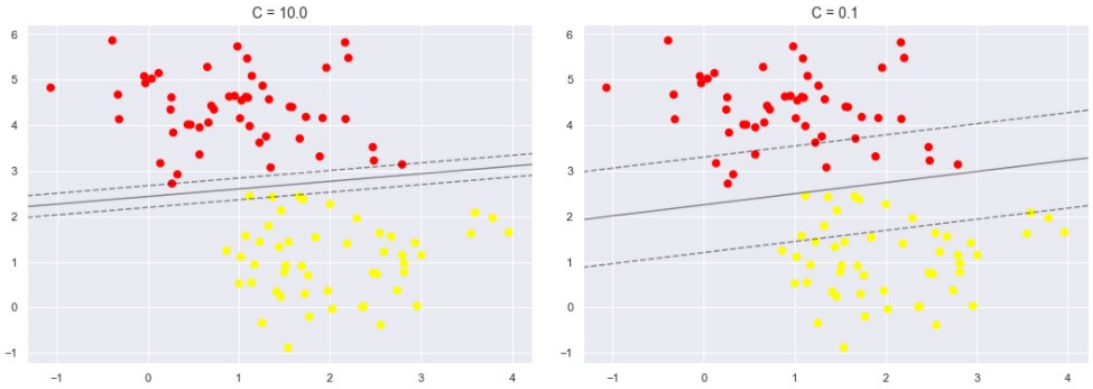
é um kernel de uso geral, usado quando não há conhecimento prévio sobre os dados.

Laplace RBF Kernel:

$$k(x_i, x_j) = e^{\frac{-\|(x-y)\|}{\sigma}}$$

também é usado quando não há conhecimento prévio sobre os dados.

Outro conceito essencial nos algoritmos do Support Vector Machine é o parâmetro de regularização, este nos ajuda a lidar com problemas de sobreposição. Basicamente a implementação do SVM possui um fator de correção que "suaviza"a margem, ou seja, permite que alguns dos pontos entrem na margem se isso permitir um melhor ajuste. A dureza da margem é controlada por um parâmetro de ajuste, mais conhecido como C. Podemos manter a regularização ajustando-a nos parâmetros C, este denota um parâmetro de penalidade que representa um erro ou qualquer forma de classificação incorreta. Com essa classificação incorreta, pode-se entender quanto do erro é realmente suportável. Com isso, você pode anular a compensação entre o termo não classificado e o limite da decisão. Com um valor C menor, obtemos hiperplano de pequena margem e com um valor C maior, obtemos hiperplano de maior valor. O gráfico mostrado abaixo fornece uma imagem visual de como um C em mudança O parâmetro afeta o ajuste final, através do suavização da margem:



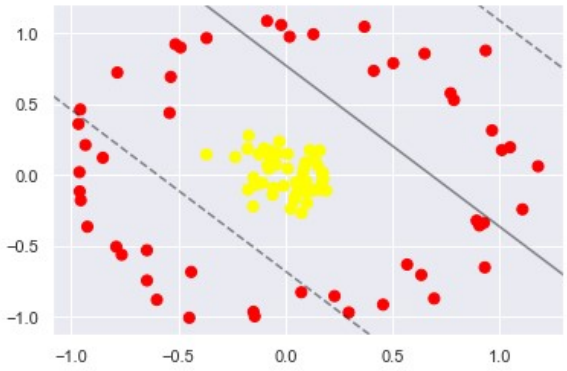
Para encontrarmos o valor ideal de C, o parâmetro dependerá do seu conjunto de dados e deve ser ajustado usando a validação cruzada ou um procedimento semelhante.

Por fim, vamos discutir quando as classes não são linearmente separáveis, como o SVM separa os dados?

Quando os dados não são linearmente separáveis, não podemos desenhar uma linha reta para separar as duas classes. Para resolver o problema, usamos dados não linearmente separáveis no espaço n-dimensional. Transforme-o em um espaço dimensional mais alto para torná-lo linearmente separável.

O SVM se torna extremamente poderoso quando ele é combinado com os kernels. Vimos os papéis dos kernels para SVM, projetamos nossos dados no espaço de maior dimensão, definido por polinômios e funções de base gaussiana, e assim conseguimos ajustar relações não lineares com um classificador linear.

Para motivar a necessidade de kernels, vejamos alguns dados que não são linearmente separáveis:

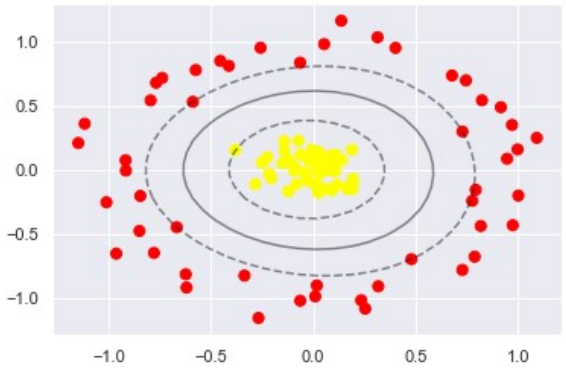


Os dados acima não são linearmente separáveis no espaço bidimensional, pois não há linha que possa separar as duas classes neste plano xy. Para contornar esse problema podemos aplicar uma transformação e adicionar mais uma dimensão, criando um eixo r, ou seja, projetar os dados para uma dimensão superior de tal forma que um separador linear seria suficiente.

Uma estratégia para esse fim é calcular uma função básica centralizada em todos os pontos do conjunto de dados e deixar o algoritmo SVM filtrar os resultados. Esse tipo de transformação de função básica é conhecido como transformação do kernel, pois é baseado em um relacionamento de similaridade (ou kernel) entre cada par de pontos.

Estas são funções que pegam espaço de entrada dimensional baixo e o transformam em um espaço dimensional superior, isto é, ele converte problema não separável em problema separável, essas funções são chamadas de kernels. Podemos aplicar SVM kernelizado simplesmente alterando nosso kernel

linear para um kernel RBF (função de base radial).



Usando o Support Vector Machine kernelizado, temos um limite de decisão não linear adequado. Essa estratégia de transformação do kernel é frequentemente usada no aprendizado de máquina para transformar métodos lineares em métodos não lineares, especialmente para modelos nos quais o truque do kernel pode ser usado.

Vantagens do SVM

É eficaz em espaços de alta dimensão. É eficaz nos casos em que o número de dimensões é maior que o número de amostras. Optimalidade garantida: devido à natureza da otimização convexa, a solução sempre será um mínimo global e não um mínimo local. O SVM pode ser usado para dados separáveis linearmente e também não linearmente separáveis. Dados separáveis linearmente são a margem fixa, enquanto dados separáveis não linearmente representam uma margem flexível. Os SVMs fornecem conformidade com os modelos de aprendizado semi-supervisionados. Pode ser usado em áreas onde os dados são rotulados e não rotulados. Requer apenas uma condição para o problema de minimização, conhecido como SVM Transdutivo. O Mapeamento de Recursos costumava ser bastante oneroso na complexidade computacional do desempenho geral do treinamento do modelo. No entanto, com a ajuda do Kernel Trick, o SVM pode executar o mapeamento de recursos usando um simples produto de ponto.

Desvantagens do SVM

O SVM é incapaz de manipular estruturas de texto. Isso leva à perda de informações seqüenciais e, portanto, a um desempenho pior. O SVM de baunilha não pode retornar o valor de confiança probabilístico que é semelhante à regressão logística. Isso não fornece muita explicação, pois a confiança na previsão é importante em várias aplicações. A escolha do kernel é talvez a maior limitação da máquina de vetores de suporte. Considerando tantos núcleos presentes, torna-se difícil escolher o caminho certo para os dados. Não tem um bom desempenho quando temos um grande conjunto de dados porque o tempo de treinamento necessário é maior.

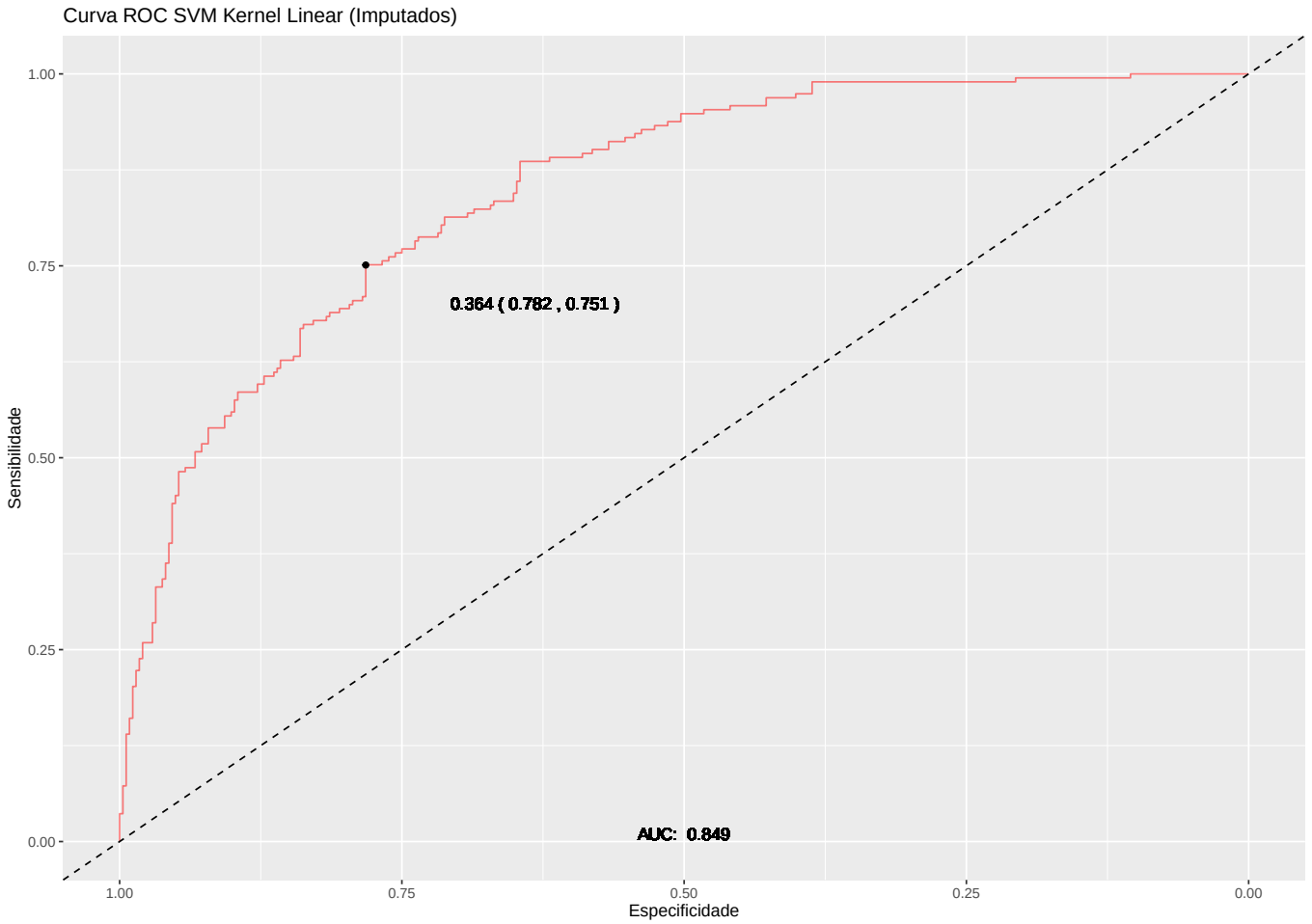
Maquinas de Vetores de Suporte com *Kernel* Linear

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	98	27
Yes	10	26

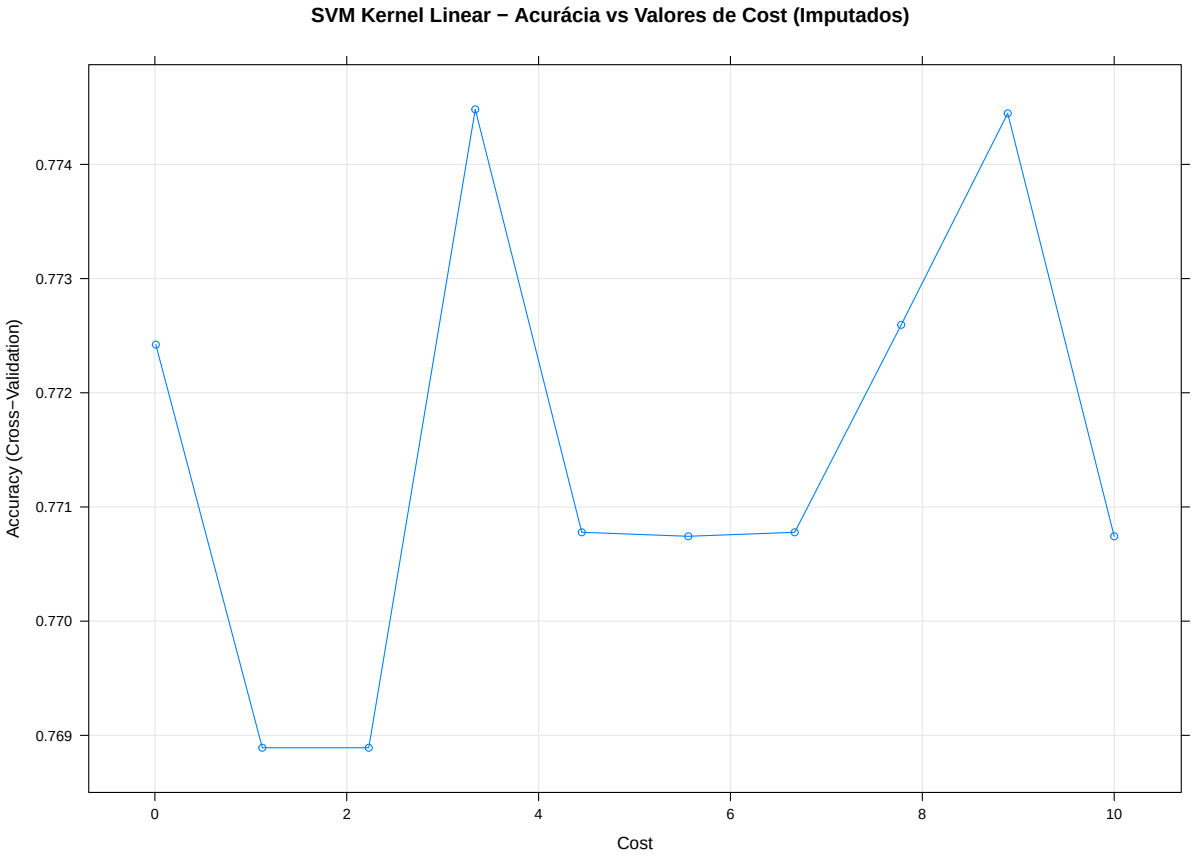
O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.849.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.7702	Sensitivity	0.9074
95% CI	(0.6974, 0.8327)	Specificity	0.4906
No Information Rate	0.6708	Pos Pred Value	0.7840
P-Value [Acc >NIR]	0.003815	Neg Pred Value	0.7222
		Prevalence	0.6708
Kappa	0.4334	Detection Rate	0.6087
		Detection Prevalence	0.7764
Mcnemar's Test P-Value	0.008529	Balanced Accuracy	0.6990

O gráfico a seguir mostra a acurácia em relação ao custo.

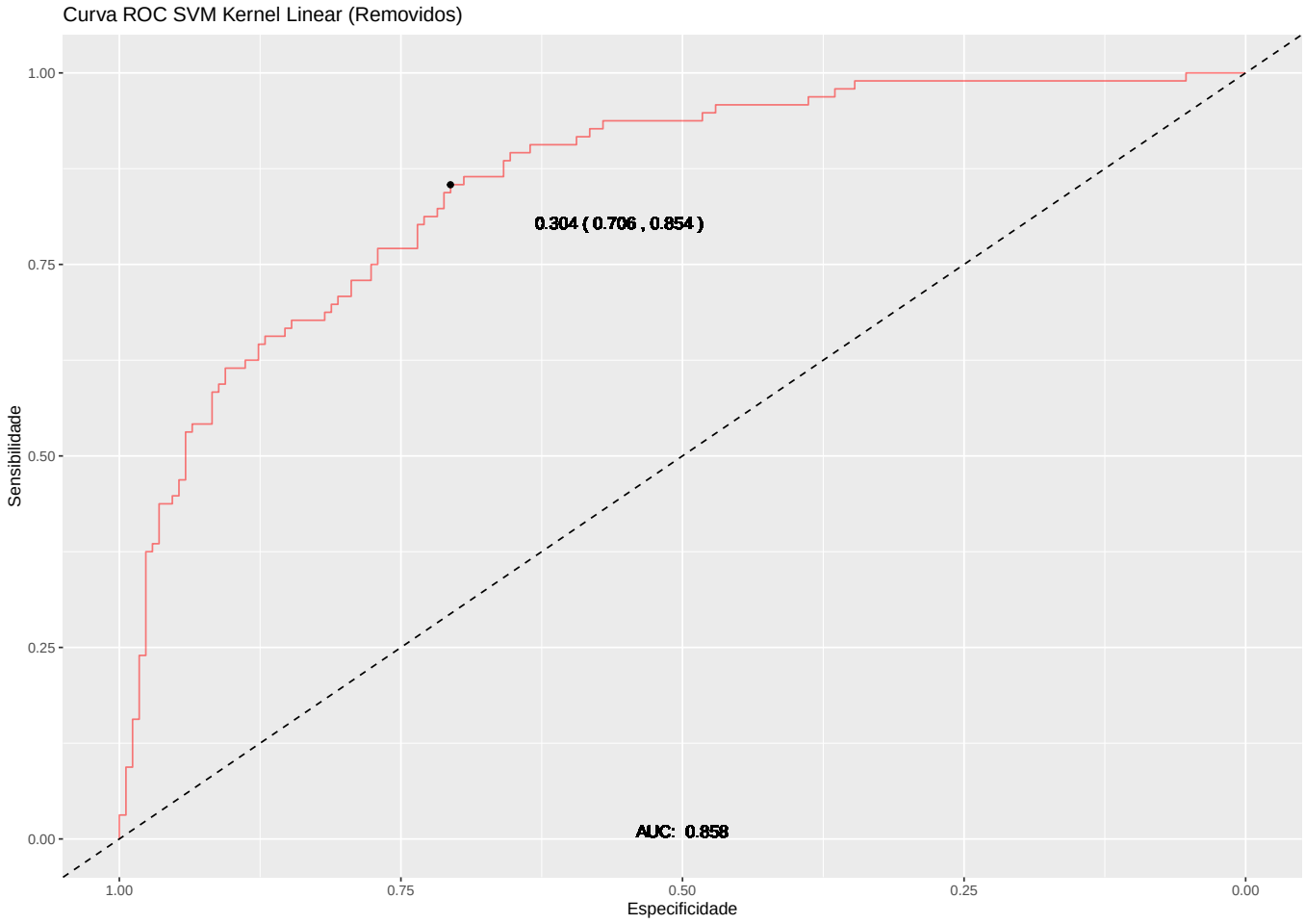


(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	64	12
Yes	4	9

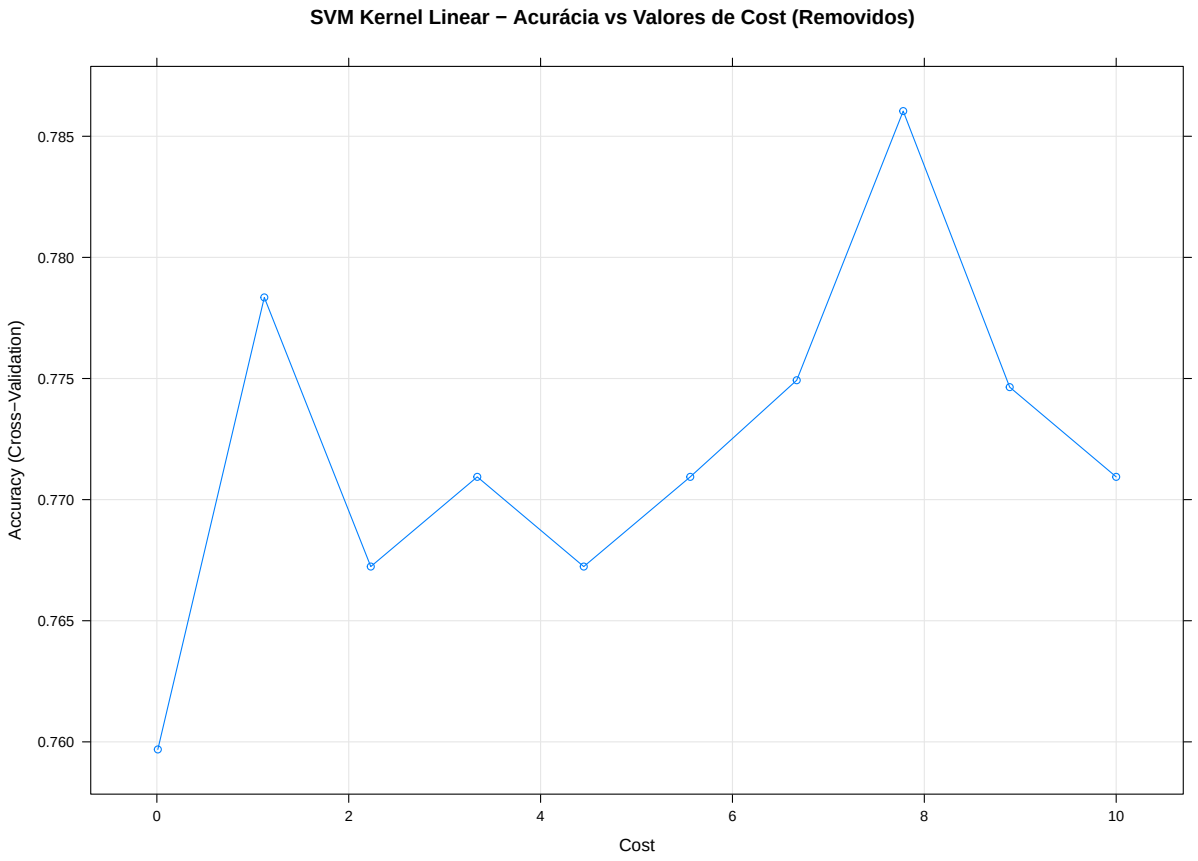
O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.858.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.8202	Sensitivity	0.9412
95% CI	(0.7245, 0.8936)	Specificity	0.4286
No Information Rate	0.764	Pos Pred Value	0.8421
P-Value [Acc >NIR]	0.12910	Neg Pred Value	0.6923
		Prevalence	0.7640
Kappa	0.4258	Detection Rate	0.7191
		Detection Prevalence	0.8539
Mcnemar's Test P-Value	0.08012	Balanced Accuracy	0.6849

O gráfico a seguir mostra a acurácia em relação ao custo.



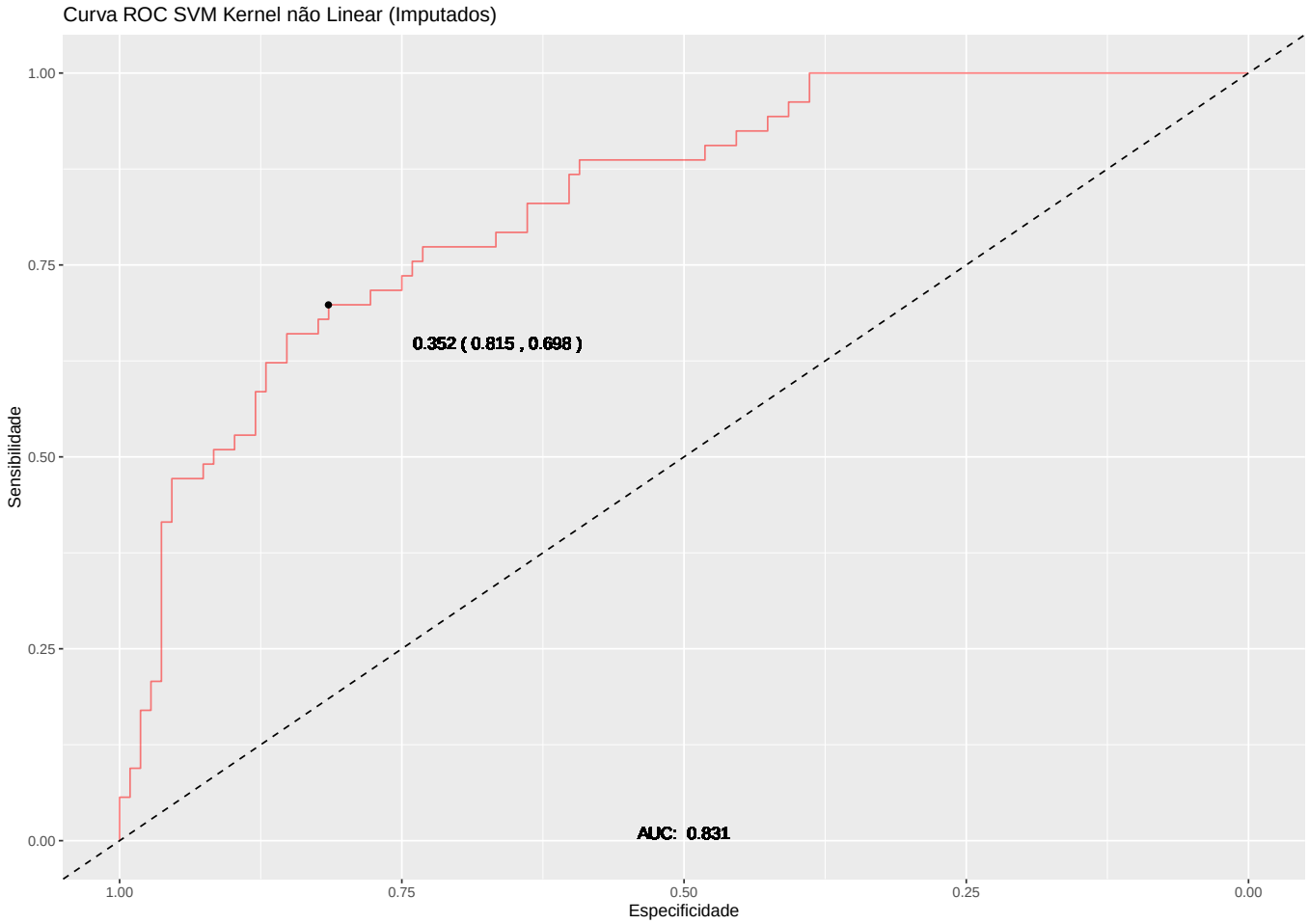
Maquinas de Vetores de Suporte com *Kernel* Não-Linear

(a) (Partição 1)

Para o conjunto de treinamento da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	99	26
Yes	9	27

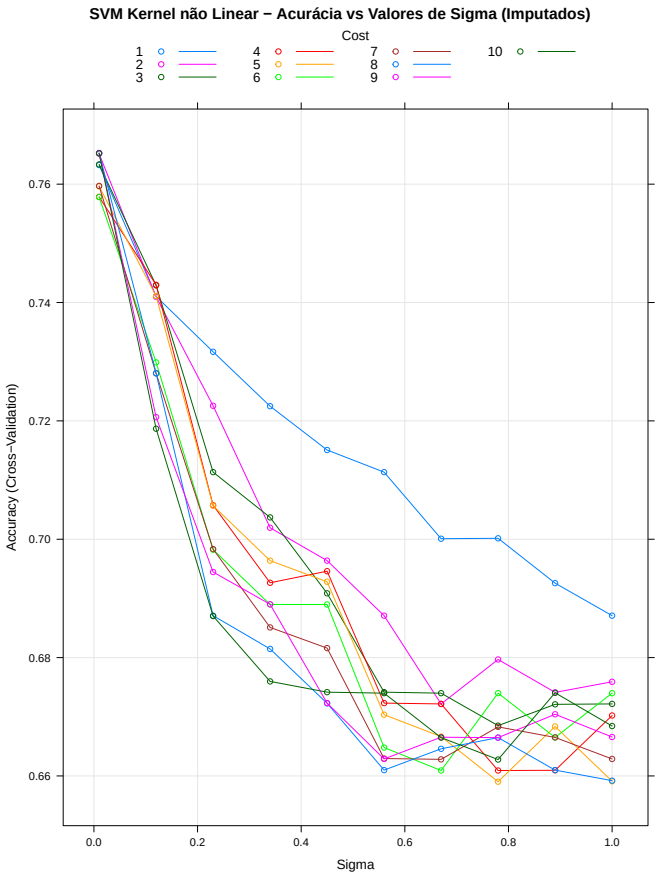
O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.831.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 1 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.7826	Sensitivity	0.9167
95% CI	(0.7109, 0.8437)	Specificity	0.5094
No Information Rate	0.6708	Pos Pred Value	0.7920
P-Value [Acc >NIR]	0.001230	Neg Pred Value	0.7500
		Prevalence	0.6708
Kappa	0.464	Detection Rate	0.6149
		Detection Prevalence	0.7764
Mcnemar's Test P-Value	0.006841	Balanced Accuracy	0.7131

O gráfico a seguir mostra a acurácia em relação aos valores de sigma.

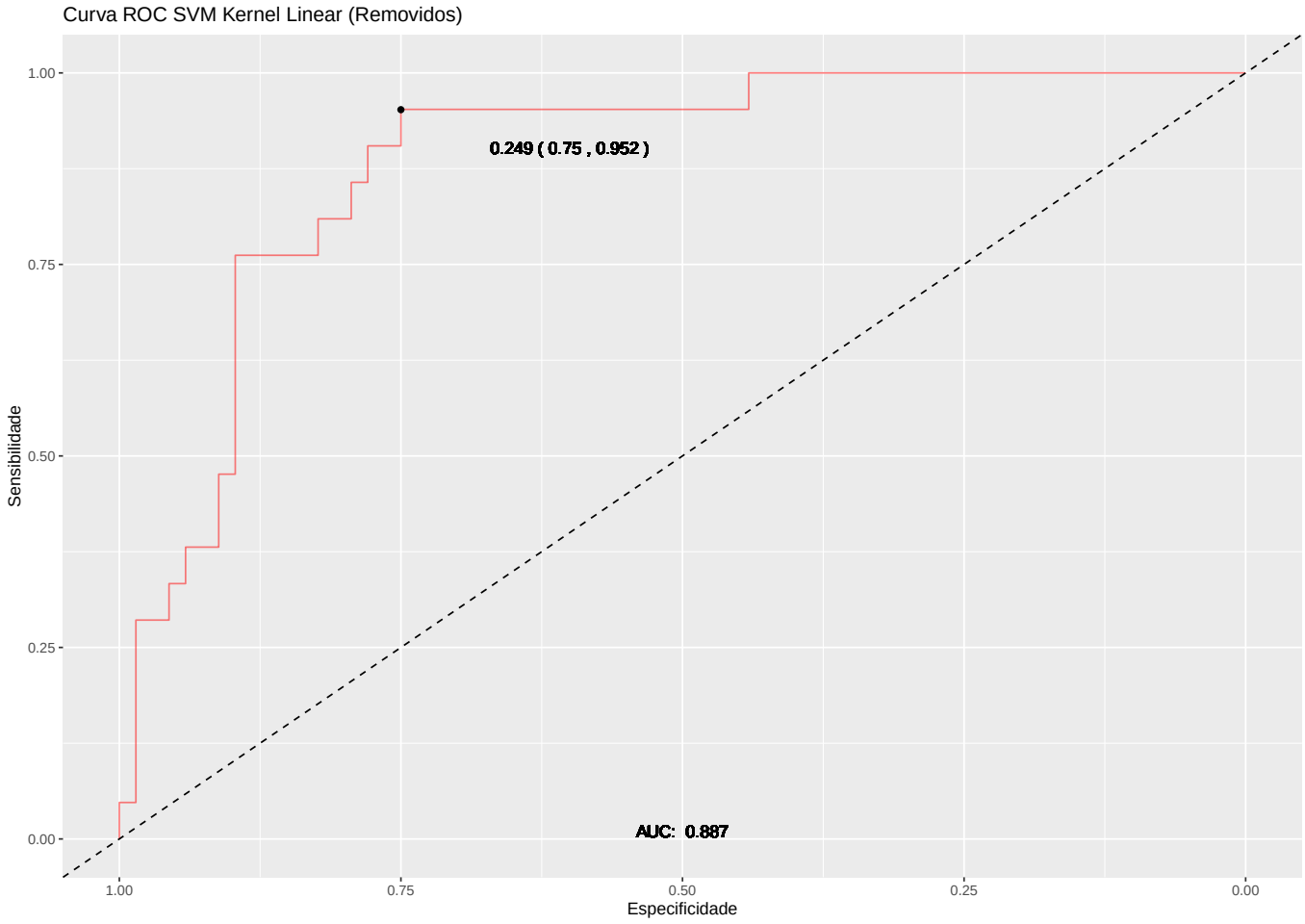


(b) (Partição 2)

Para o conjunto de treinamento da Partição 2, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	63	13
Yes	5	8

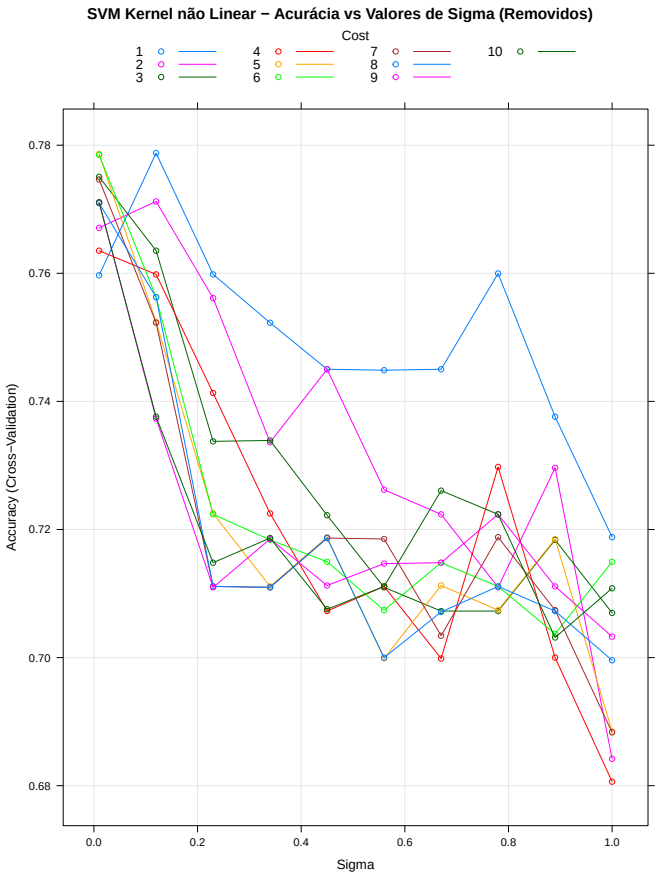
O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.887.



Finalmente o desempenho geral do modelo sobre o conjunto de validação da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.7978	Sensitivity	0.9265
95% CI	(0.6993, 0.8755)	Specificity	0.3810
No Information Rate	0.764	Pos Pred Value	0.8289
P-Value [Acc >NIR]	0.27088	Neg Pred Value	0.6154
		Prevalence	0.7640
Kappa	0.354	Detection Rate	0.7079
		Detection Prevalence	0.8539
Mcnemar's Test P-Value	0.09896	Balanced Accuracy	0.6537

O gráfico a seguir mostra a acurácia em relação aos valores de sigma.



Escolha dos Modelos e Desempenho Final

Para avaliação do melhor desempenho, todas as métricas foram inspecionadas, mas, apenas algumas foram priorizadas para compor a base de escolha final do modelo. Para o processo de decisão, foram considerados principalmente as métricas de acurácia, sensibilidade, especificidade e acurácia balanceada.

Todas as métricas calculadas em cada modelo avaliam, de certa maneira, o desempenho do modelo. Mas a acurácia, sensibilidade e especificidade, além de serem as mais usualmente utilizadas e conhecidas, mostraram-se capazes de sintetizar a performance geral. A escolha pela consideração adicional da acurácia balanceada foi devido à questão do desbalanceamento das categorias da variável resposta, conforme mostrado na análise descritiva. Como explicado, a acurácia balanceada corrige pelo desbalanceamento das classes.

Como a capacidade preditiva em todos os modelos foi baseada nas duas partições, a decisão sobre o melhor modelo também considerou cada partição. Isto é, para cada partição foi escolhido o modelo com melhor desempenho. Assim, cabe ao cliente final ou ao pesquisador a escolha do conjunto de dados mais adequado às suas finalidades específicas.

Assim, a Tabela 3 traz novamente o resultado das métricas para todos os modelos utilizados. Os destaques em vermelho referem-se aos valores mais elevados. Os destaques em azul referem-se aos valores mais baixos. As caselas brancas indicam os valores médios. A partir desta tabela é possível verificar que a acurácia média dos modelos foi superior no conjunto de dados removidos comparado ao conjunto de dados imputados.

Tabela 3: Comparativo de Desempenho

Modelo	Imputados				Removidos			
	Acuracia	Sensibilidade	Especificidade	Acurácia Balanceada	Acurácia	Sensibilidade	Especificidade	Acurácia Balanceada
AD Linear	0.77	0.91	0.47	0.69	0.82	0.94	0.43	0.68
AD Flexível	0.76	0.88	0.53	0.70	0.80	0.90	0.48	0.69
AD Quadrática	0.76	0.89	0.49	0.69	0.84	0.88	0.71	0.8
RL Simples	0.78	0.91	0.49	0.70	0.83	0.94	0.48	0.71
RL Regularizada	0.77	0.91	0.47	0.69	0.83	0.94	0.48	0.71
Random Forest	0.73	0.86	0.47	0.67	0.80	0.87	0.57	0.72
SVM Linear	0.77	0.91	0.49	0.7	0.82	0.94	0.43	0.68
SVM Ñ Linear	0.78	0.92	0.51	0.71	0.80	0.93	0.38	0.65

Como se verifica, para a o conjunto dos dados imputados (Partição 1), o SVM com *kernel* não-linear apresentou as melhores métricas em três dos quatro critérios. Por esse motivo, foi escolhido para ser finalmente avaliado no conjunto de teste.

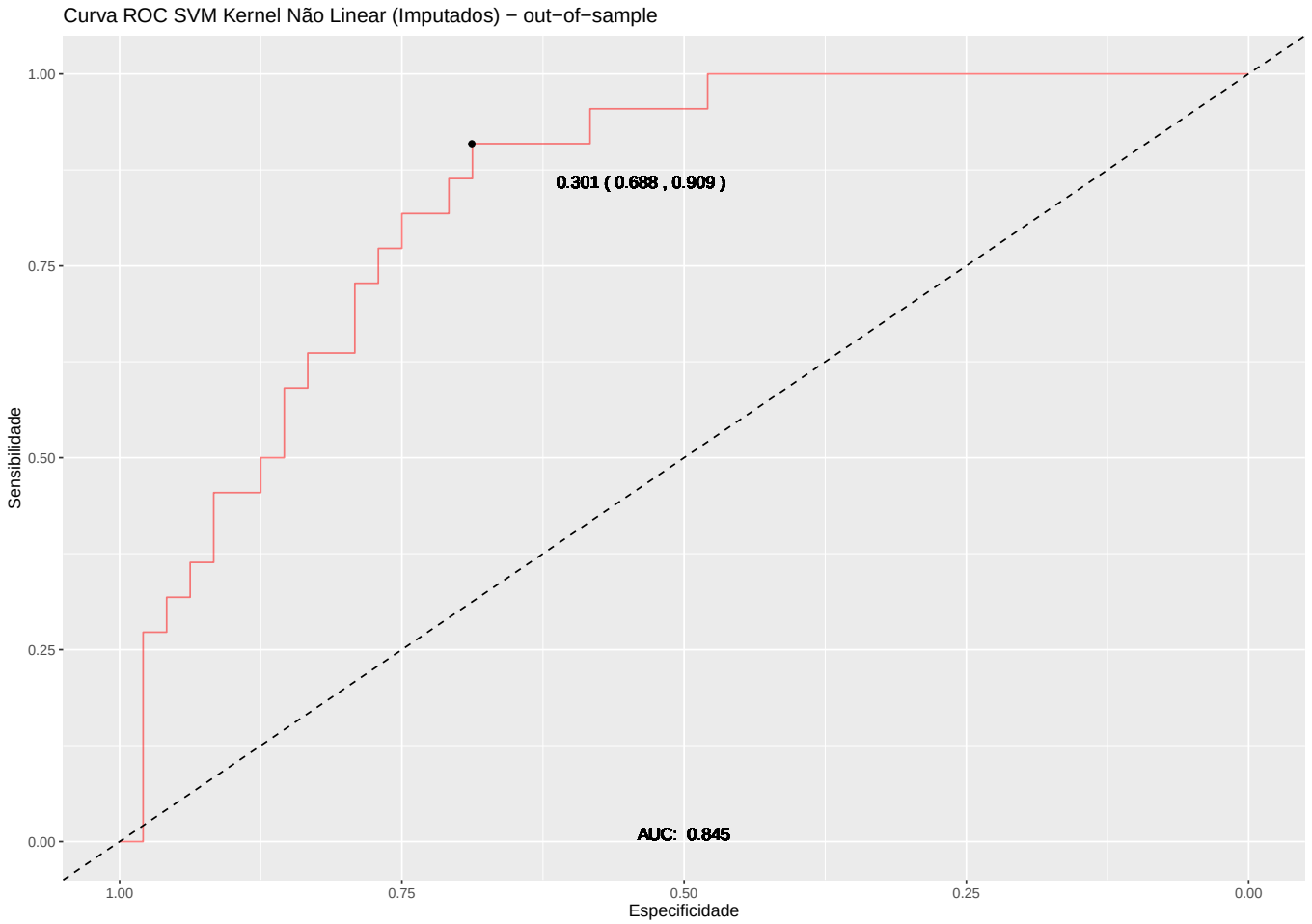
Já para o conjunto dos dados faltantes removidos (Partição 2), a análise discriminante quadrática apresentou as melhores métricas em três dos quatro critérios. Assim, para esse conjunto de dados, foi escolhida para ser avaliada no conjunto de teste.

[–] **SVM *Kernel* Não-Linear**: Conjunto de Teste (*out-of-sample*) da Partição 1

Para o conjunto de teste da Partição 1, obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	38	8
Yes	10	14

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.845.



Finalmente o desempenho geral do modelo sobre o conjunto de teste da Partição 1 é apresentado na tabela seguinte.

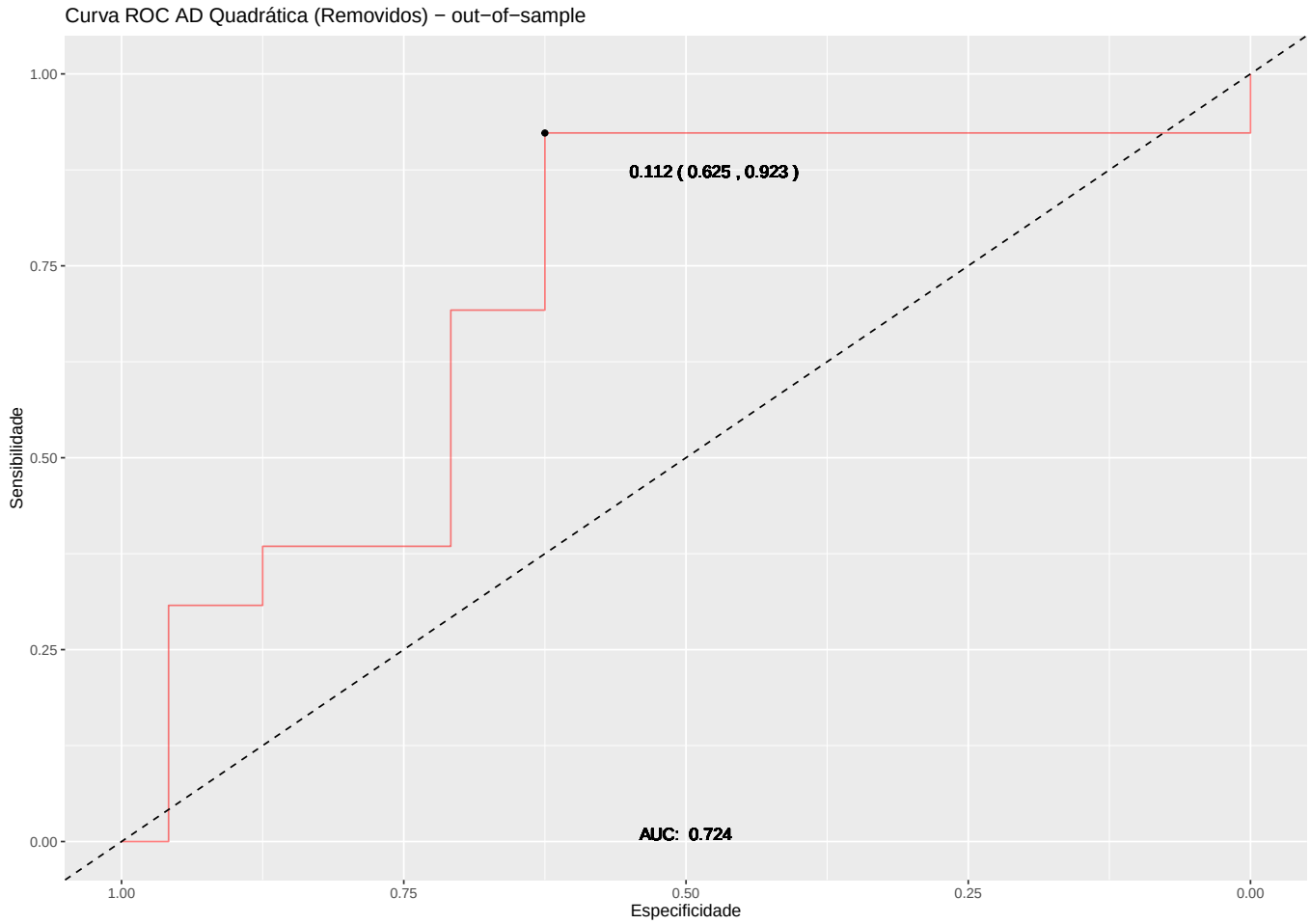
Desempenho do Modelo			
Accuracy	0.7429	Sensitivity	0.7917
95% CI	(0.6244, 0.8399)	Specificity	0.6364
No Information Rate	0.6857	Pos Pred Value	0.8261
P-Value [Acc >NIR]	0.1846	Neg Pred Value	0.5833
		Prevalence	0.6857
Kappa	0.4177	Detection Rate	0.5429
		Detection Prevalence	0.6571
Mcnemar's Test P-Value	0.8137	Balanced Accuracy	0.7140

[–] **Análise Discriminante Quadrática**: Conjunto de Teste (*out-of-sample*) da Partição 2

Para o conjunto de teste da Partição 2 (*missings* removidos), obteve-se a matriz de confusão da tabela seguinte.

	Referência	
Predito	**No**	**Yes**
No	17	5
Yes	7	8

O desempenho na curva ROC é mostrado no gráfico seguinte. O AUC foi de 0.724.



Finalmente o desempenho geral do modelo sobre o conjunto de teste da Partição 2 é apresentado na tabela seguinte.

Desempenho do Modelo			
Accuracy	0.6757	Sensitivity	0.7083
95% CI	(0.5021, 0.8199)	Specificity	0.6154
No Information Rate	0.6486	Pos Pred Value	0.7727
P-Value [Acc >NIR]	0.4384	Neg Pred Value	0.5333
		Prevalence	0.6486
Kappa	0.3127	Detection Rate	0.4595
		Detection Prevalence	0.5946
Mcnemar's Test P-Value	0.7728	Balanced Accuracy	0.6619

Conclusão

Dada a dificuldade no diagnóstico da diabetes, um mecanismo que possa prever se determinado paciente possui a doença baseado em seus dados e exames clínicos poderia auxiliar muito no diagnóstico dos médicos. Foi com esse objetivo maior que uma série de diferentes modelos preditivos foi avaliada. Com base em um conjunto de dados problemático por apresentar muitos dados faltantes, foram consideradas duas estratégias de análise. A primeira foi a avaliação de desempenho sobre um conjunto de dados imputados (Partição 1). A segunda foi a avaliação sobre um conjunto com os dados faltantes removidos (Partição 2). Em cada um desses conjuntos foi separada uma parte de treinamento, uma parte de validação e uma parte de teste (utilizada somente após a decisão final sobre o modelo). A partir desta metodologia foi possível identificar qual a melhor estratégia (utilizar dados imputados ou remoção de dados faltantes) que maximiza a acurácia para cada método preditivo empregado.

Por meio de diferentes modelos preditivos, dentre eles: análise discriminante, regressão logística, florestas aleatórias e máquinas de vetores de suporte, foi possível avaliar os conjuntos de dados de treinamento segundo diversas métricas de desempenho e eleger o modelo mais adequado ao conjunto de dados analisado. Ao fim, foi escolhido um modelo de máquina vetorial de suporte para a primeira partição e um modelo de análise discriminante para a segunda partição. O desempenho final desses modelos foi avaliado com os conjuntos de testes onde se obteve uma acurácia superior a 0.74 de predição para os dados imputados e de 0.67 para os dados não imputados. A abordagem proposta de avaliar diferentes métodos de predição e conseguinte votação do melhor método para o conjunto de dados avaliado demonstrou maior robustez em comparação a utilização de um único método. Essa robustez está relacionada ao fato de diminuir o viés empregado por cada método, assim como, identificar possíveis limitações dos mesmos. Nesse sentido, acreditamos que a abordagem proposta possa servir de grande auxilio na seleção de modelos preditivos robustos no campo de aprendizagem estatística e que a mesma possa ser aplicada em diferentes contextos de problemas de classificação em dados biomédicos e na redução de custos de saúde pública.

Referências

[1] AKAIKE, Hirotugo. *A Bayesian Analysis of The Minimum AIC Procedure*. **Annals of the Institute of Statistical Mathematics**. Vol.30, 1978, pp. 9–14.

[2] ALADE, O. A., Selamat A., Sallehuddin R. *The Effects of Missing Data Characteristics on the Choice of Imputation Techniques*. Vietnam J. Comp. Sc. v. 7, p. 161–177. No. 2, Vietnam, mar, 2020.

[3] ARLOT, Sylvain; CELISSE, Alain. *A Survey of Cross-Validation Procedures for Model Selection*. **Statistics Surveys**. Vol. 4, 2010, pp. 40–79.

[4] FERREIRA, Paulo H.; LOUZADA, Francisco; DINIZ, Carlos A. R. *Credit Scoring Modeling With State-Dependent Sample Selection: a Comparison Study With The Usual Logistic Modeling*. **Pesquisa Operacional**. Vol. 35, Nº 1, 2015.

[5] GARETH James, Daniela Witten, Trevor Hastie, Robert Tibshirani *An Introduction to Statistical Learning: with Applications in R*. **Springer, New York.** 2014.

[6] HASTIE, Trevor , Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. **Springer** , 2 ed. New York, 2009.

[7] JAKE VanderPlas *Python Data Science Handbook: Essential Tools for Working with Data*. 1ª Edição, eBook Kindle, 2017.

[8] JOHNSON, Richard Arnold, Dean W. Wichern. *Applied Multivariate Statistical Analysis*. **Pearson Prentice Hall.** 6 ed, 2007.

[9] KHARROUBI, A. T., Darwish, H. M. *Diabetes mellitus: The epidemic of the century*. **World J Diabetes.** v. 25; 6(6): p. 850–867, jun 2015.

[10] OSHIRO, Thais Mayumi. *Uma abordagem para a construção de uma única árvore a partir de uma Random Forest para classificação de bases de expressão gênica*. **Dissertação (Mestrado em Bioinformática) – Universidade de São Paulo, Ribeirão Preto**, 2013.

[11] SMITH, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. *Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus*. **In: Proceedings of the Symposium on Computer Applications and Medical Care . IEEE Computer Society Press.** pp. 261–265, 1988.

[12] SINGER, Julio da Motta; MORETTIN, Pedro Alberto. Introdução à Ciência de Dados: *Fundamentos e Aplicações*. Versão parcial e preliminar. **Instituto de Matemática e Estatística (IME)**, São Paulo, Maio de 2020.

[13] TIBISHIRANI, R., James, G., Witten D., Hastie, T. *An Introduction to Statistical Learning: with Applications in R.* **Springer**, 8 ed. New York, 2013.

[14] VIANA, Renato Frazzato. Técnicas de Classificação Aplicadas a *Credit Scoring*: Revisão Sistemática e Comparação. **Dissertação (Mestrado em Estatística) - Estatística Interinstitucional do ICMC e UFSCar, Universidade de São Paulo**, São Carlos, 2015