Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Classificação aplicada a dados extraídos de áudio de biossinais de animais

Augusto Paulo Silva

São Carlos - SP

Classificação aplicada a dados extraídos de áudio de biossinais de animais

Augusto Paulo Silva

***Orientador*: Prof. Dr. Elaine Parros Machado de Sousa**

Monografia do projeto de conclusão de curso, submetida ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, em cumprimento parcial dos requerimentos do curso de Engenharia de Computação.
Área de Concentração: Mineração de Dados.

USP - São Carlos
Junho 2021

Silva, Augusto Paulo.
Classificação aplicada a dados extraídos de áudio de biossinais de animais / Augusto Paulo Silva - São Carlos - SP, 2021.
60 p. ; 29,7 cm.

Orientador: Elaine Parros Machado de Sousa.
Monografia (Graduação) - Instituto de Ciências Matemáticas e de computação (ICMC/USP), São Carlos - SP, 2021

1. Mineração de dados. 2. Análise exploratória. 3. Dados de áudio. 4. Biossinais. Ⅰ. Sousa, Elaine Parros Machado de. Ⅱ. Instituto de Ciências Matemáticas e Computação (ICMC/USP).
Ⅲ. Classificação aplicada a dados extraídos de áudio de biossinais de animais

Resumo

Neste projeto foi realizado um estudo exploratório para encontrar, empiricamente, métodos de pré-processamento e de aprendizado que favorecem uma análise supervisionada de gravações de vocalizações animais, utilizando um conjunto construído a partir de dados fornecidos pela Faculdade de Medicina Veterinária e Zootecnia da USP (FMVZ-USP). A análise realizada indicou métodos eficazes para a classificação do uso de Cloridrato de Ractopamina (RAC), incluindo seleções de dados e pré-processamentos que refinam a performance desses métodos. Os resultados obtidos podem ser utilizados em projetos de pesquisa futuros na área de análise de áudio de biossinais animais.

Índice

Introdução

1.1 Motivação do estudo

A demanda do mercado internacional por produtos animais e carnes está em ascensão, com expectativa de um consumo dobrado até o ano de 2050, um aumento que pode ser explicado pela melhoria global nos padrões de vida (ROJAS-DOWNING et al., 2017). O aumento acelerado na demanda, incentiva o estudo de técnicas com alta produtividade, resultando na criação das tecnologias de tipo "agricultura 4.0", que envolvem, por exemplo, análise de dados durante a criação de animais. Câmeras e sensores podem ser instalados para a coleta de informação, gerando dados 24/7 que podem ser armazenados e analisados de forma pouco intrusiva e com custo relativamente baixo. A análise dos dados coletados pode gerar *insights* sobre a atividade e consumo dos animais, algo que pode trazer ganhos significativos para a produção.

A detecção da variação comportamental dos animais pode ser considerada algo extremamente importante dentro deste contexto, pois um aumento na agressividade, por exemplo, pode impactar negativamente a produtividade e bem estar dos animais durante a sua criação. Alguns estudos mostram que a adição de Cloridrato de Ractopamina (RAC), por exemplo, tem alguma correlação com o aumento da agressividade, comportamentos mais alertas e aumento de vocalizações de suínos (POLETTO et al., 2010).

Bons resultados podem ser obtidos neste contexto, com o uso de técnicas de análise de áudio. A análise dos ruídos animais pode permitir a identificação de conhecimento relevante utilizando apenas gravações de áudio, algo que foi realizado em estudos anteriores de forma altamente acurada, com o objetivo de identificar doenças suínas debilitantes (CHUNG et al., 2013).

Pesquisadores da Faculdade de Medicina Veterinária e Zootecnia (FMZV) e da Universidade de São Paulo (USP) estão conduzindo estudos sobre a provisão de RAC em um grupo de porcos em fase de terminação. Além de dados coletados

manualmente, foi criada uma infraestrutura de câmeras com monitoramento de 24 horas que é armazenado em formato de vídeo. A partir das faixas de áudio presentes no conjunto de vídeos, foram aplicadas técnicas de análise de áudio para a extração de *features*, formando um conjunto de dados que será descrito em detalhes na Seção 2.1. Neste trabalho de conclusão de curso, foi realizada uma análise exploratória, com o objetivo principal de determinar métodos eficazes para a realização análise supervisionada neste estudo de casos, e *insights* sobre o conjunto de dados que podem ser utilizados em estudos futuros.

1.2 Objetivos de projeto

A partir de um conjunto de dados inicial, contendo *features* extraídas do áudio fornecido por pesquisadores da FMVZ, foi feita uma análise exploratória, buscando identificar variações do comportamento suíno, antes e após a adição de RAC em sua dieta. Em seguida, os casos com melhores resultados de predição serão selecionados para execução em subconjuntos diferentes do utilizado durante a análise anterior.

Este projeto é uma continuidade do trabalho de conclusão de curso desenvolvido por Souza (2020), que realizou uma análise exploratória baseada em métodos não supervisionados de aprendizado. Este projeto busca estender o escopo de análise, utilizando métodos supervisionados de aprendizado, comparando a performance de diferentes métodos de classificação, e executando múltiplos algoritmos de seleção de *features*.

Os resultados obtidos neste trabalho, podem auxiliar análises futuras de vocalização animal utilizando áudio, permitindo uma identificação mais simples e acurada da ocorrência de variações comportamentais.

1.3 Organização do Texto

Este documento é organizado da seguinte maneira:

- O Capítulo 2 aborda as ferramentas e algoritmos utilizados no decorrer do projeto

- O Capítulo 3 descreve a abordagem utilizada durante a análise, apresenta os resultados e estuda os dados obtidos em cada experimento.
- O Capítulo 4 discute as conquistas do projeto, incluindo sugestões para pesquisas futuras e alguns comentários sobre a experiência obtida pelo autor.

Ferramentas e Conceitos

2.1 Conjunto de dados

O conjunto de dados foi construído a partir de um conjunto de vídeos fornecido por pesquisadores da FMZV. Utilizando as faixas de áudio presentes nos vídeos, foram extraídas as *features* do conjunto, produzindo um total de 92 atributos e 12106 observações, das quais 8872 observações foram feitas após a adição de RAC na dieta.

As *features* presentes neste conjunto de dados inicial são utilizadas frequentemente em análise de áudio (Chung et al., 2013, Klapuri e Davy, 2006, e Giannakopoulos e Pikrakis, 2014b), e o conjunto completo de atributos é listado na Tabela 1.

Tabela 1: *Features* existentes no conjunto de dados. Caso não especificado, a variável tem tipo float.

0. datetime (datetime)	31. mfcc0	62. mfcc_delta_10
1. ala (string)	32. mfcc_delta_0	63. mfcc_accelerate_10
2. grupo (int)	33. mfcc_accelerate_0	64. mfcc11
3. file_name (string)	34. mfcc1	65. mfcc_delta_11
4. zero_crossing_rate	35. mfcc_delta_1	66. mfcc_accelerate_11
5. zero_crossings	36. mfcc_accelerate_1	67. mfcc12
6. spectrogram	37. mfcc2	68. mfcc_delta_12
7. mel_spectrogram	38. mfcc_delta_2	69. mfcc_accelerate_12
8. harmonics	39. mfcc_accelerate_2	70. mfcc13
9. perceptual_shock_wave	40. mfcc3	71. mfcc_delta_13
10. spectral_centroids	41. mfcc_delta_3	72. mfcc_accelerate_13
11. spectral_centroids_delta	42. mfcc_accelerate_3	73. mfcc14
12. spectral_centroids_accelerate	43. mfcc4	74. mfcc_delta_14
13. chroma1	44. mfcc_delta_4	75. mfcc_accelerate_14
14. chroma2	45. mfcc_accelerate_4	76. mfcc15
15. chroma3	46. mfcc5	77. mfcc_delta_15
16. chroma4	47. mfcc_delta_5	78. mfcc_accelerate_15
17. chroma5	48. mfcc_accelerate_5	79. mfcc16
18. chroma6	49. mfcc6	80. mfcc_delta_16
19. chroma7	50. mfcc_delta_6	81. mfcc_accelerate_16
20. chroma8	51. mfcc_accelerate_6	82. mfcc17
21. chroma9	52. mfcc7	83. mfcc_delta_17
22. chroma10	53. mfcc_delta_7	84. mfcc_accelerate_17
23. chroma11	54. mfcc_accelerate_7	85. mfcc18
24. chroma12	55. mfcc8	86. mfcc_delta_18
25. tempo_bpm	56. mfcc_delta_8	87. mfcc_accelerate_18
26. spectral_rolloff	57. mfcc_accelerate_8	88. mfcc19
27. spectral_flux	58. mfcc9	89. mfcc_delta_19
28. spectral_bandwidth_2	59. mfcc_delta_9	90. mfcc_accelerate_19
29. spectral_bandwidth_3	60. mfcc_accelerate_9	91. rac (bool)
30. spectral_bandwidth_4	61. mfcc10	

(Fonte: Elaborado pelo Autor)

2.2 Métodos de Classificação

Foram testados vários métodos de classificação neste trabalho, buscando de forma empírica os classificadores mais eficazes para o caso abordado. Nesta seção, será realizada uma breve descrição de cada método utilizado durante a fase de análise.

2.2.1 Gaussian Naive Bayes (GNB)

O Gaussian Naive Bayes (GNB) utiliza o teorema de Bayes para prever, dada uma observação X, qual classe C tem a maior probabilidade de conter X (Han et al., 2011).

O processo é iniciado calculando a probabilidade P(Xa|C) para cada atributo a. Após este passo inicial, a probabilidade P(X|C) é obtida utilizando a multiplicação dos resultados de cada atributo.

Equação 1: $P(X|C) = P(X_1|C) * P(X_2|C) \cdots * P(X_3|C).$

Por fim o teorema de Bayes é utilizado, obtendo o resultado P(C|X).

Equação 2: $P(C|X) = [P(X|C) * P(C)] / P(X)$

Este método é repetido para todas as classes, sendo a mais provável escolhida como classe candidata.

Vale notar que, a multiplicação de probabilidades realizada durante o processo só é uma operação probabilisticamente válida caso os atributos envolvidos sejam independentes. Esta é uma simplificação assumida pelo método *Naive*, com o objetivo de otimizar a performance do código, no entanto ela introduz imprecisões no cálculo, reduzindo a eficácia prática deste método.

2.2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) é um método que interpreta o conjunto de dados geometricamente, sendo cada eixo correspondente a um atributo do conjunto. A partir desta visão geométrica, o método divide o conjunto de dados em grupos distintos, utilizando hiperplanos (Han et al., 2011).

Os hiperplanos de divisão são escolhidos de forma a maximizar as margens, ou seja, aumentando a distância entre as observações mais próximas de cada extremidade (chamadas *support vectors*) e o hiperplano. Devido à natureza de escolha dos hiperplanos, a complexidade do modelo resultante depende apenas dos *support vectors,* tornando o método menos propenso a *overfitting*.

2.2.3 K Nearest Neighbors (KNN)

O K Nearest Neighbors (KNN) é um método *lazy*, que prediz a classificação de uma observação comparando-a com os K vizinhos mais próximos (Han et al., 2011).

Para a determinação dos vizinhos próximos, é calculado inicialmente a distância entre cada atributo do conjunto de dados, utilizando uma medida de distância determinada pelo usuário. A distância final é então obtida, utilizando uma somatória entre as distâncias de atributo. É possível também atribuir pesos às distâncias de atributo, resultando em uma combinação linear para o cálculo de distância final.

As medidas de distância mais populares são a distância euclidiana, utilizada em atributos numéricos, e a codificação 0 (iguais) e 1 (diferentes) é comumente utilizada no caso de atributos nominais. O valor K pode ser escolhido empiricamente, de acordo com a performance do método no conjunto de dados aplicado.

2.2.4 Random Forest Classifier (RFC)

O Random Forest Classifier (RFC) é um método que se baseia em executar múltiplas árvores de decisão e selecionar a classe mais popular entre elas (Han et al., 2011). As árvores geradas são construídas utilizando seleção aleatória de atributos em cada nó, e a classificação final do RFC é obtida a partir de uma votação entre os resultados preditos pelas árvores.

Árvores de decisão consistem em construir fluxogramas dividindo o conjunto de dados em diferentes grupos de acordo com o valor de um atributo, até a geração de grupos com uma única classe. A maioria dos algoritmos inicia o processo a partir de um nó global, que contém toda o conjunto de dados, escolhendo um atributo de

divisão a cada nó, até a geração de nós indivisíveis, onde a classificação é realizada de acordo com a classe mais popular entre as observações restantes.

2.2.5 Multi Layer Perceptron (MLP)

O MLP é um método que utiliza múltiplos Perceptrons para realizar a classificação iterativamente. O MLP é composto de 3 partes, a entrada, que recebe os dados de treinamento, a saída, que irá realizar a predição baseando-se nas informações obtidas, e o *hidden layer*, que é responsável por realizar a maior parte do processamento (Han et al., 2011).

2.3 Métodos de Pré-Processamento

2.3.1 Variance Threshold

O Variance Threshold[1] é um método de seleção de *features* que consiste na remoção de atributos de baixa variância no conjunto de dados. Dada uma *threshold* atribuída pelo usuário, atributos com menor variância serão descartados do conjunto de dados resultando em um conjunto com apenas atributos de alta variância dos quais é esperada uma maior relevância ao problema.

2.3.2 Select K Best

O Select K Best[2] é um método de seleção que quantifica a correlação de cada atributo com as classes de saída, e seleciona as K melhores pontuações. O cálculo de correlação é atribuído pelo usuário, tornando possível a personalização do tipo de seleção feita pelo método. Para fins deste projeto, serão utilizadas apenas as funções de pontuação fornecidas pelo sklearn (f_classif[3], mutual_info_classif[4], chi2[5]), durante a execução desta seleção.

---

[1] <https://bit.ly/3xFJvh5>
[2] <https://bit.ly/3BanQjw>
[3] <https://bit.ly/3gzRXaC>
[4] <https://bit.ly/3cHkYQA>
[5] <https://bit.ly/3iR6xNo>

2.3.3 Select Percentile

O Select Percentile[6] é um método de seleção que se comporta de forma análoga ao Select K Best, quantificando a correlação de cada atributo e as classes de saída utilizando uma função dada pelo usuário. No entanto, este método seleciona os atributos com K% melhores resultados, ou seja, é escolhida uma parcela dos casos de melhor pontuação durante esta seleção, no lugar de um número absoluto como esperado em Select K Best.

2.3.4 Min Max Scaler

O Min Max Scaler é um método de normalização no qual o conjunto de dados é escalado de forma a satisfazer o intervalo atribuído pelo usuário (Han et al., 2011). Isto é realizado utilizando a seguinte equação, onde max e min são definidos pelo usuário:

$$std = (X - X_{min})/(X_{max} - X_{min})$$

Equação 3:

$$X_{sca} = std * (max - min) + min$$

2.3.5 Robust Scaler

O Robust Scaler[7] é um método de normalização no qual o conjunto é escalado utilizando cálculos robustos contra *outliers*. O método consiste em remover a mediana e escalar os dados de acordo com a Interquantile Range (IQR), o intervalo entre o primeiro e terceiro quantile.

O uso da mediana e exclusão dos valores extremos reduzem o efeito de *outliers* durante a normalização, no entanto tornam a execução deste método mais lenta para grandes conjuntos.

[6] <https://bit.ly/3yTVgkh>
[7] <https://bit.ly/2VvT18m>

2.3.6 Principal Component Analysis (PCA)

O Principal Component Analysis (PCA), é um método de redução de *features*. O método consiste na geração de K vetores ortogonais, para K menor ou igual à quantidade total de *features* no conjunto. O conjunto de dados é então transformado para a nova base K de vetores gerados pelo PCA (Han et al., 2011).

Os vetores do PCA são denominados *principal components*, e devido à sua ortogonalidade, minimizam a quantidade de informação redundante entre os atributos. Isto possibilita representar o conjunto de dados em uma menor quantidade de *features*, evitando perdas significativas de informação durante o processo.

2.4 Medidas de Qualidade

Medidas de qualidade possibilitam a quantificação da eficácia de um método de classificação, permitindo uma avaliação comparativa entre cada método utilizado.

Utilizando *true positive* (TP), *false positive* (FP), *true negative* (TN), *false negative* (FN) para notação, as medidas de qualidade utilizadas durante as análises deste projeto são (Han et al., 2011):

- Accuracy - (TP+TN) / (TP+FP+TN+FN)
- Precision - TP / (TP+FP)
- Recall - TP / (TP+FN)
- F1-Score - (2 x *precision* x *recall*) / (*precision* + *recall*)
- ROC-AUC - Área abaixo da curva do gráfico de visualização *Receiver Operating Characteristic* (ROC)

2.5 Bibliotecas Usadas

Este projeto utiliza Python 3 para a implementação do código, utilizando as seguintes bibliotecas:

- **Matplotlib**[8] - Biblioteca que simplifica a implementação de visualizações de dados

[8] <https://matplotlib.org/>

- **Numpy**[9] - Fornece ferramentas para processamento de alta performance
- **Pandas**[10] - Simplifica a manipulação e análise de dados
- **Scikit Learn**[11] - Implementa ferramentas utilizadas em aprendizado de máquina, como métodos classificadores e normalização.

Como o scikit learn implementa algoritmos de aprendizado de máquina, esta é uma das bibliotecas mais críticas para a execução do projeto. Devido à sua importância, funções desta biblioteca importantes para a realização da análise foram listadas separadamente.

- **GaussianNB**[12] - Implementa um classificador GNB.
- **SVC**[13] - Implementa um classificador SVM.
- **NearestNeighbors**[14] - Implementa um classificador KNN.
- **RandomForestClassifier**[15] - Implementa um classificador RFC.
- **MLPClassifier**[16] - Implementa um classificador MLP.
- **VarianceThreshold**[17] - Implementa a seleção Variance Threshold
- **SelectKBest**[18] - Implementa a seleção Select K Best.
- **SelectPercentile**[19] - Implementa a seleção Select Percentile.
- **MinMaxScaler**[20] - **I**mplementa a normalização Min Max Scaler.
- **RobustScaler**[21] - Implementa a normalização Robust Scaler.
- **GridSearchCV**[22] - Implementa uma busca exaustiva das configurações mais eficazes para um classificador.
- **PCA**[23] - Implementa a redução PCA.

---

[9] <https://numpy.org/>
[10] <https://pandas.pydata.org/>
[11] <https://scikit-learn.org/>
[12] <https://bit.ly/3pXXlsd>
[13] <https://bit.ly/3gsJUxu>
[14] <https://bit.ly/3xCMnez>
[15] <https://bit.ly/3wt2Rpn>
[16] <https://bit.ly/3cKvGpk>
[17] <https://bit.ly/3vwd9Ua>
[18] <https://bit.ly/3wveWdE>
[19] <https://bit.ly/3xshQj9>
[20] <https://bit.ly/3gpz6jB>
[21] <https://bit.ly/2S0UYIl>
[22] <https://bit.ly/2RXFXHt>
[23] <https://bit.ly/35s19se>

2.6 Considerações Finais

Neste capítulo, foram apresentados os conceitos e ferramentas estudados para a realização da análise. Adicionalmente, foram informadas as características do conjunto de dados e as dependências para a execução do código implementado.

Após uma etapa inicial de aprendizado, o passo seguinte envolveu a realização dos experimentos do estudo, abordados no Capítulo 3.

Experimentos e Resultados

A partir do conjunto de dados apresentado na Seção 2.1, foi escolhido para o Experimento 1 (Seção 3.1), um subconjunto de observações com o objetivo de refinar os resultados da análise a partir da redução do efeito de fatores externos ao problema.

A partir do subconjunto escolhido, foi realizada uma análise exploratória do problema, utilizando diferentes métodos de pré-processamento com o objetivo de classificar de forma eficaz as observações entre antes e após a aplicação de RAC na dieta.

Os resultados obtidos foram quantificados por medidas de qualidade, demonstrando empiricamente os métodos e subconjuntos mais efetivos para a análise deste problema. Por fim, os melhores métodos foram selecionados para as análises adicionais do Experimento 2 (Seção 3.2), seleções de subconjuntos alternativos do conjunto de dados inicial.

O objetivo principal das análises é apresentar atributos de áudio e metodologias eficazes para a detecção de mudanças na vocalização animal. Estas análises podem gerar novas intuições na aquisição de áudio e extração de *features* em estudos futuros de problemas da mesma natureza.

3.1 Estudo Experimental 1

O Experimento 1 consistiu na análise exploratória utilizando vários métodos diferentes durante o pré-processamento, para a busca de resultados mais acurados.

A busca consistiu na escolha de um algoritmo classificador, a escolha de um método de normalização, e escolha de um método de seleção, dentre os indicados na Seção 2.3. Foi feita uma permutação entre as escolhas de algoritmos, executando todas as combinações possíveis dentro do escopo, e registrando as medidas de qualidade observadas em cada caso.

Para finalizar a análise exploratória, os resultados de cada método de classificação foram comparados, e por fim foram escolhidas algumas combinações que foram utilizadas no Experimento 2.

3.1.1 Seleção de Dados de Interesse

A partir do conjunto de dados inicial, foi realizada a seleção dos dados de interesse, utilizando 3 critérios:

- Utilização de observações obtidas apenas entre as 9 e 12 horas. Este é um período de interferência humana reduzida devido à rotina de tratamento e alimentação, segundo especialistas.
- Remoção de *features* redundantes identificadas durante o trabalho anterior (Souza, 2020).
- Seleção das observações correspondentes à câmera com maior quantidade de gravações. Isto é realizado para evitar problemas relacionados à variação na qualidade de áudio.

Após a realização desta seleção utilizando os critérios descritos, o conjunto resultante foi salvo no arquivo features_readyC.csv, que é utilizado no restante do Experimento 1. Este conjunto possui 71 atributos e 517 observações, das quais 358 foram realizadas após a adição de RAC na dieta.

Os atributos restantes no conjunto após esta seleção são listados na Tabela 2.

Tabela 2: *Features* no conjunto features_readyC, resultante após a seleção inicial.

1. datetime (datetime)	30. mfcc_delta_6	59. mfcc16
2. ala (string)	31. mfcc_accelerate_6	60. mfcc_delta_16
3. grupo (int)	32. mfcc7	61. mfcc_accelerate_16
4. file_name (string)	33. mfcc_delta_7	62. mfcc17
5. zero_crossing_rate	34. mfcc_accelerate_7	63. mfcc_delta_17
6. zero_crossings	35. mfcc8	64. mfcc_accelerate_17
7. spectral_centroids	36. mfcc_delta_8	65. mfcc18
8. spectral_centroids_delta	37. mfcc_accelerate_8	66. mfcc_delta_18
9. spectral_centroids_accelerate	38. mfcc9	67. mfcc_accelerate_18
10. spectral_rolloff	39. mfcc_delta_9	68. mfcc19
11. mfcc0	40. mfcc_accelerate_9	69. mfcc_delta_19
12. mfcc_delta_0	41. mfcc10	70. mfcc_accelerate_19
13. mfcc_accelerate_0	42. mfcc_delta_10	71. rac (bool)
14. mfcc1	43. mfcc_accelerate_10	
15. mfcc_delta_1	44. mfcc11	
16. mfcc_accelerate_1	45. mfcc_delta_11	
17. mfcc2	46. mfcc_accelerate_11	
18. mfcc_delta_2	47. mfcc12	
19. mfcc_accelerate_2	48. mfcc_delta_12	
20. mfcc3	49. mfcc_accelerate_12	
21. mfcc_delta_3	50. mfcc13	
22. mfcc_accelerate_3	51. mfcc_delta_13	
23. mfcc4	52. mfcc_accelerate_13	
24. mfcc_delta_4	53. mfcc14	
25. mfcc_accelerate_4	54. mfcc_delta_14	
26. mfcc5	55. mfcc_accelerate_14	
27. mfcc_delta_5	56. mfcc15	

28.	mfcc_accelerate_5	57.	mfcc_delta_15		
29.	mfcc6	58.	mfcc_accelerate_15		

(Fonte: Elaborado pelo Autor)

3.1.2 Normalização

Foram utilizados 2 métodos distintos de normalização, MinMaxScaler e RobustScaler, executados em configuração padrão em todos os casos de análise.

Devido às diferenças de escala, não é possível reutilizar resultados de análise entre cada método de normalização. Algo que também deve ser notado, é que o RobustScaler gera uma escala centrada em 0, ou seja com a presença de números negativos. Devido a este fato, há uma incompatibilidade entre este método e a funçõa chi2 de pontuação, que assume o uso de apenas números positivos. Isto reduz as alternativas de kernel de SelectKBest, durante a análise exploratória do RobustScaler.

3.1.3 Seleção e Extração de Features

Foram utilizados 3 métodos distintos de seleção de *features*: VarianceThreshold, SelectKBest, SelectPercentile. Adicionalmente, foram utilizadas diferentes medidas de qualidade durante a execução do SelectKBest, de forma a gerar 3 casos distintos durante a análise desta seleção.

Após a escolha do método de seleção, a análise foi executada repetidamente, com redução progressiva do número de *features*. Isto foi realizado com o objetivo de buscar empiricamente a quantidade de *features* ótima para a seleção executada. Após a realização desta busca, foi selecionado o caso de maior acurácia e menor número de *features*.

Por fim, foi realizada uma redução PCA no caso mais acurado observado durante a etapa anterior. Este passo foi realizado de forma semelhante à seleção de *features*, utilizando execuções com progressivamente menor número de atributos. O caso com menor número de atributos e maior acurácia é escolhido para a representação do algoritmo durante a avaliação final do experimento.

3.1.4 Classificação e Avaliação de Qualidade

Nesta seção serão apresentados os resultados individuais de cada método classificador e por fim será apresentado uma avaliação sucinta dos resultados do Experimento 1.

3.1.4.1 GaussianNB

Este método de classificação, devido à sua relativa simplicidade, tem uma quantidade reduzida de hiperparâmetros, com efeitos não significativos durante a classificação para este conjunto de dados. Isto resultou no uso apenas da configuração padrão do método neste projeto, removendo a necessidade de uma etapa de otimização.

A análise exploratória foi realizada, utilizando os algoritmos listados na Seção 2.5.1. O melhor caso observado consiste na combinação de RobustScaler, SelectKBest com medida f_classif e redução PCA, que resulta no uso de apenas 5 features, e tem a pontuação descrita na Tabela 3. Todos os resultados da execução completa do experimento com o GaussianNB são apresentados na Seção A.1 do Apêndice.

Melhor Caso: Conjunto pós seleção de *features*
spectral_centroids, spectral_rolloff, mfcc0, mfcc15, mfcc16, mfcc18

Tabela 3: Medidas de qualidade para o melhor caso observado com uso de GaussianNB.

Melhor Caso: GNB + RobustScaler + SelectKBest + PCA (5 Features)	
Accuracy	0.85 ± 0.07
Precision	0.89 ± 0.07
Recall	0.90 ± 0.11
F1-Score	0.81 ± 0.09
ROC-AUC	0.89 ± 0.07

(Fonte: Elaborado pelo Autor)

3.1.4.2 SVC

O SVC é um algoritmo que possui alguns hiperparâmetros notáveis, cuja configuração altera a eficácia do método de forma perceptível, neste caso de uso. Com o auxílio da função GridSearchCV, foram testados os hiperparâmetros *C*, *class_weight*, *gamma*, *degree* e *kernel*, em busca das configurações mais acuradas neste conjunto de dados. O escopo da busca é apresentado na Tabela 4.

Tabela 4: Escopo de busca do GridSearchCV para o método SVC.

	C	class_weight	gamma	degree	kernel
Valores Atribuídos	0.001, 0.01, 0.1, 1, 10, 100, 1000, 10000	'balanced', None	0.001, 0.01, 0.1, 1, 10, 100, 'scale'	3, 4, 5, 6, 7, 8	'rbf', ‘poly’, ‘sigmoid’, ‘linear’

(Fonte: Elaborado pelo Autor)

Após a execução desta etapa, a configuração *{'C': 1000, 'class_weight': None, 'gamma': 0.001, 'kernel': 'rbf'}* foi escolhida para uso durante a análise com MinMaxScaler.

O caso mais eficaz observado neste método, consiste na combinação dos algoritmos MinMaxScaler, SelectKBest com medida chi2, e redução PCA, resultando em um conjunto de 9 features com a sua performance apresentada na Tabela 5. Todos os resultados da execução completa do experimento com o SVC são apresentados na Seção A.2 do Apêndice.

Melhor Caso: Conjunto pós seleção de *features*
zero_crossing_rate, zero_crossings, spectral_centroids, spectral_rolloff, mfcc0, mfcc2, mfcc3, mfcc4, mfcc5, mfcc6, mfcc8, mfcc9, mfcc11, mfcc15, mfcc_accelerate_15, mfcc16, mfcc18

Tabela 5: Medidas de qualidade para o melhor caso observado com uso de SVC.

Melhor Caso: SVC + MinMaxScaler + SelectKBest + PCA (9 Features)	
Accuracy	0.91 ± 0.03
Precision	0.93 ± 0.05
Recall	0.94 ± 0.07
F1-Score	0.89 ± 0.04
ROC-AUC	0.97 ± 0.02

(Fonte: Elaborado pelo Autor)

3.1.4.3 NearestNeighbors

O NearestNeighbors possui 2 hiperparâmetros notáveis, que foram explorados durante a sua otimização: *n_neighbors* e *weights*. O escopo da busca atribuída ao GridSearchCV é apresentado na Tabela 6.

Tabela 6: Escopo de busca do GridSearchCV para o método NearestNeighbors

	n_neighbors	weights
Valores	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	uniform, distance

(Fonte: Elaborado pelo Autor)

A configuração escolhida após a busca é {'n_neighbors': 16, 'weights': 'uniform'}, para o RobustScaler.

Após a etapa de otimização, é realizada a análise exploratória, utilizando os algoritmos da Seção 2.5.1. O melhor caso observado durante esta análise consiste no uso do RobustScaler, SelectKBest com medida f_classif, e redução PCA, resultando em um conjunto com 7 *features* e performance apresentada na Tabela 7. Todos os resultados da execução completa do experimento com o NearestNeighbors são apresentados na Seção A.3 do Apêndice.

Melhor Caso: Conjunto pós seleção de *features*
zero_crossing_rate, zero_crossings, spectral_centroids, spectral_rolloff, mfcc0, mfcc4, mfcc5, mfcc6, mfcc8, mfcc9, mfcc11, mfcc15, mfcc16, mfcc18

Tabela 7: Medidas de qualidade para o melhor caso observado com uso de KNN.

Melhor Caso: KNN + RobustScaler + SelectKBest + PCA (7 Features)	
Accuracy	0.87 ± 0.05
Precision	0.89 ± 0.05
Recall	0.94 ± 0.07
F1-Score	0.84 ± 0.06
ROC-AUC	0.93 ± 0.06

(Fonte: Elaborado pelo Autor)

3.1.4.4 RandomForestClassifier

O RandomForestClassifier possui 2 hiperparâmetros notáveis, utilizados durante a otimização com GridSearchCV: max_features e min_samples_split. O escopo desta busca é apresentado na Tabela 8.

Tabela 8: Escopo de busca do GridSearchCV, para o método RFC.

	max_features	min_samples_split
Valores	sqrt, log2	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

(Fonte: Elaborado pelo Autor)

A configuração escolhida para o método é {'max_features': 'sqrt', 'min_samples_split': 11}, para o MinMaxScaler.

Após a escolha da configuração do método, é realizada a análise exploratória do RFC. O caso escolhido para este método consiste no uso dos algoritmos, MinMaxScaler, SelectKBest com medida mutual_info_classif, e redução PCA resultando em um conjunto de 4 *features* com a performance apresentada na Tabela 9. Todos os resultados da execução completa do experimento com o RandomForestClassifier são apresentados na Seção A.4 do Apêndice.

Melhor Caso: Conjunto pós seleção de *features*
spectral_centroids, spectral_rolloff, mfcc0, mfcc8, mfcc9, mfcc15, mfcc18

Tabela 9: Medidas de qualidade para o melhor caso observado com uso de RFC.

Melhor Caso: RFC + MinMaxScaler + SelectKBest + PCA (4 Features)	
Accuracy	0.86 ± 0.06
Precision	0.89 ± 0.07
Recall	0.92 ± 0.07
F1-Score	0.83 ± 0.08
ROC-AUC	0.92 ± 0.04

(Fonte: Elaborado pelo Autor)

3.1.4.5 MLPClassifier

O MLPClassifier possui 3 hiperparâmetros notáveis, *solver*, *alpha*, *activation*, que foram aplicados ao GridSearchCV para a otimização do método. O escopo da busca de otimização é apresentado na Tabela 10.

Tabela 10: Escopo de busca do GridSearchCV, para o método MLP.

	solver	alpha	activation
Valores	lbfgs, adam, sgd	0.0001, 0.001, 0.01, 0.1, 1	identity, logistic, tanh, relu

(Fonte: Elaborado pelo Autor)

A configuração escolhida para o método é {'activation': 'relu', 'alpha': 0.001, 'solver': 'adam'}, para o MinMaxScaler.

Após a realização da análise exploratória é escolhida uma combinação de algoritmos mais eficaz neste conjunto de dados. Os algoritmos escolhidos são o MinMaxScaler, SelectKBest com medida f_classif, e redução PCA, resultando em um conjunto de 10 *features* após o pré-processamento. A performance final do método é apresentada na Tabela 11. Todos os resultados da execução completa do experimento com o MLPClassifier são apresentados na Seção A.5 do Apêndice.

Melhor Caso: Conjunto pós seleção de *features*
zero_crossing_rate, zero_crossings, spectral_centroids, spectral_rolloff, mfcc0, mfcc4, mfcc5, mfcc6, mfcc8, mfcc9, mfcc11, mfcc15, mfcc16, mfcc18

Tabela 11: Medidas de qualidade para o melhor caso observado com uso de MLP.

Melhor Caso: MLP + MinMaxScaler + SelectKBest + PCA (10 Features)	
Accuracy	0.92 ± 0.04
Precision	0.94 ± 0.03
Recall	0.94 ± 0.08
F1-Score	0.90 ± 0.05
ROC-AUC	0.97 ± 0.02

(Fonte: Elaborado pelo Autor)

3.1.5 Considerações do Experimento 1

A realização do Experimento 1 tornou perceptível as vantagens na utilização de seleção de *features* para este estudo. Entre os melhores casos observados durante o experimento, foram utilizadas no máximo 10 *features* após o pré-processamento, uma redução notável na dimensionalidade do conjunto de dados. Adicionalmente, foi observada uma leve melhoria nas medidas de qualidade dos melhores casos, se comparadas às execuções de conjunto completo, sugerindo uma redução no ruído de aprendizado após a realização de seleção de *features*.

Os resultados do experimento também sugerem presença reduzida de *outliers* neste conjunto de dados. Isto foi inferido a partir do fato de vários métodos utilizarem em seus melhores casos, a função MinMaxScaler. A função MinMaxScaler utiliza a média, valores máximos e mínimos em sua escala, tornando-a suscetível a deslocamentos causados por *outliers*, que podem reduzir a eficácia de análise em certos conjuntos de dados. A alternativa, RobustScaler evita deslocamentos de *outiliers*, devido ao seu uso de mediana e *percentiles*, e seria esperada uma preferência clara a classificações com esta normalização em casos com alta interferência de *outliers* no conjunto.

Outra conclusão que o Experimento 1 permitiu é a escolha dos métodos de classificação eficazes para este conjunto de dados. É notável que os métodos SVC e MLPClassifier possuem as melhores medidas de qualidade observadas no experimento, com uma acurácia de 0.91 e 0.92, respectivamente. Com estas informações, o Experimento 2 foi realizado utilizando a combinação presente nos melhores casos do SVC e MLPClassifier.

3.2 Estudo Experimental 2

O Experimento 2 executou novamente os casos com maior eficácia do Experimento 1, desta vez utilizando seleções diferentes do subconjunto de dados inicial. Para esta análise, foram selecionados os melhores casos do MLPClassifier e SVC, que apresentaram os melhores resultados observados.

A remoção de *features* redundantes apresentada na Seção 3.1.1 foi mantida, portanto a lista de 71 *features* dos subconjuntos do Experimento 2 é correspondente à do Experimento 1, apresentada na Tabela 2. Outro passo mantido durante esta análise é a seleção de observações de apenas uma câmera, a mais frequente.

Foram utilizados 2 subconjuntos diferentes para este Experimento:

- **Intervalo B:** Gravações ocorridas entre as 6 e 18h, com 2873 observações, das quais 1990 foram realizadas após a adição de RAC.
- **Intervalo C:** Gravações durante as 24h, com 6336 observações, das quais 4439 foram realizadas após a adição de RAC.

Os resultados da performance destes subconjuntos serão comparados com a performance observada durante o Experimento 1 (**intervalo A**), no subconjunto de gravações entre as 9 e 12h.

3.2.1 Classificação e Avaliação de Qualidade

O Experimento 2 executou os métodos SVC e MLPClassifier em novos conjuntos de dados. Os melhores casos do Experimento 1 foram reutilizados, com as mesmas combinações de algoritmos e configurações nos métodos de classificação. A

única alteração envolveu a quantidade de *features* removidas durante a seleção, que foi reavaliada para os novos conjuntos do Experimento 2.

O objetivo deste experimento é verificar alterações na eficácia dos métodos de classificação durante a aplicação de conjuntos mais heterogêneos e de maior volume em relação ao Experimento 1.

3.2.1.1 SVC

O SVC foi configurado com os hiperparâmetros *{'C': 1000, 'class_weight': None, 'gamma': 0.001, 'kernel': 'rbf'}*, obtidos durante o experimento 1, e os algoritmos executados consistem no melhor caso: SVC, MinMaxScaler, SelectKBest com medida chi2 e PCA. Os resultados obtidos para o intervalo B e C são apresentados nas Tabelas 12 e 13, respectivamente. Todos os resultados da execução completa do experimento com o SVC são apresentados na Seção B.1 do Apêndice.

Tabela 12: Medidas de qualidade para o intervalo B com uso de SVC.

Intervalo B: SVC + MinMaxScaler + SelectKBest + PCA (8 Features)	
Accuracy	0.87 ± 0.08
Precision	0.90 ± 0.06
Recall	0.93 ± 0.12
F1-Score	0.84 ± 0.10
ROC-AUC	0.93 ± 0.06

(Fonte: Elaborado pelo Autor)

Tabela 13: Medidas de qualidade para o intervalo C com uso de SVC.

Intervalo C: SVC + MinMaxScaler + SelectKBest + PCA (11 Features)	
Accuracy	0.88 ± 0.07
Precision	0.91 ± 0.05
Recall	0.92 ± 0.09
F1-Score	0.85 ± 0.08
ROC-AUC	0.94 ± 0.04

(Fonte: Elaborado pelo Autor)

3.2.1.2 MLPClassifier

O MLPClassifier foi configurado com os hiperparâmetros {'activation': 'relu', 'alpha': 0.001, 'solver': 'adam'}, e os algoritmos executados são: SVC, MinMaxScaler, SelectKBest com medida f_classif e PCA. Os resultados obtidos para o intervalo B e C são apresentados nas Tabelas 14 e 15, respectivamente. Todos os resultados da execução completa do experimento com o MLPClassifier são apresentados na Seção B.2 do Apêndice.

Tabela 14: Medidas de qualidade para o intervalo B com uso de MLP.

Intervalo B: MLP + MinMaxScaler + SelectKBest + PCA (27 Features)	
Accuracy	0.89 ± 0.07
Precision	0.93 ± 0.06
Recall	0.93 ± 0.10
F1-Score	0.87 ± 0.09
ROC-AUC	0.97 ± 0.03

(Fonte: Elaborado pelo Autor)

Tabela 15: Medidas de qualidade para o intervalo C com uso de MLP.

Intervalo C: MLP + MinMaxScaler + SelectKBest + PCA (17 Features)	
Accuracy	0.91 ± 0.08
Precision	0.94 ± 0.06
Recall	0.93 ± 0.10
F1-Score	0.88 ± 0.09
ROC-AUC	0.97 ± 0.03

(Fonte: Elaborado pelo Autor)

3.2.2 Considerações do Experimento 2

Durante a execução do Experimento 2, foi observada uma ligeira queda na qualidade da seleção de *features* realizada durante o pré-processamento. A quantidade de atributos selecionados subiu, de 10 para 27 *features* nos casos de maior eficácia. Adicionalmente, houve uma queda discreta nas pontuações de medidas de qualidade,

principalmente no caso do método SVC. Esta queda na eficácia do aprendizado pode ser atribuída ao comportamento suíno, durante os novos horários incluídos no conjunto.

O intervalo A utilizado no Experimento 1, entre as 9 e 12 horas, foi recomendado por especialistas devido à menor interferência humana e à rotina de tratamento dos animais. Este intervalo favorece uma análise mais eficiente, pois além da redução de interação humana, um intervalo diário reduzido minimiza o efeito de variações comportamentais rotineiras dos animais durante a análise.

No subconjunto B, de intervalo entre as 6 e 18 horas, o horário diurno como um todo é incluído. Este horário é comprovadamente o mais ativo em áudio (Souza, 2020), e também existem outros fatores que podem ter influência nos resultados, como interferência humana e comportamento rotineiro dos animais. É provável que este conjunto seja o mais desafiador em termos de aprendizado, devido à relevância das variações externas.

Por fim, foi realizada uma análise utilizando todas as observações do dia, 24 horas (intervalo C). Este conjunto adiciona o horário noturno ao caso anterior, intervalo que se mostra menos ativo em áudio (Souza, 2020). Devido à redução geral de ruídos, é seguro inferir que as observações noturnas são relativamente homogêneas, se comparado às diurnas. Adicionalmente, o intervalo noturno ocupa metade do conjunto de observações, reduzindo a importância global das variações observadas durante o horário diurno. É possível concluir então que, devido à metade noturna, a complexidade desta análise é reduzida se comparada ao caso B, resultando em uma performance ligeiramente mais favorável durante a execução do caso C.

3.3 Considerações Finais

A análise exploratória permitiu a identificação de métodos de pré-processamento e aprendizado supervisionado adequados para este problema, visando auxiliar análises de comportamento animal baseado em áudio. O Experimento 1 demonstrou uma ótima performance na seleção de *features*, com redução de dimensionalidade e inclusive melhoria na eficácia dos métodos de classificação. Isto

sugere que há uma parcela significativa de atributos correlacionados e redundantes no conjunto, que não agregam informações relevantes para a classificação e por isso interferem nos resultados da análise no conjunto completo. Os métodos de classificação SVC e MLPClassifier demonstraram melhores resultados e foram escolhidos para uso no restante dos estudos.

O Experimento 2 mostrou os efeitos de uma seleção inteligente de observações durante o aprendizado de classificação. O intervalo A apresentou os melhores resultados, pois efeitos externos como interferência humana e rotina animal foram reduzidos neste intervalo. Por outro lado, o intervalo B apresentou uma queda significativa na eficácia, pois acrescentou observações heterogêneas na análise, aumentando a complexidade do problema. Por fim, o intervalo C apresentou uma melhoria discreta em relação ao intervalo B, devido à adição do horário noturno que adicionou um grupo de observações homogêneo no conjunto, reduzindo a complexidade do problema.

Conclusão

Este projeto utilizou uma abordagem exploratória para a realização de análises supervisionadas sobre um conjunto de dados baseado em vocalização animal. O Experimento 1 demonstrou a eficácia de vários métodos de classificação, os atributos mais relevantes para a identificação de vocalizações e *insights* sobre o conjunto de dados construído. O Experimento 2 demonstrou os efeitos da seleção de observações na análise, permitindo a identificação de limitadores da eficácia de análise no estudo de caso, e algumas correções capazes de otimizar a acurácia de predição.

4.1 Contribuições

As contribuições deste projeto podem ser sumarizadas da seguinte forma:

- Os objetivos iniciais foram alcançados após a realização do Experimento 1, que apresentou métodos e atributos eficazes na análise supervisionada para este estudo de caso.
- O Experimento 2 estendeu os resultados do estudo, demonstrando o impacto de uma seleção mais abrangente de dados de interesse. Isto permitiu a identificação de limitantes de eficácia na análise supervisionada deste estudo de caso.
- Foi constatada uma diferença entre horários diurnos e noturnos, onde horários noturnos apresentam uma menor atividade sonora, o que torna observações neste intervalo relativamente mais homogêneas se comparado às observações diurnas. Isto sugere a realização de uma divisão entre as análises de cada intervalo, para otimização da eficácia dos modelos de predição.

A realização deste projeto foi uma experiência muito positiva na formação do autor, pois se tornou uma oportunidade de executar em prática metodologias de Mineração de Dados em situações reais, buscando modelos eficazes para o estudo, gerando visualizações intuitivas, e a produção de conhecimento útil a partir da análise dos resultados de aprendizado. Outros conhecimentos adquiridos são as bibliotecas e a

linguagem Python 3, ferramentas que o autor tinha pouco domínio antes da realização do projeto.

4.2 Pesquisa Futura

Para a realização de pesquisas futuras, é possível realizar as seguintes ações para estender os resultados de estudo:

- Utilização de diferentes intervalos do dia para a análise, evitando a utilização de intervalos muito longos e dividindo horários diurnos e noturnos, para maior eficácia de predição.
- Utilização dos dados da outra câmera, que foram descartados inicialmente para uma análise mais eficaz. A diferença na qualidade de áudio entre os dados das 2 câmeras não permite o uso do conjunto completo em uma única análise, mas é possível obter novas informações no uso de uma amostra diferente do conjunto de dados em uma nova análise.
- O escopo da busca exploratória pode ser estendido para a obtenção de modelos mais eficazes. No projeto foram selecionados para uso métodos e hiperparâmetros populares de análise supervisionada, muitos algoritmos não foram incluídos no escopo e podem gerar resultados interessantes em seu uso.

Por fim, utilizando os resultados dos estudos, é possível identificar os passos mais importantes para a construção de uma análise supervisionada eficaz em casos de identificação de variações no comportamento animal a partir de biossinais. Os resultados obtidos podem então auxiliar a geração de novos conhecimentos na área de comportamento animal.

Referências Bibliográficas

CHUNG, Y.; OH, S.; LEE, J.; PARK, D.; CHANG, H.-H.; KIM, S. Automatic detection and recognition of pig wasting diseases using sound data in audio surveillance systems. Sensors , v. 13, n. 10, p. 12929–12942, 2013. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/13/10/12929>

GIANNAKOPOULOS, T.; PIKRAKIS, A. (Ed.). Introduction to Audio Analysis. Oxford: Academic Press, 2014. p. 59 – 103. ISBN 978-0-08-099388-1. Disponível em: <http://www.sciencedirect.com/science/article/pii/B9780080993881000042>.

HAN, J.; KAMBER M.; PEI, J.; Data Mining Concepts and Techniques, v. 3, p. 271-471, 2011

KLAPURI, A.; DAVY, M. Signal processing methods for music transcription. In: . [S.l.]: Springer Science & Business Media, 2006. cap. 2. ISBN 978-0-387-30667-4.

POLETTO, R.; MEISEL, R. L.; RICHERT, B. T.; CHENG, H. W.; MARCHANT-FORDE, J. N. Behavior and peripheral amine concentrations in relation to ractopamine feeding, sex, and social rank of finishing pigs. Journal of Animal Science , v. 88, n. 3, p. 1184–1194, 03 2010. ISSN 0021-8812. Disponível em: <https://doi.org/10.2527/jas.2008-1576>

ROJAS-DOWNING, M. M.; NEJADHASHEMI, A. P.; HARRIGAN, T.; WOZNICKI, S. A. Climate change and livestock: Impacts, adaptation, and mitigation. Climate Risk Management , v. 16, p. 145 – 163, 2017. ISSN 2212-0963. Disponível em: <http://www.sciencedirect.com/science/article/pii/S221209631730027X>

SOUZA A. M.; Feature extraction and mining for exploratory analysis of biological audio data. Monografia de Conclusão de Curso - Bacharelado em Ciências e Computação, ICMC-USP, 2020.

STEVENS, S. S.; VOLKMANN, J.; NEWMAN, E. B. A scale for the measurement of the psychological magnitude pitch. The Journal of the Acoustical Society of America , v. 8, n. 3, p. 185–190, 1937. Disponível em: <https://doi.org/10.1121/1.1915893>

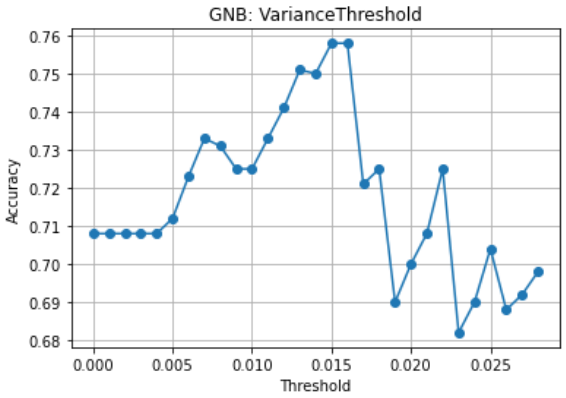
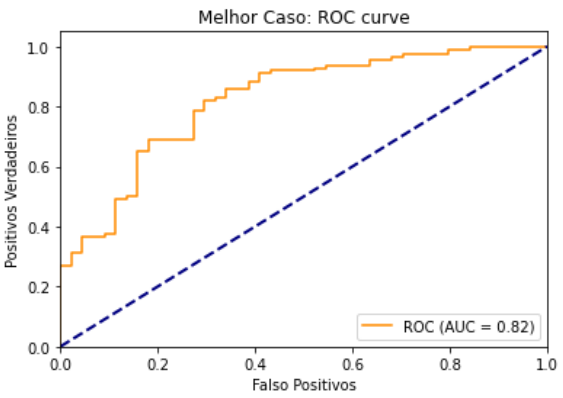
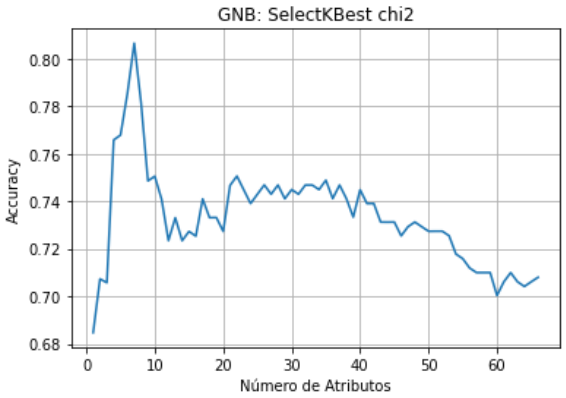
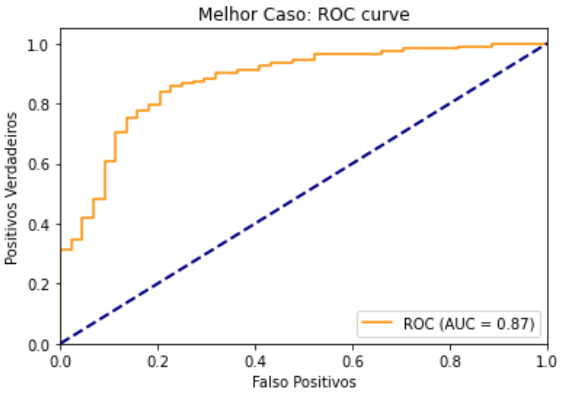
Apêndice

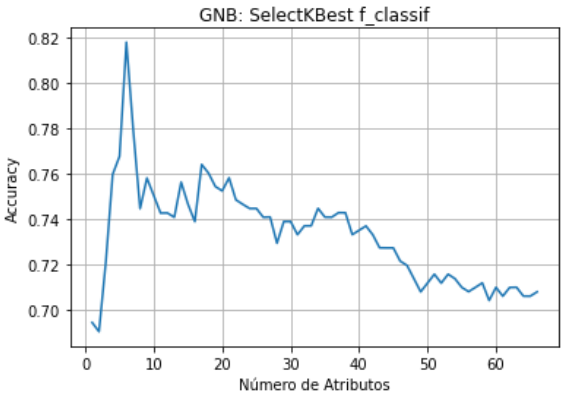
Esta seção irá apresentar todos os resultados obtidos durante a análise exploratória dos experimentos, incluindo a acurácia de predição de acordo com número de atributos incluídos no conjunto após o pré-processamento, e a ROC *curve*, que permite uma visualização intuitiva da performance de predição do caso selecionado a cada execução.

A. Experimento 1

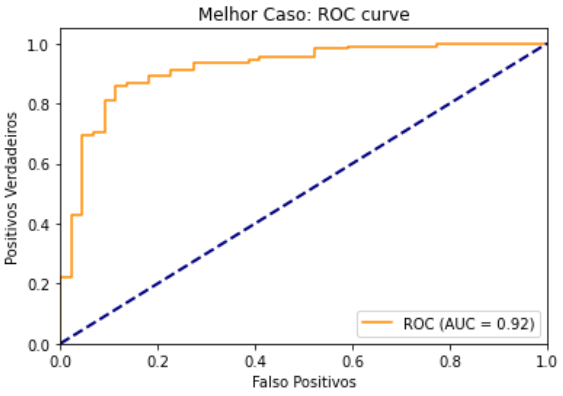
A.1 GaussianNB

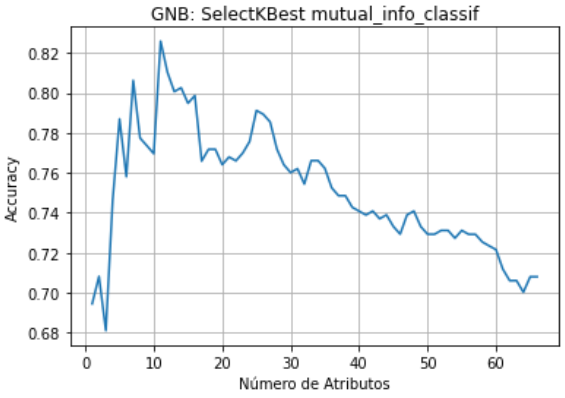
Tabela 16: Resultados da classificação GNB durante a análise exploratória utilizando MinMaxScaler.

Resultados - MinMaxScaler	
 Melhor Caso: 38 Atributos (x = 0.015)	
 Melhor Caso: 7 Atributos	

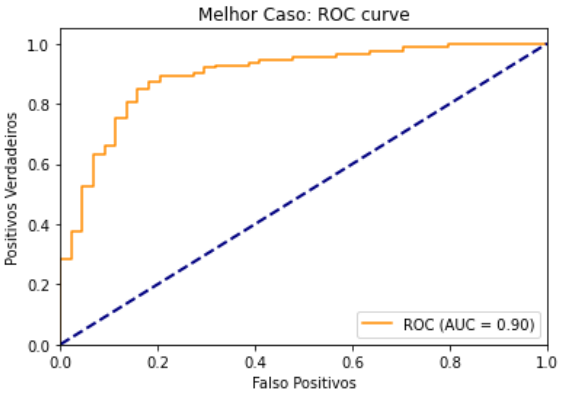


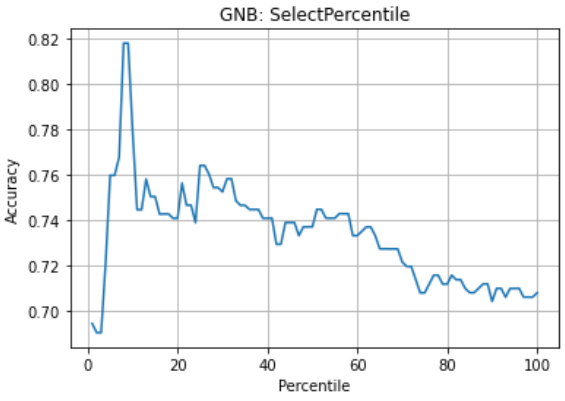
Melhor Caso: 6 Atributos



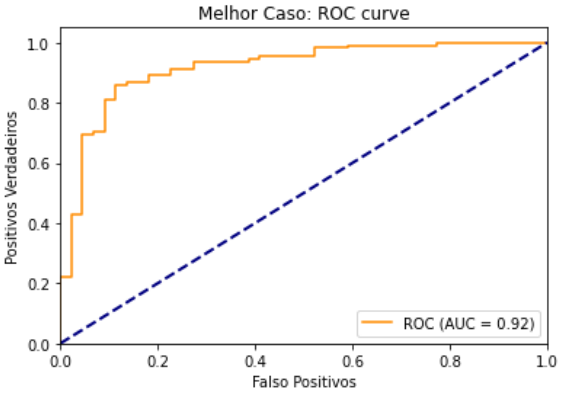


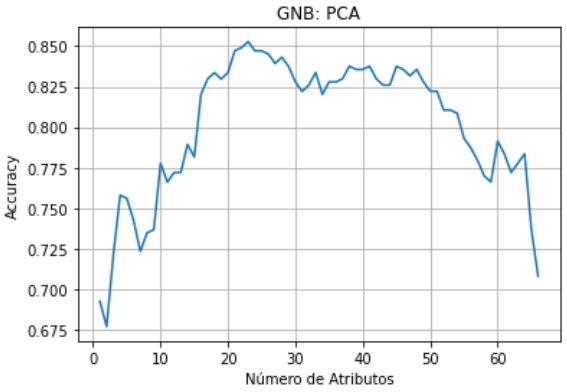
Melhor Caso: 11 Atributos



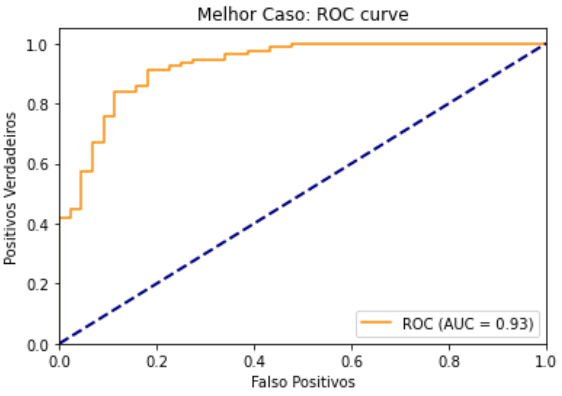


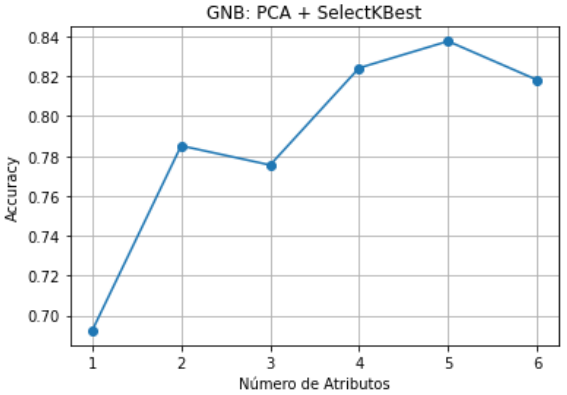
Melhor Caso: 6 Atributos (x = 8)



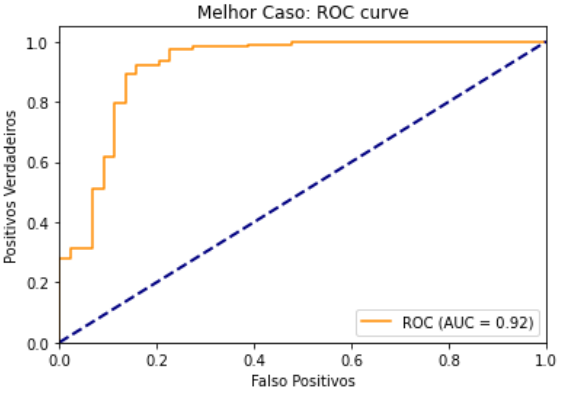


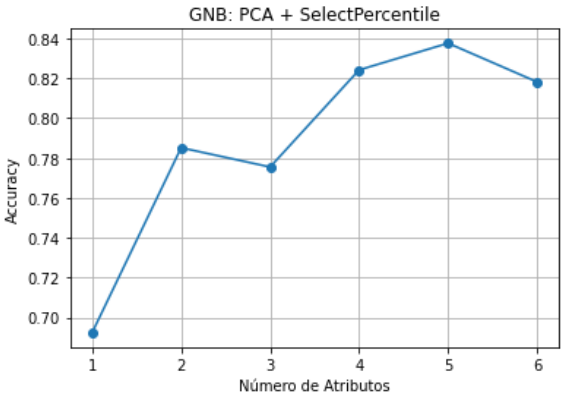
Melhor Caso: 23 Atributos



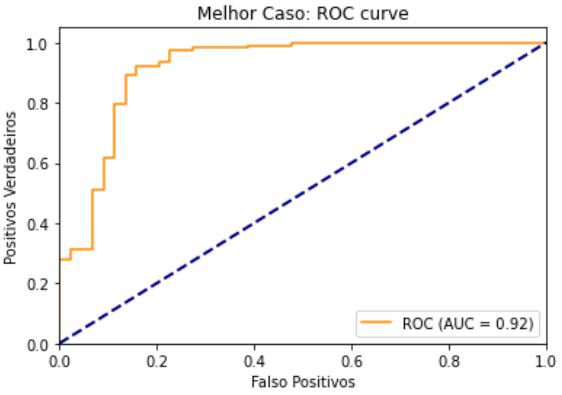


Melhor Caso: 5 Atributos



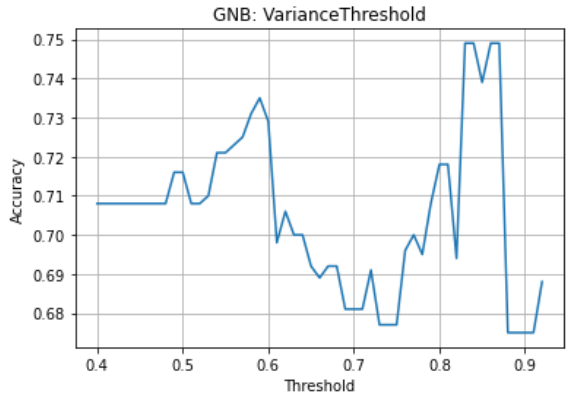
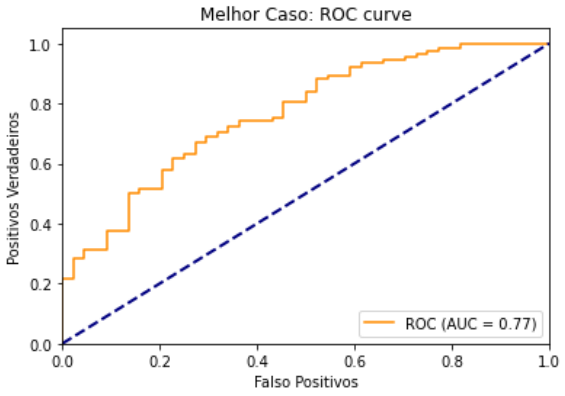
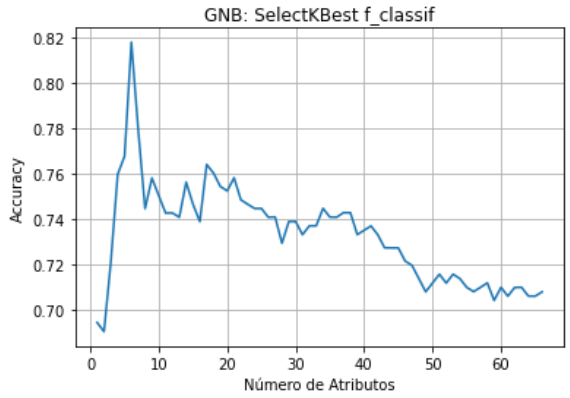
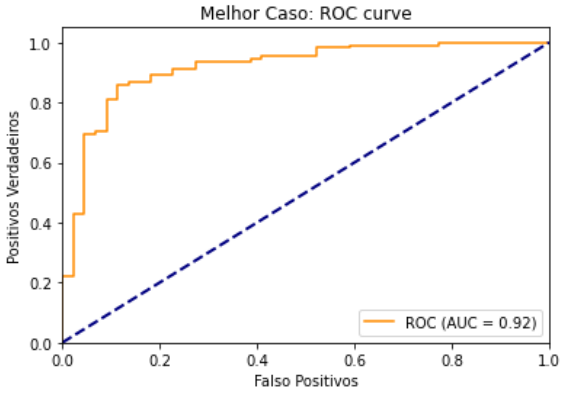
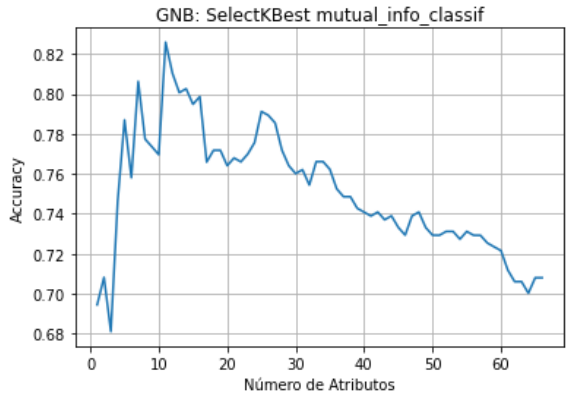
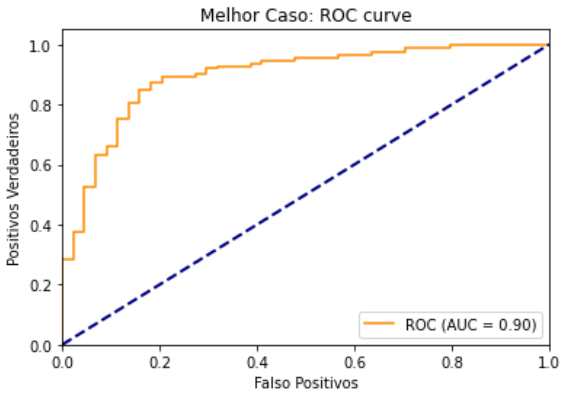


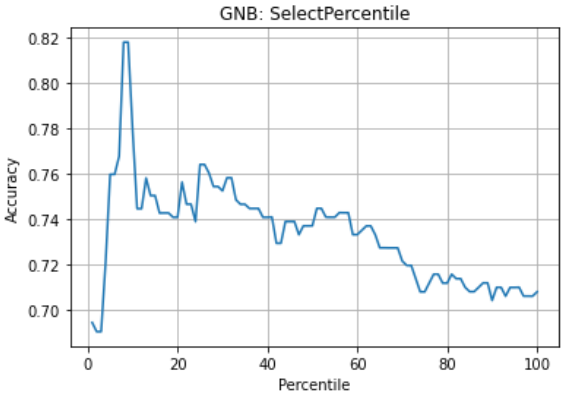
Melhor Caso: 5 Atributos



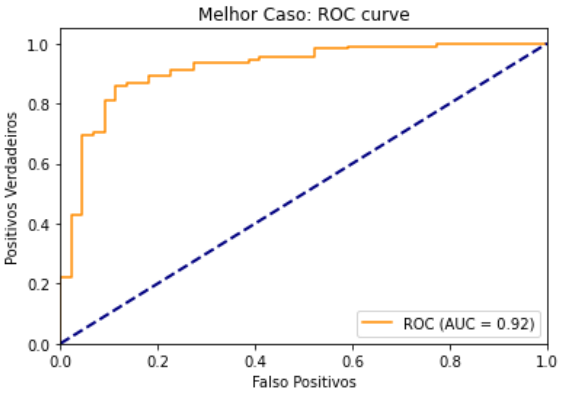
(Fonte: Elaborado pelo Autor)

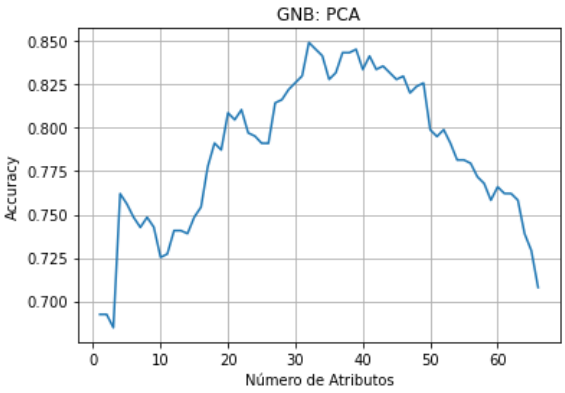
Tabela 17: Resultados da classificação GNB durante a análise exploratória utilizando RobustScaler.

Resultados - RobustScaler	
 Melhor Caso: 6 Atributos (x = 0.87)	
 Melhor Caso: 6 Atributos	
 Melhor Caso: 11 Atributos	

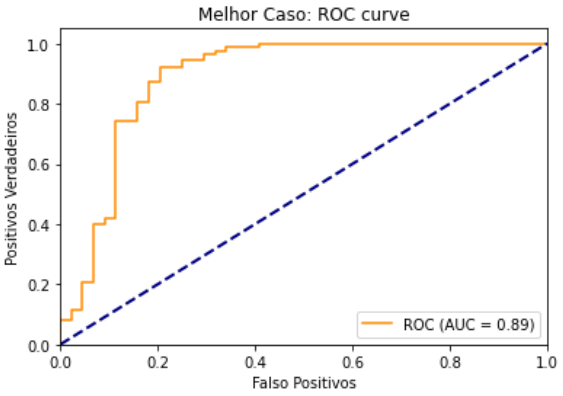


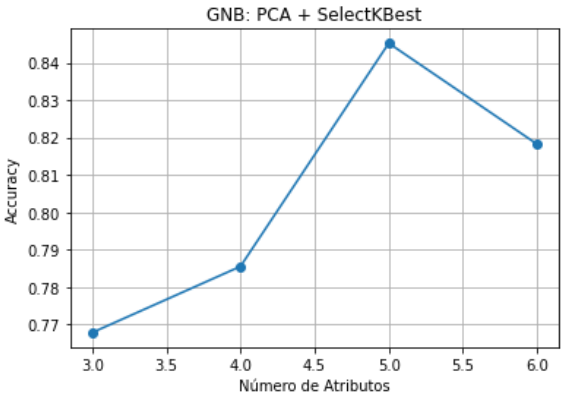
Melhor Caso: 6 Atributos (x = 8)



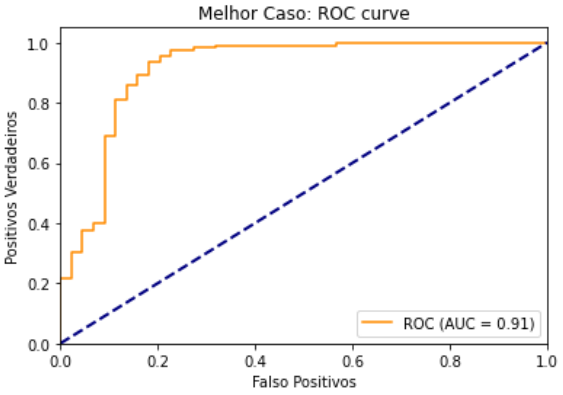


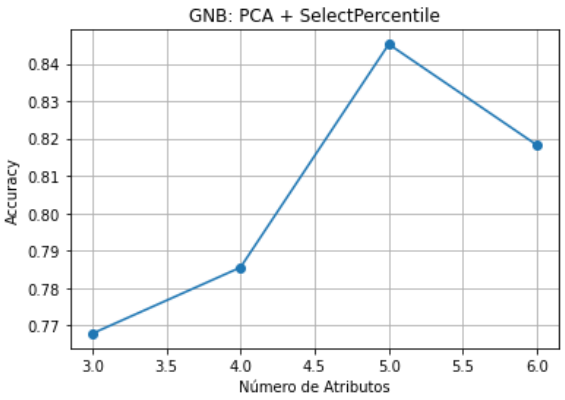
Melhor Caso: 32 Atributos

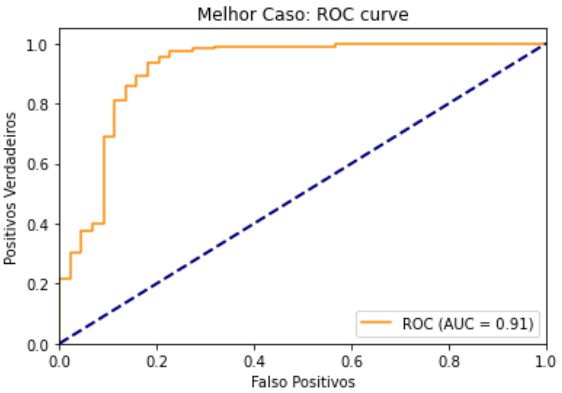




Melhor Caso: 5 Atributos





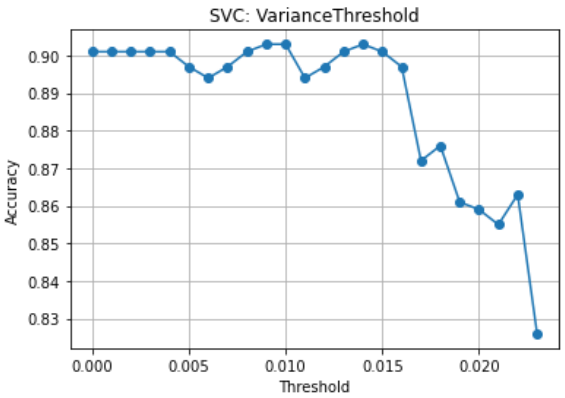
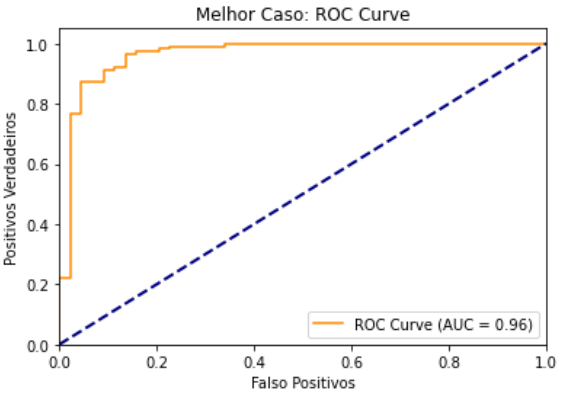
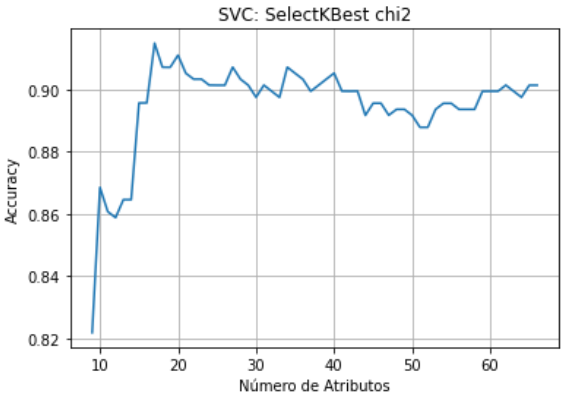
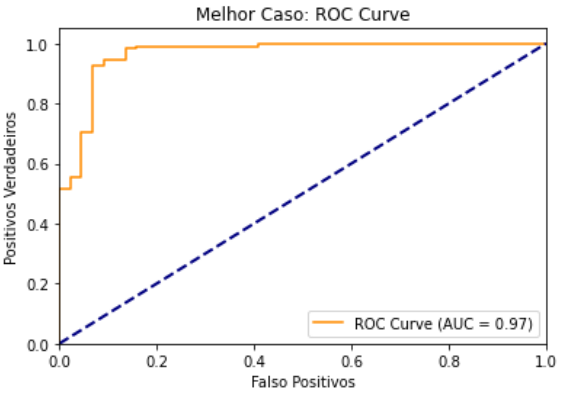


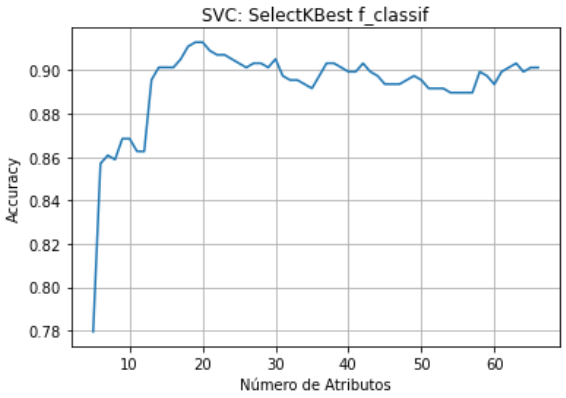
Melhor Caso: 5 Atributos

(Fonte: Elaborado pelo Autor)

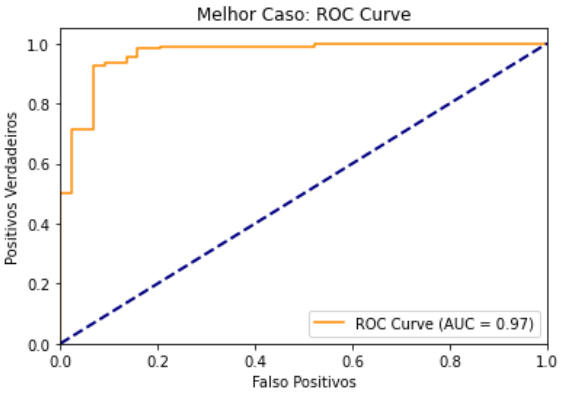
A.2 SVC

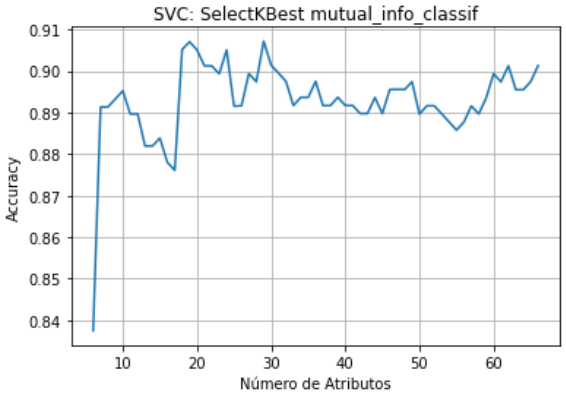
Tabela 18: Resultados da classificação SVC durante a análise exploratória utilizando MinMaxScaler.

Resultados - MinMaxScaler	
 Melhor Caso: 50 Atributos (x = 0.01)	
 Melhor Caso: 17 Atributos	

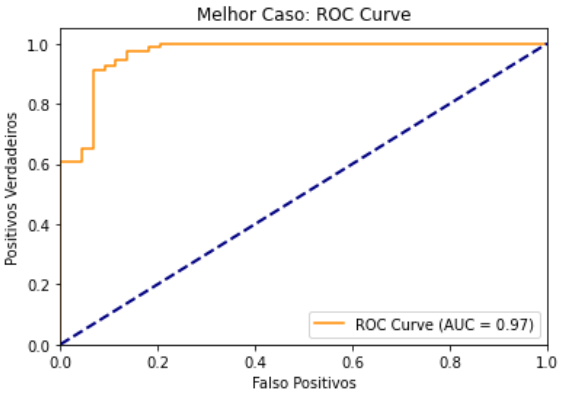


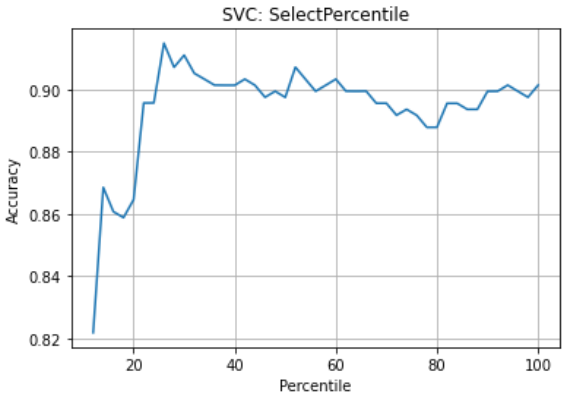
Melhor Caso: 19 Atributos



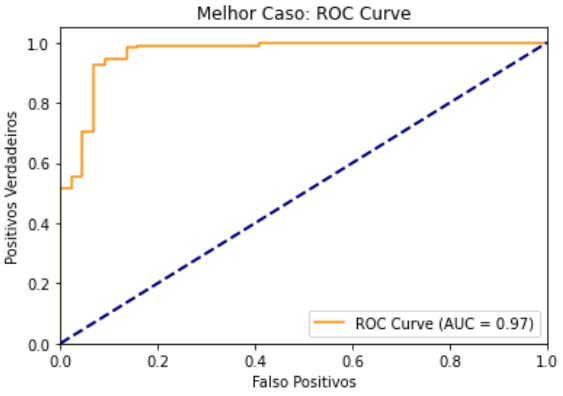


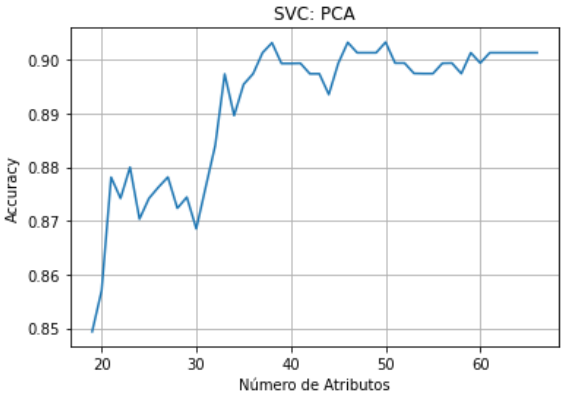
Melhor Caso: 18 Atributos



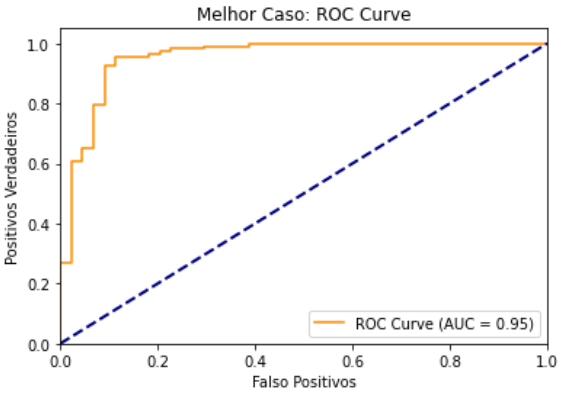


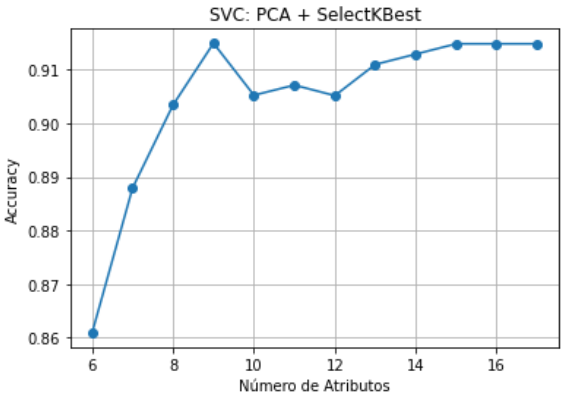
Melhor Caso: 17 Atributos (x = 25)



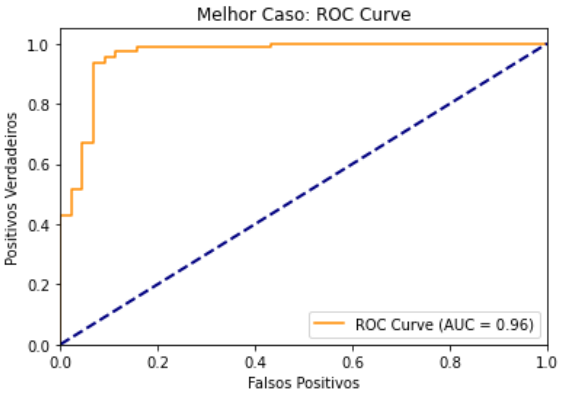


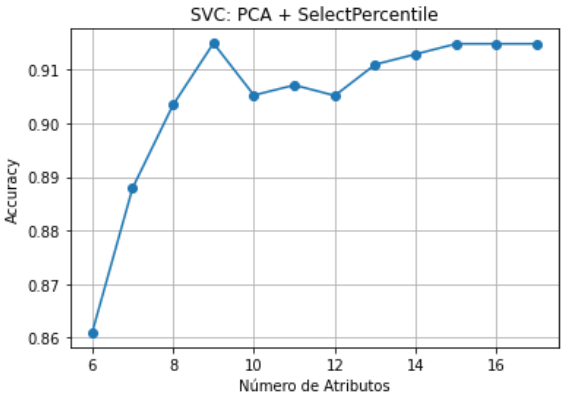
Melhor Caso: 37 Atributos



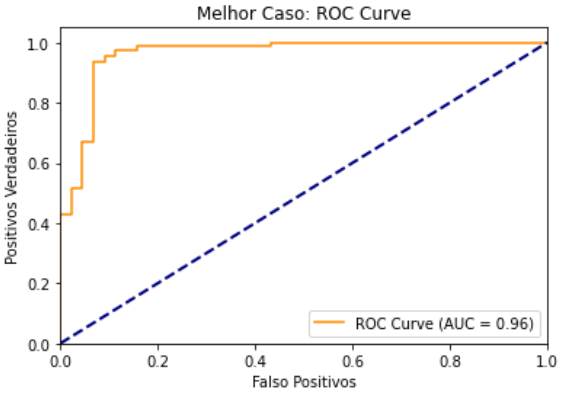


Melhor Caso: 9 Atributos



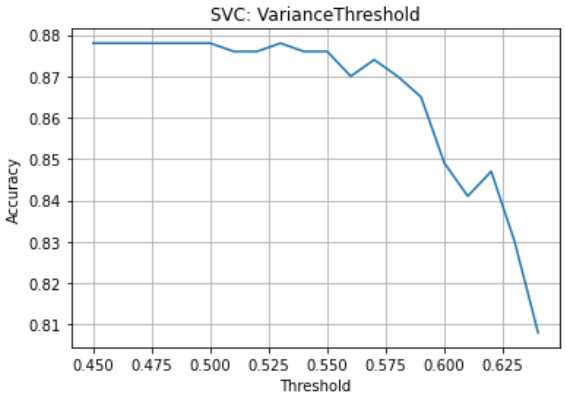
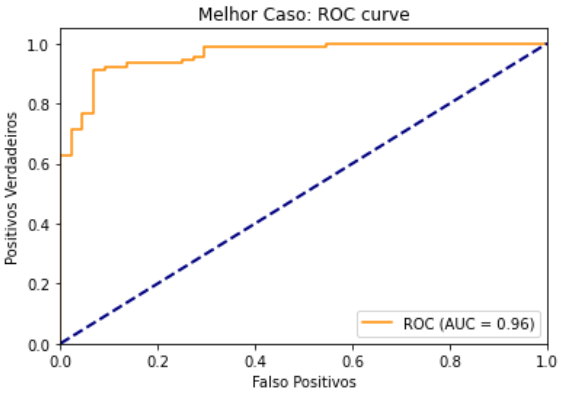
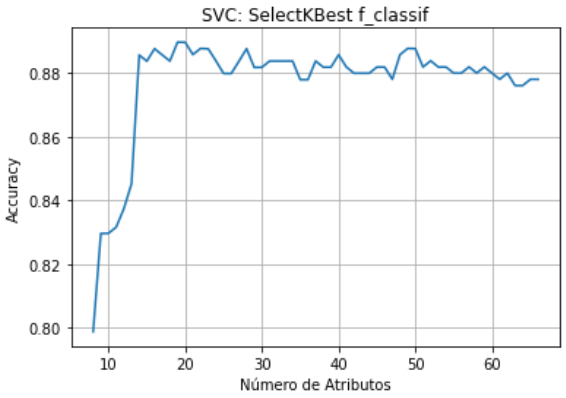
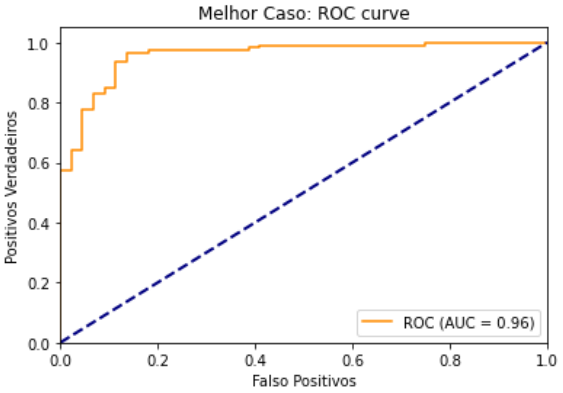
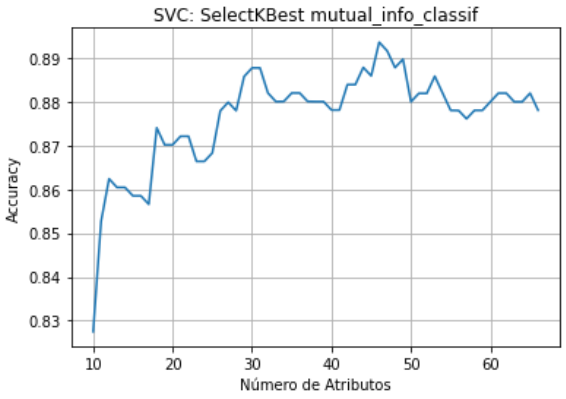
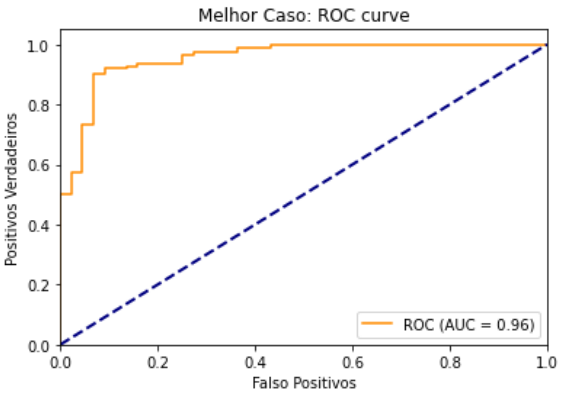


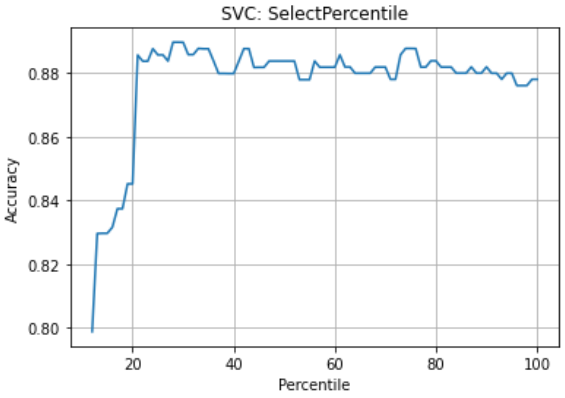
Melhor Caso: 9 Atributos



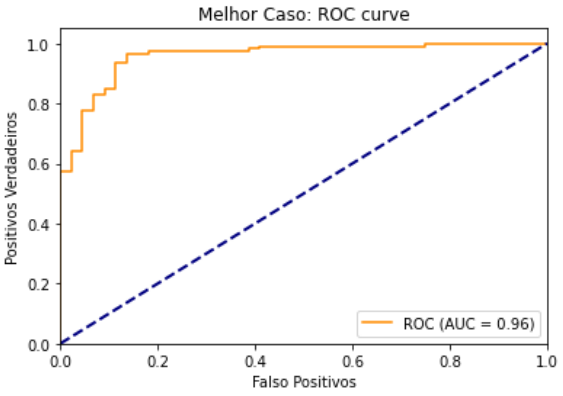
(Fonte: Elaborado pelo Autor)

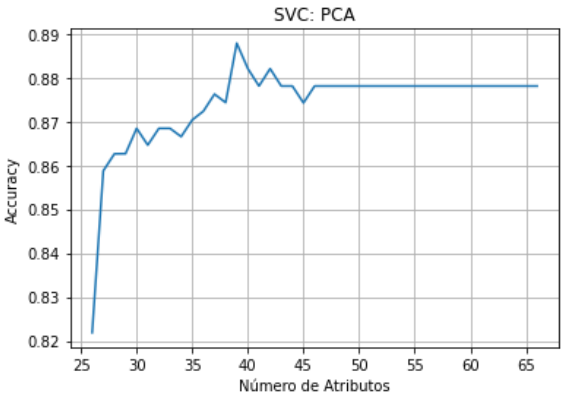
Tabela 19: Resultados da classificação SVC durante a análise exploratória utilizando RobustScaler.

Resultados - RobustScaler	
 Melhor Caso: 60 Atributos (x = 0.53)	
 Melhor Caso: 19 Atributos	
 Melhor Caso: 46 Atributos	

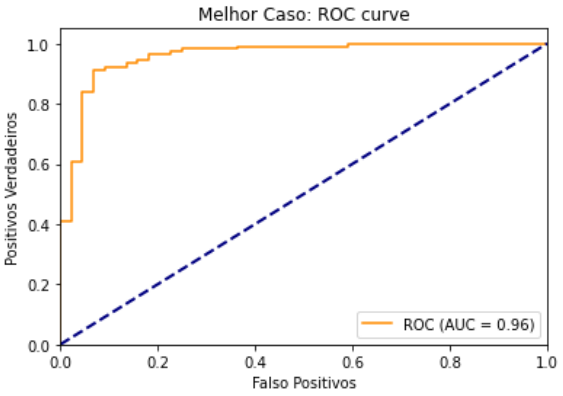


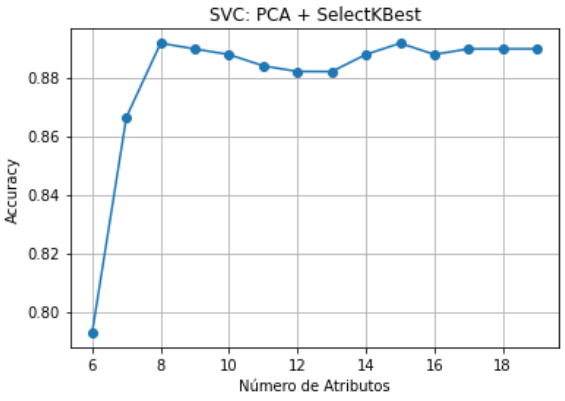
Melhor Caso: 19 Atributos (x = 28)



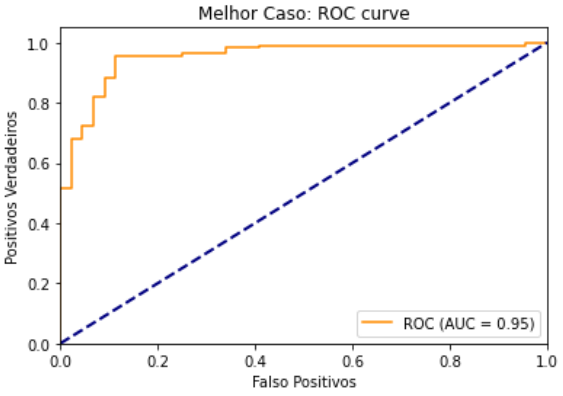


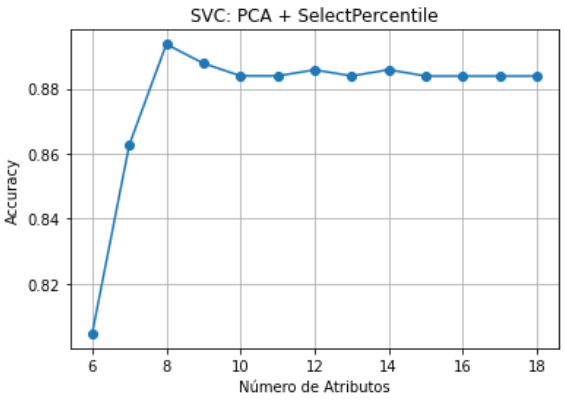
Melhor Caso: 39 Atributos

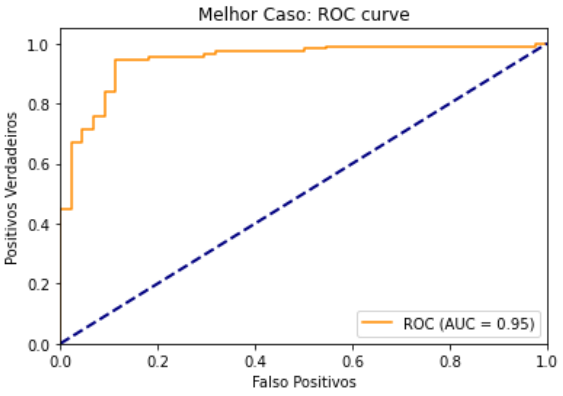




Melhor Caso: 8 Atributos





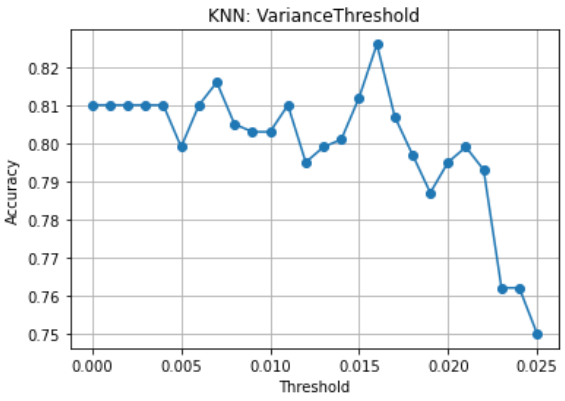
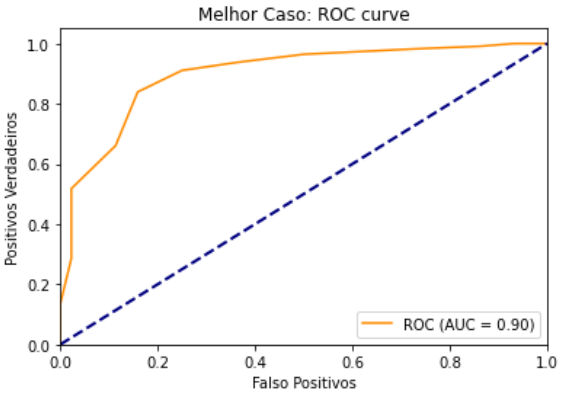
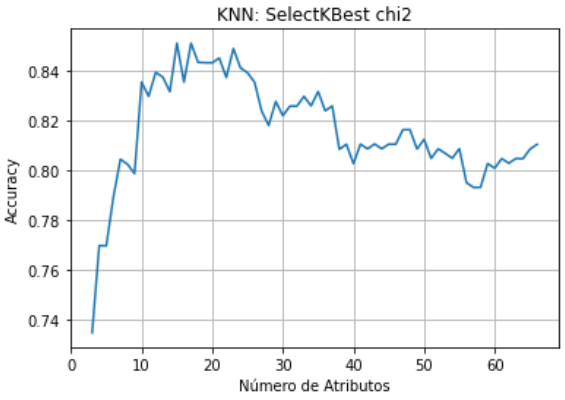
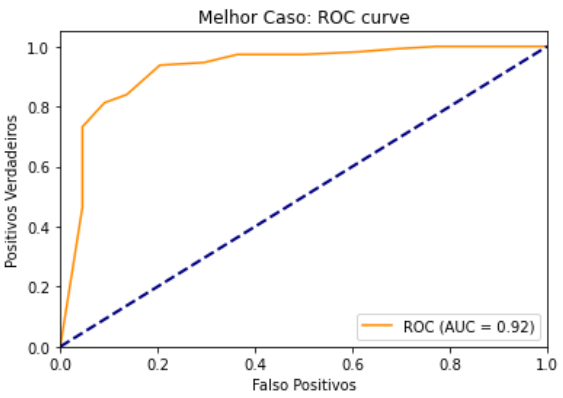


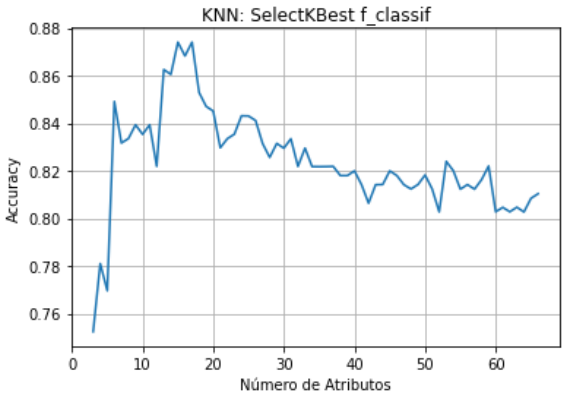
Melhor Caso: 8 Atributos

(Fonte: Elaborado pelo Autor)

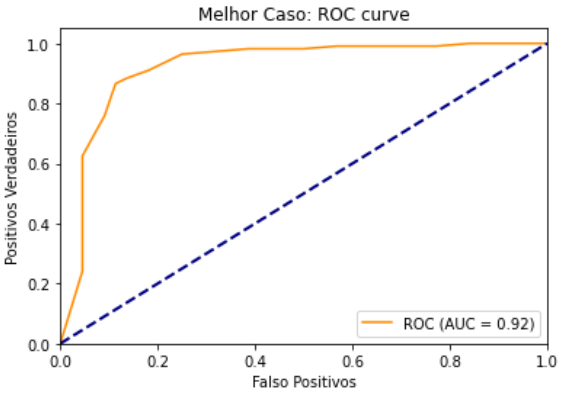
A.3 NearestNeighbors

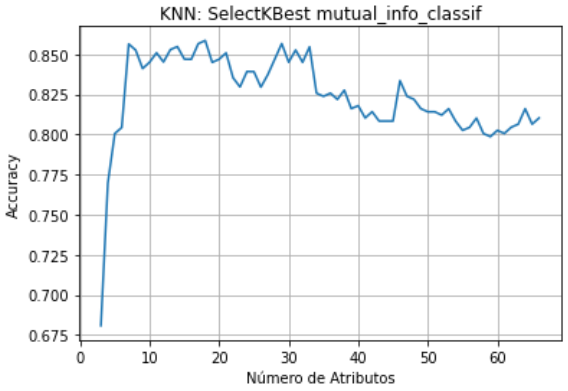
Tabela 20: Resultados da classificação KNN durante a análise exploratória utilizando MinMaxScaler.

Resultados - MinMaxScaler	
 Melhor Caso: 36 Atributos (x = 0.016)	
 Melhor Caso: 15 Atributos	

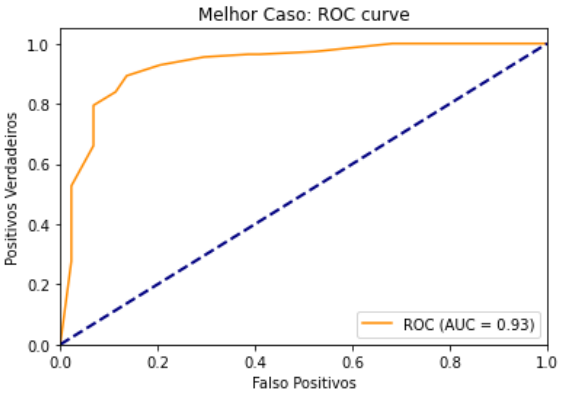


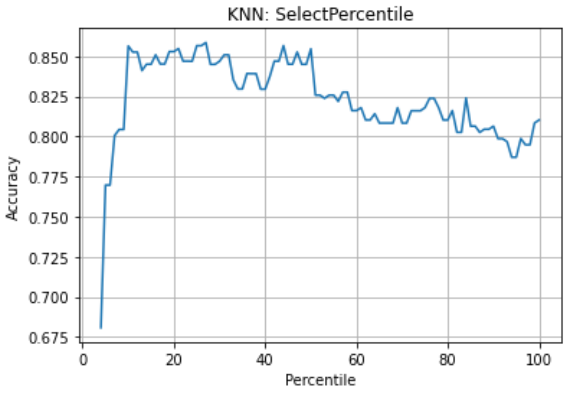
Melhor Caso: 15 Atributos



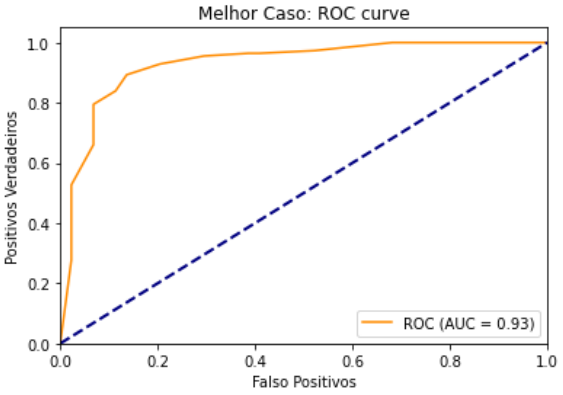


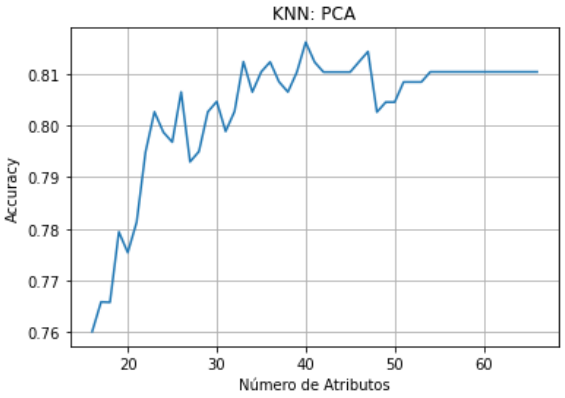
Melhor Caso: 7 Atributos



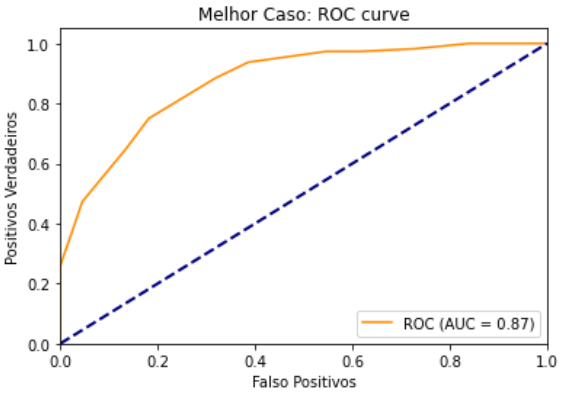


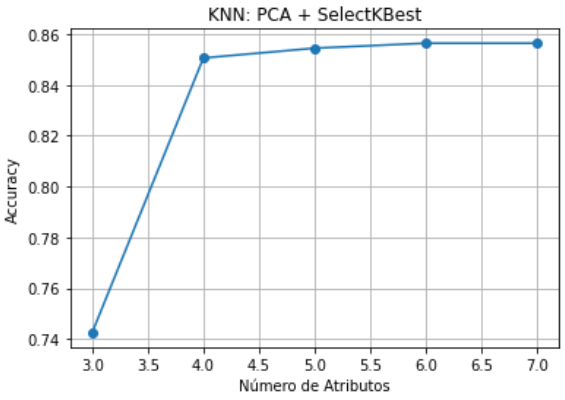
Melhor Caso: 7 Atributos (x = 10)



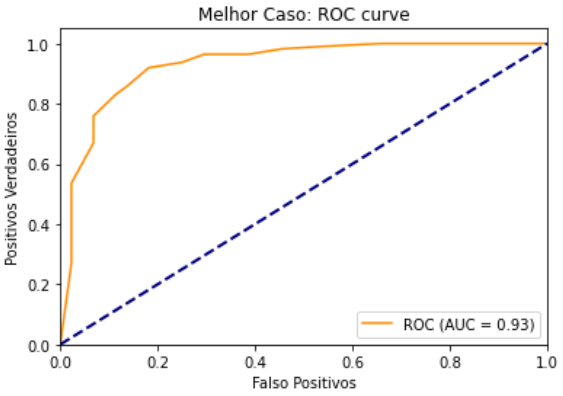


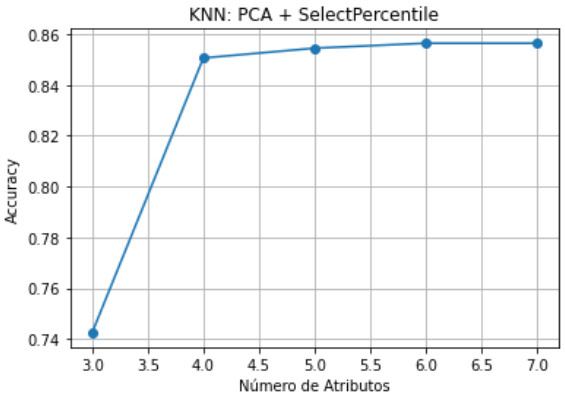
Melhor Caso: 40 Atributos



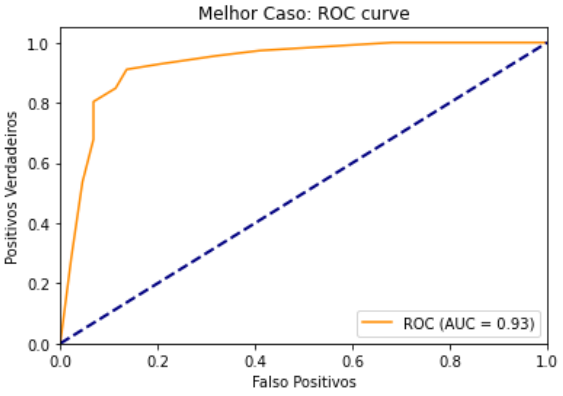


Melhor Caso: 6 Atributos



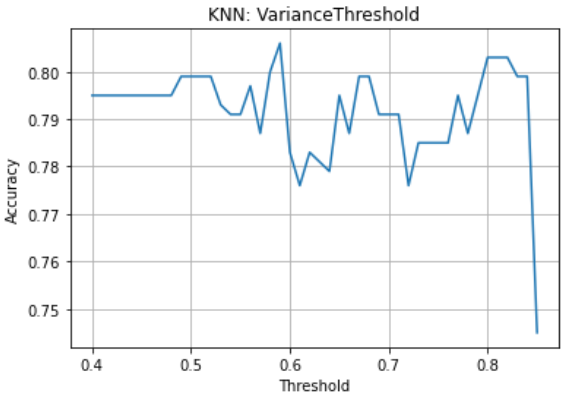
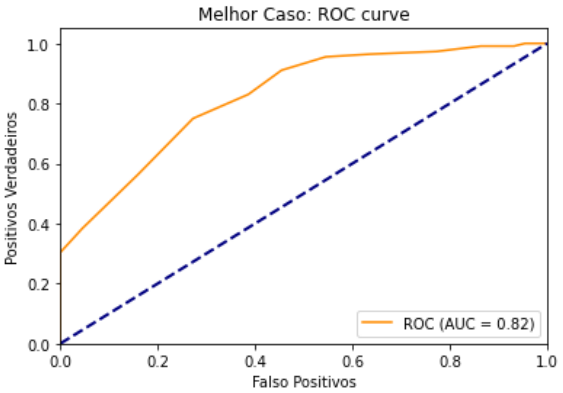
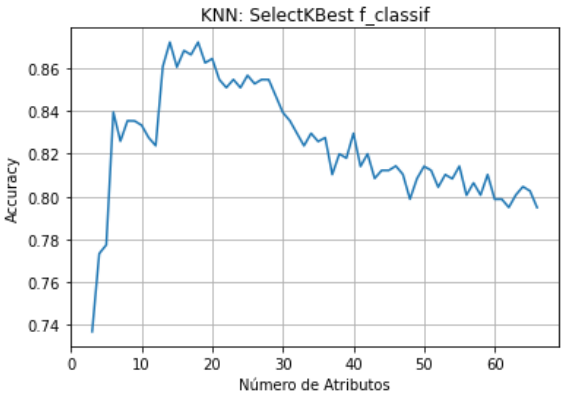
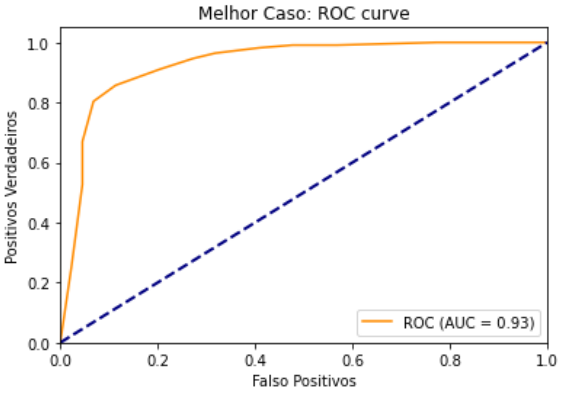
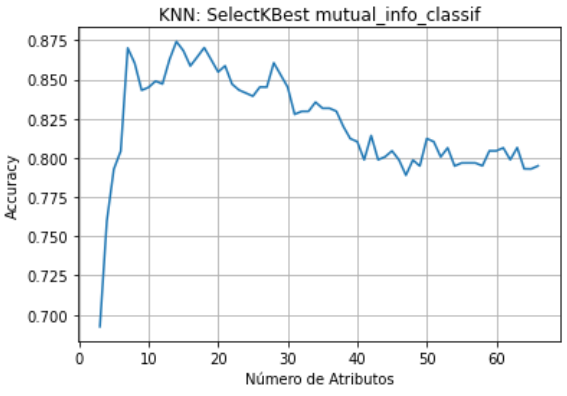
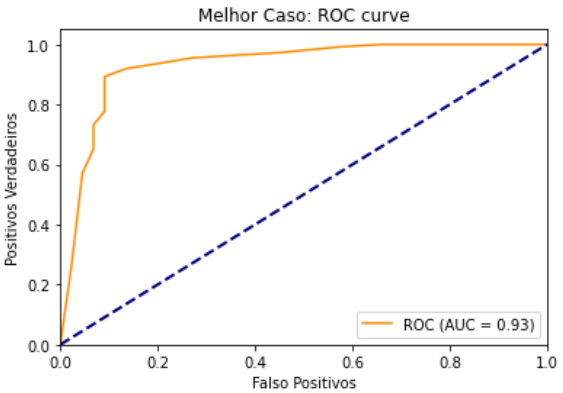


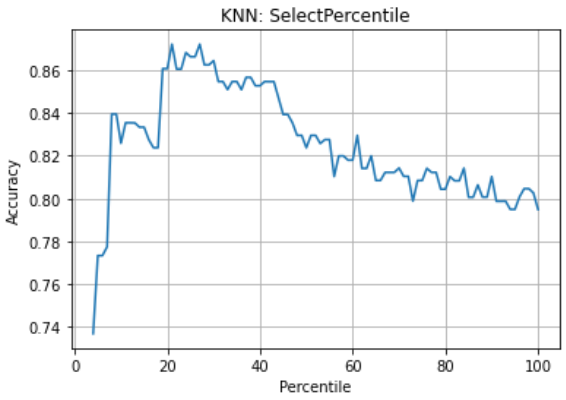
Melhor Caso: 6 Atributos



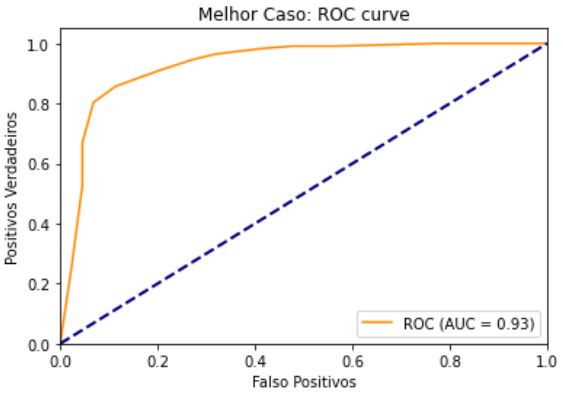
(Fonte: Elaborado pelo Autor)

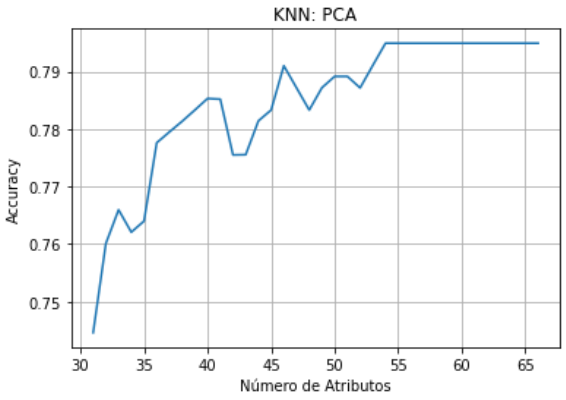
Tabela 21: Resultados da classificação KNN durante a análise exploratória utilizando RobustScaler.

Resultados - RobustScaler	
 Melhor Caso: 12 Atributos (x = 0.8)	
 Melhor Caso: 14 Atributos	
 Melhor Caso: 7 Atributos	

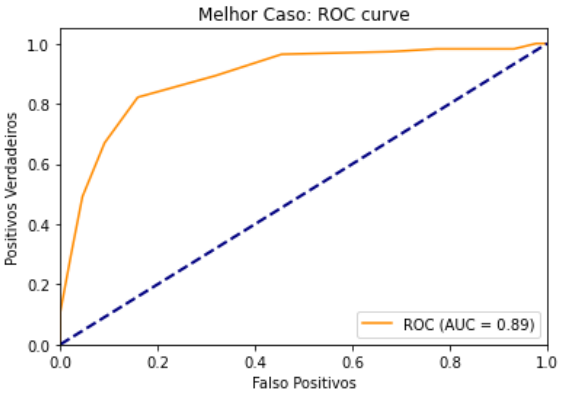


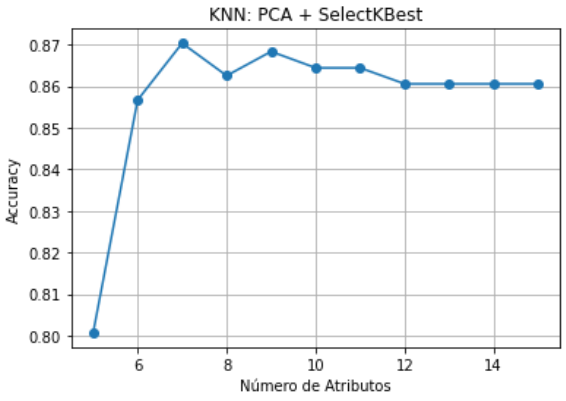
Melhor Caso: 14 Atributos (x = 21)



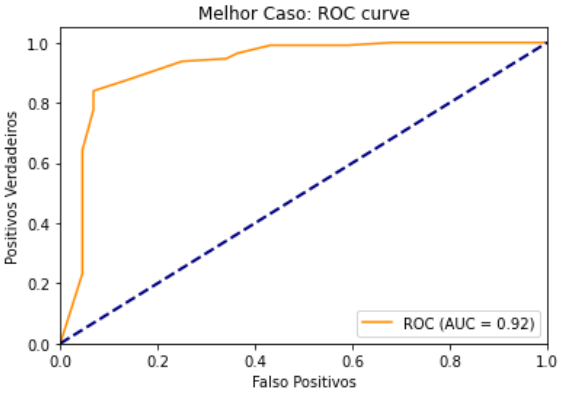


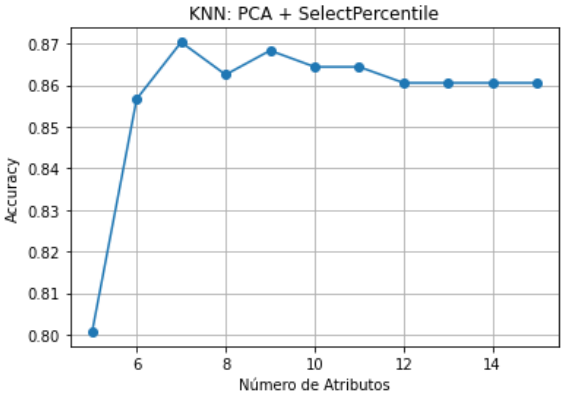
Melhor Caso: 46 Atributos

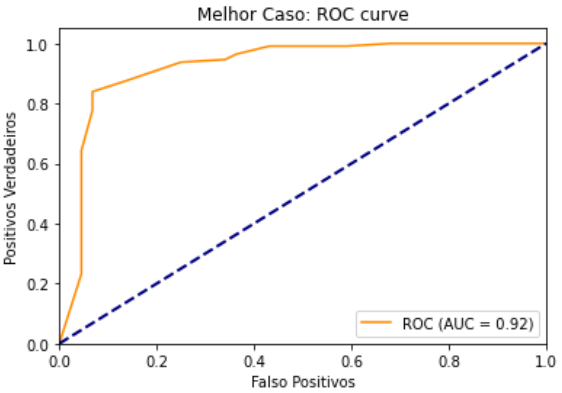




Melhor Caso: 7 Atributos





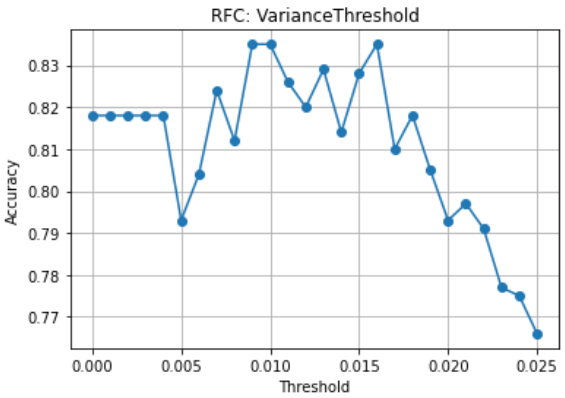
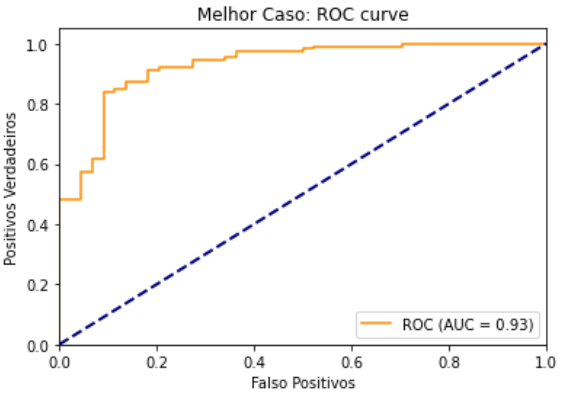
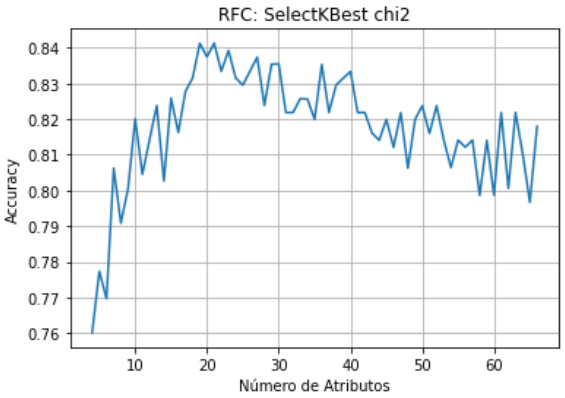
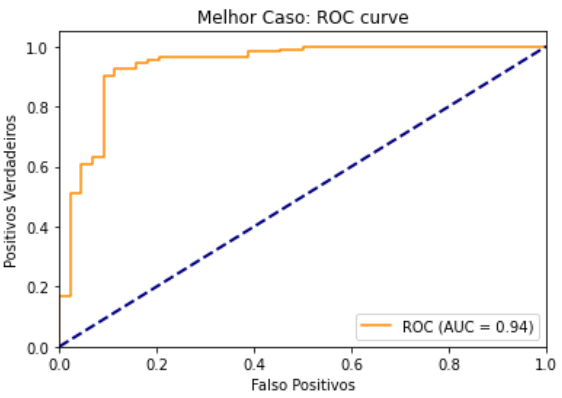


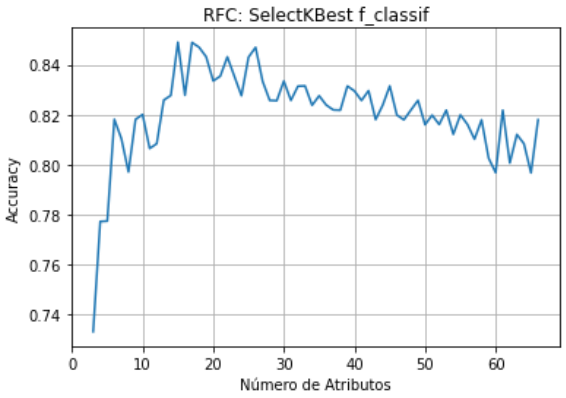
Melhor Caso: 7 Atributos

(Fonte: Elaborado pelo Autor)

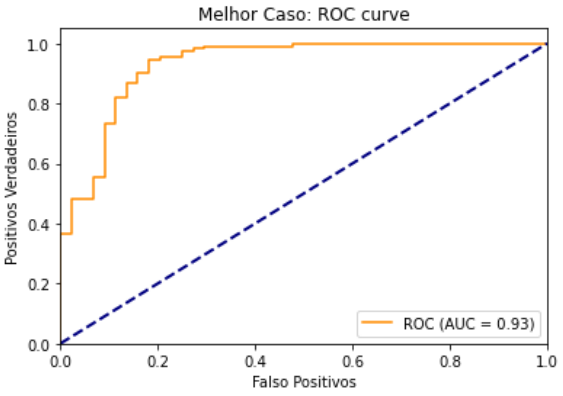
A.4 RandomForestClassifier

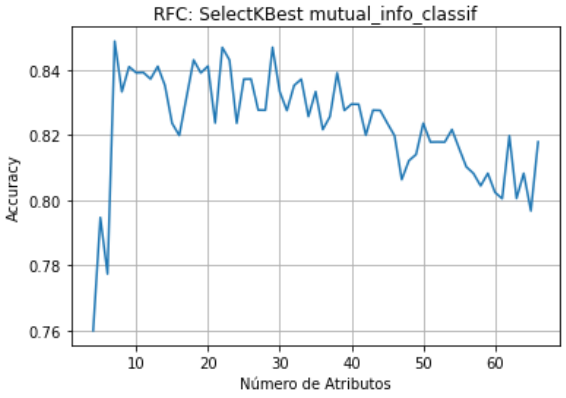
Tabela 22: Resultados da classificação RFC durante a análise exploratória utilizando MinMaxScaler.

Resultados - MinMaxScaler	
 Melhor Caso: 50 Atributos (x = 0.009)	
 Melhor Caso: 19 Atributos	

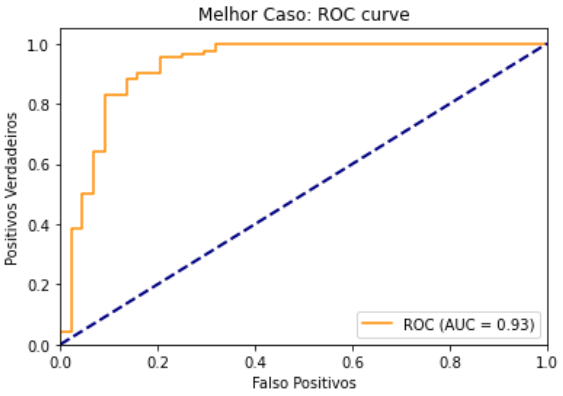


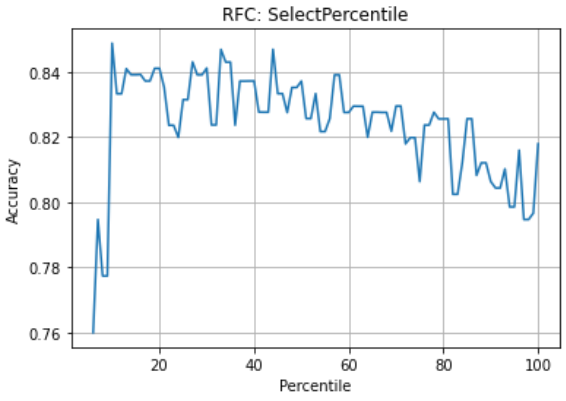
Melhor Caso: 15 Atributos



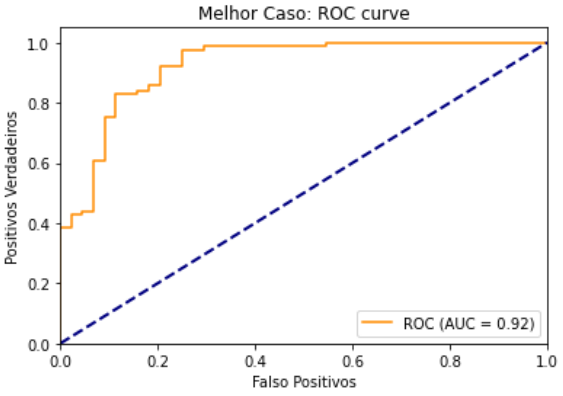


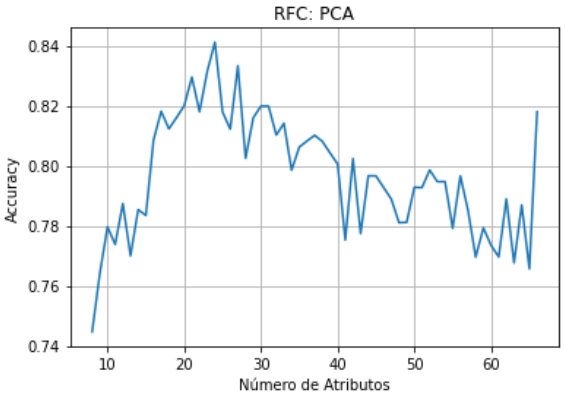
Melhor Caso: 7 Atributos



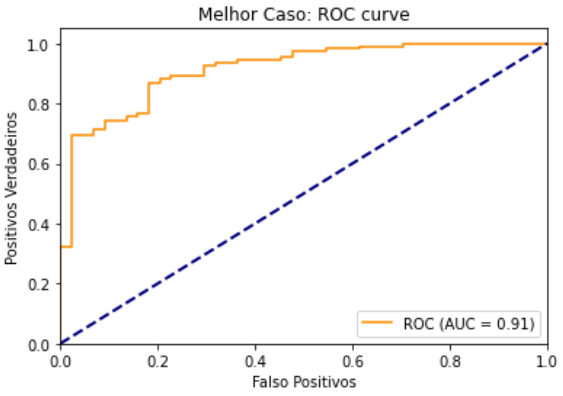


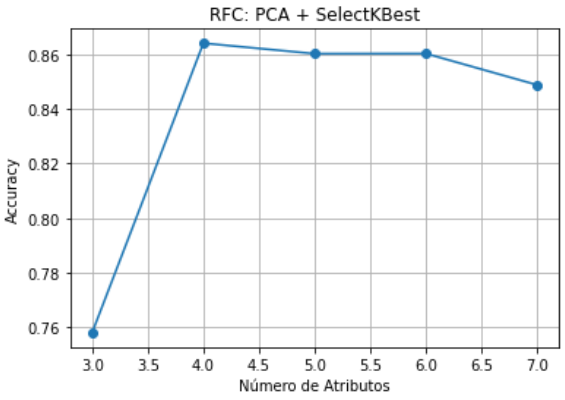
Melhor Caso: 7 Atributos (x = 22)



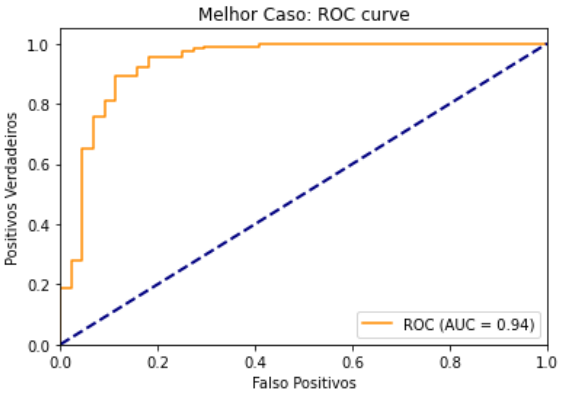


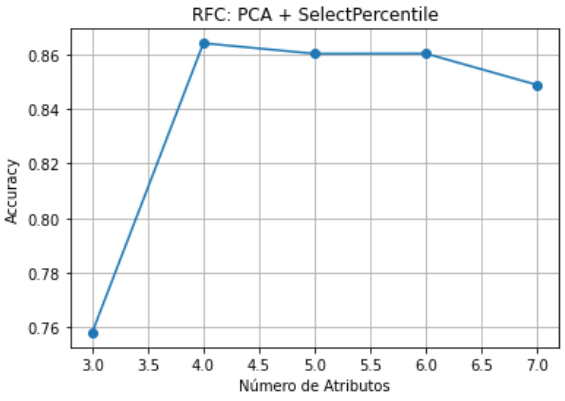
Melhor Caso: 24 Atributos



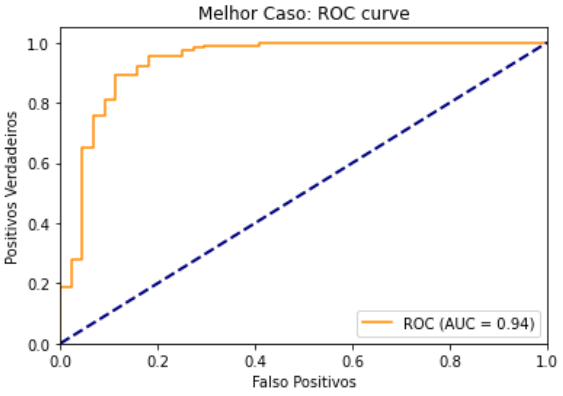


Melhor Caso: 4 Atributos



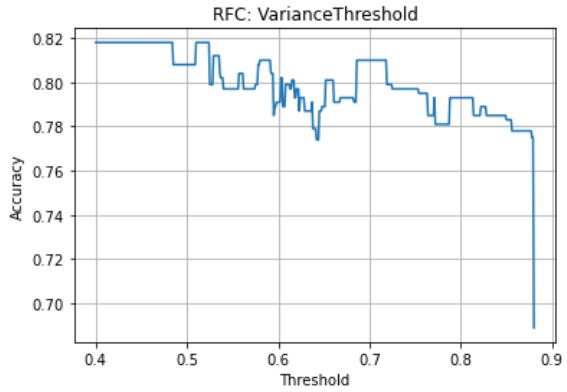
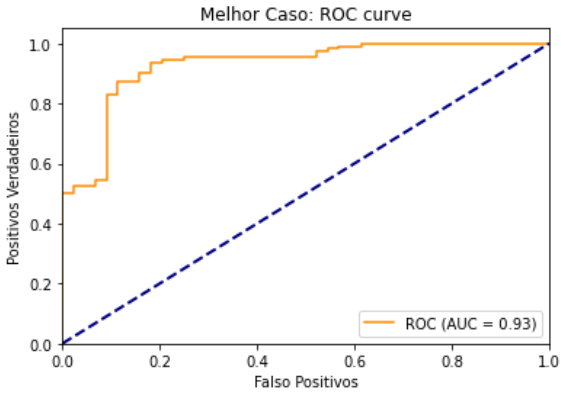
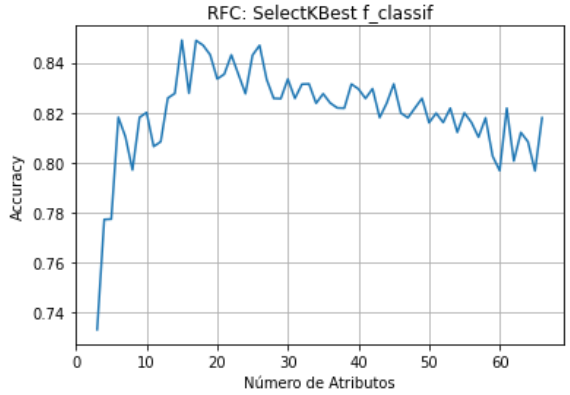
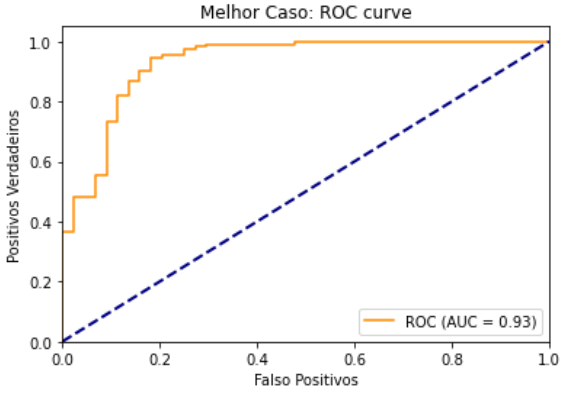
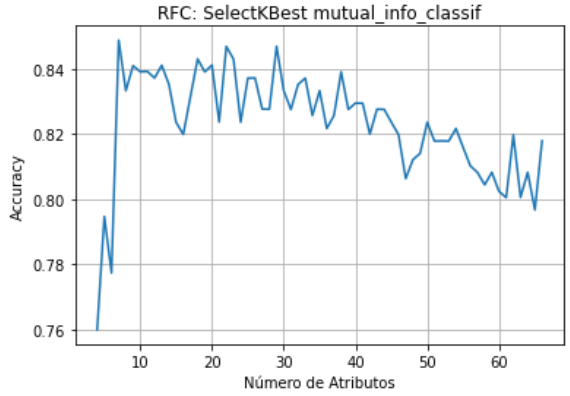
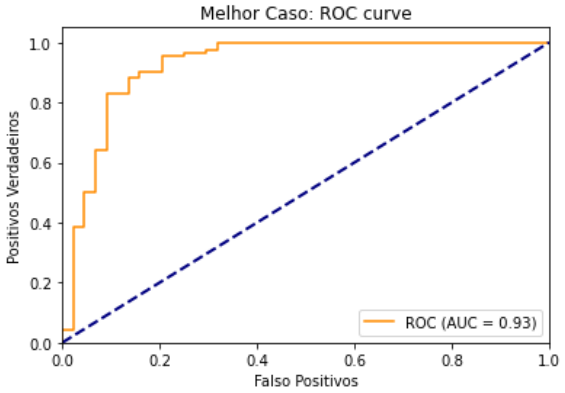


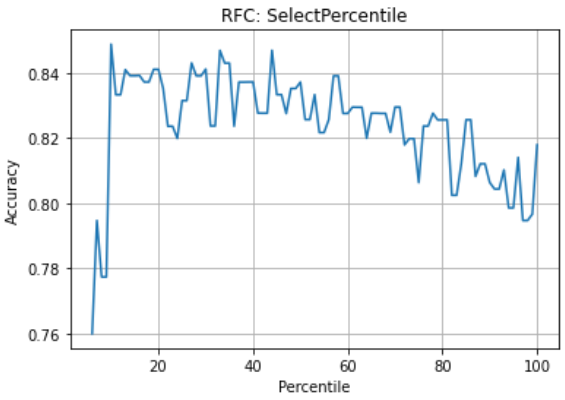
Melhor Caso: 4 Atributos

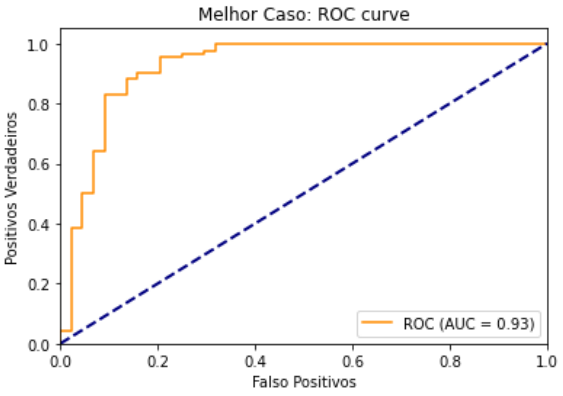


(Fonte: Elaborado pelo Autor)

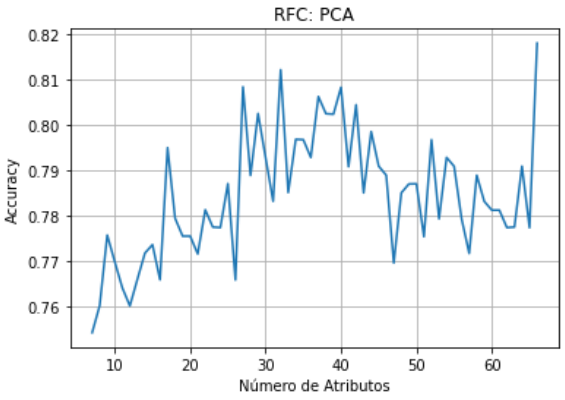
Tabela 23: Resultados da classificação RFC durante a análise exploratória utilizando RobustScaler.

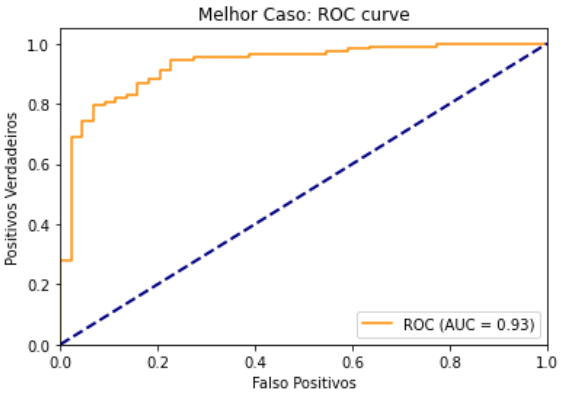
Resultados - RobustScaler	
 Melhor Caso: 21 Atributos (x = 0.52)	
 Melhor Caso: 15 Atributos	
 Melhor Caso: 7 Atributos	



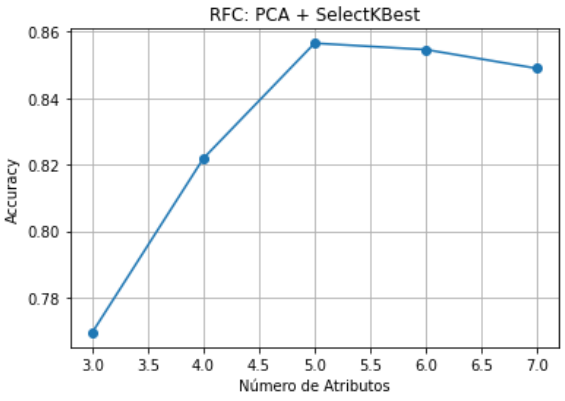


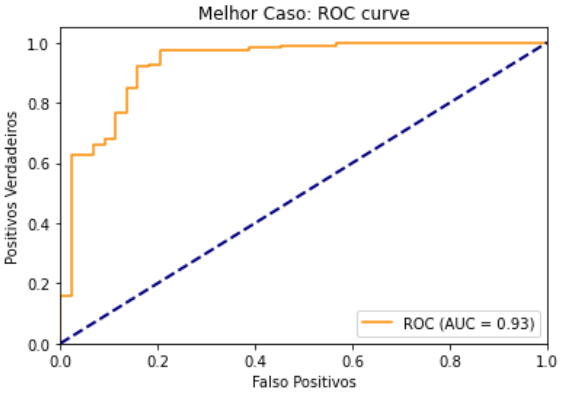
Melhor Caso: 7 Atributos (x = 10)



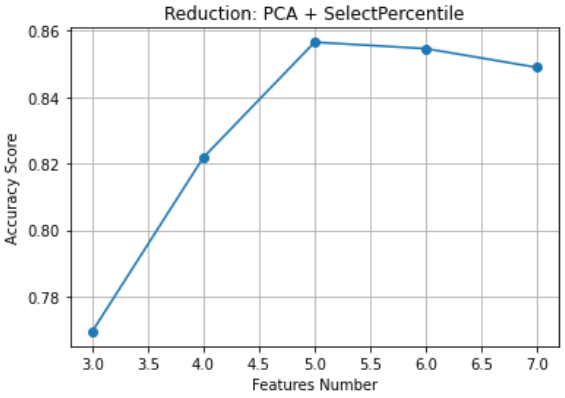


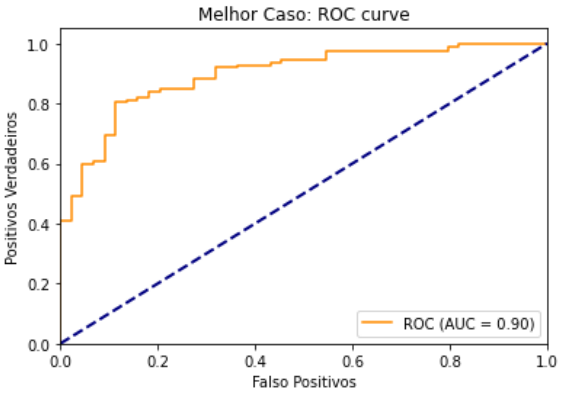
Melhor Caso: 32 Atributos





Melhor Caso: 5 Atributos



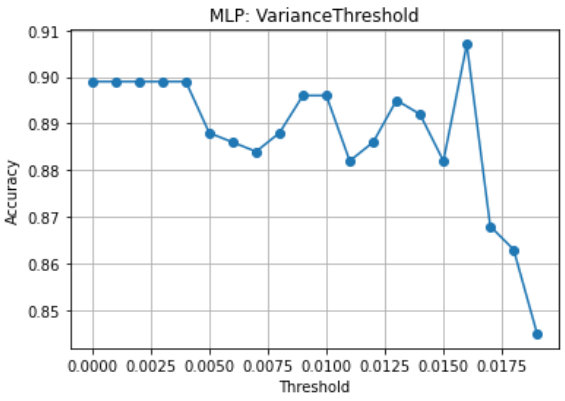
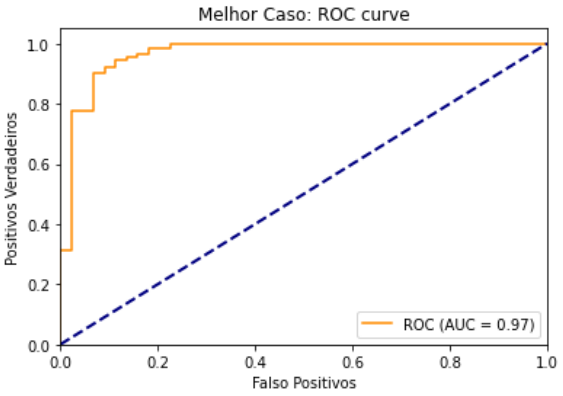
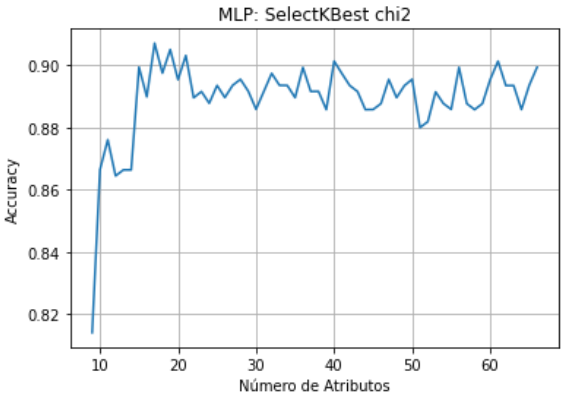
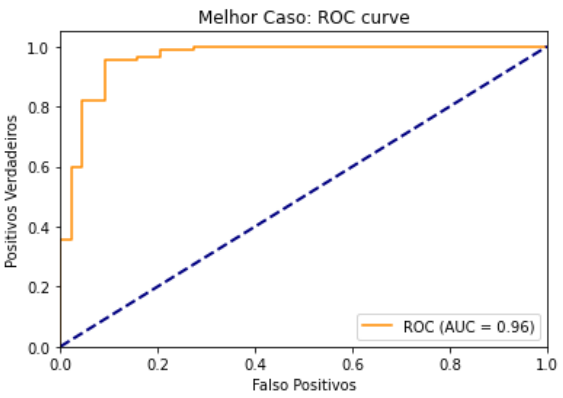


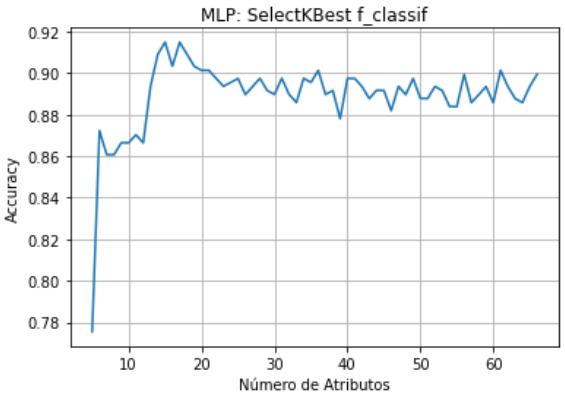
Melhor Caso: 5 Atributos

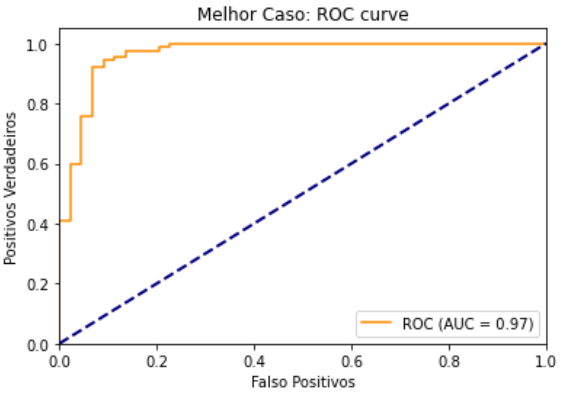
(Fonte: Elaborado pelo Autor)

A.5 MLPClassifier

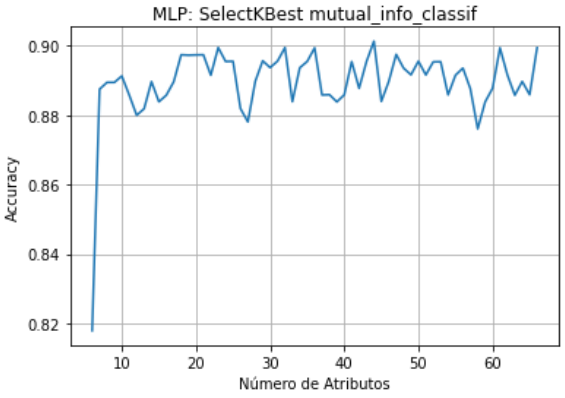
Tabela 24: Resultados da classificação MLP durante a análise exploratória utilizando MinMaxScaler.

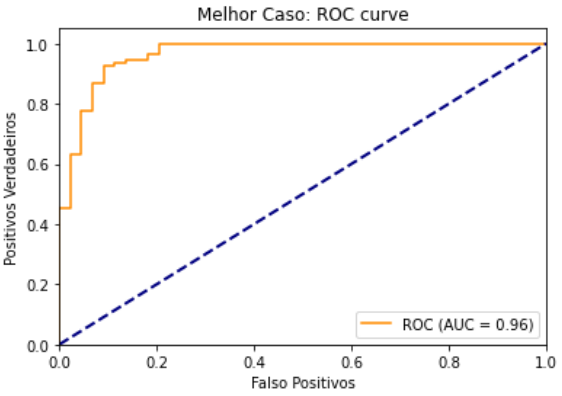
Resultados - MinMaxScaler	
 Melhor Caso: 36 Atributos (x = 0.016)	
 Melhor Caso: 17 Atributos	



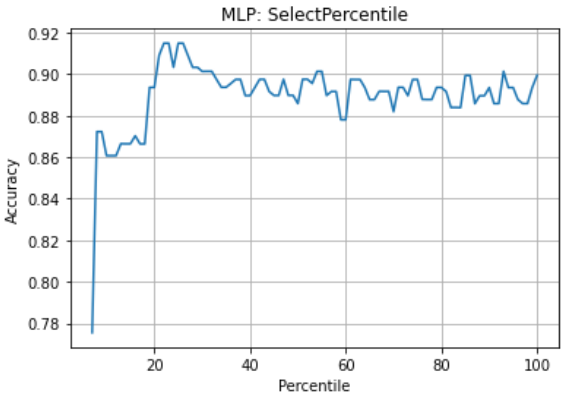


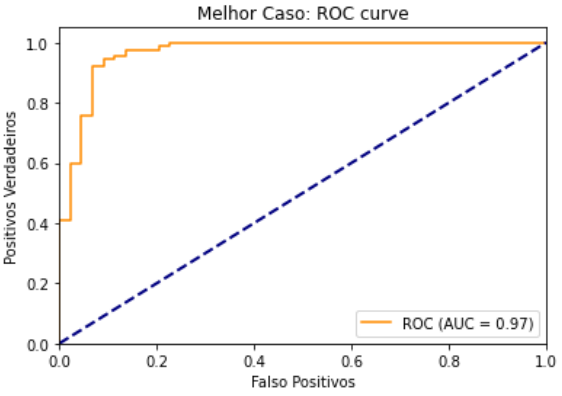
Melhor Caso: 14 Atributos



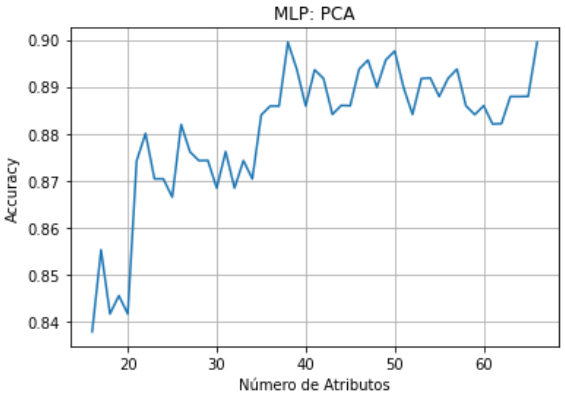


Melhor Caso: 19 Atributos

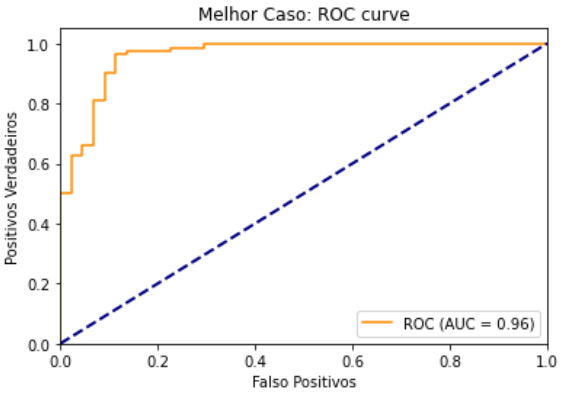


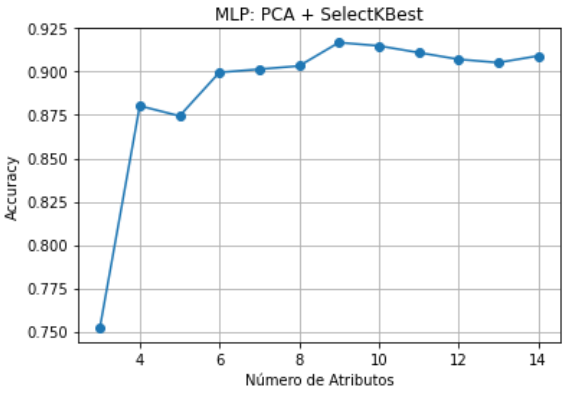


Melhor Caso: 14 Atributos (x = 21)

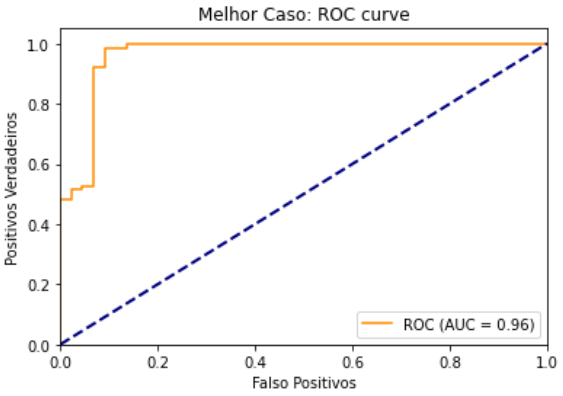


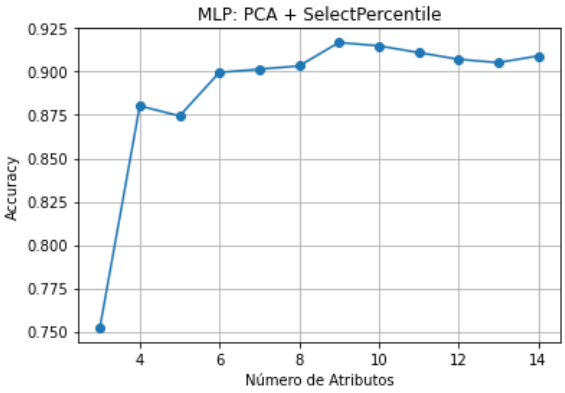
Melhor Caso: 38 Atributos



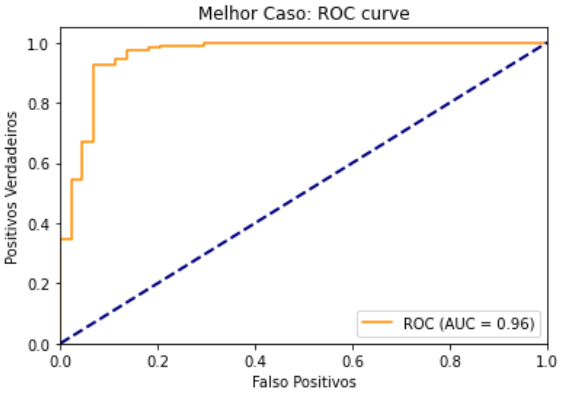


Melhor Caso: 9 Atributos



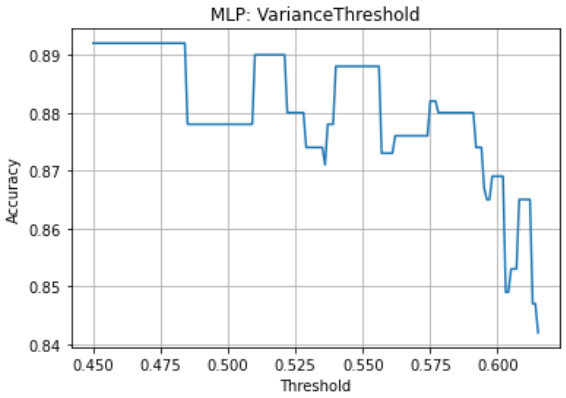
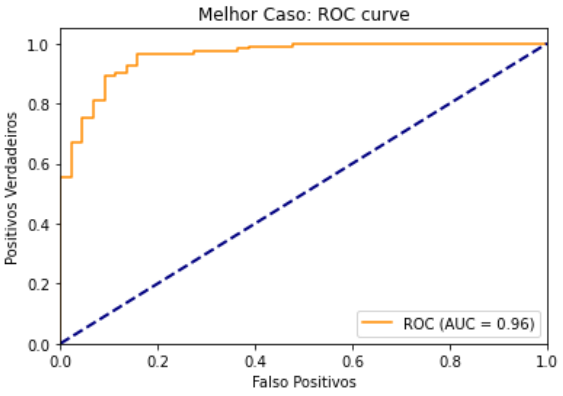
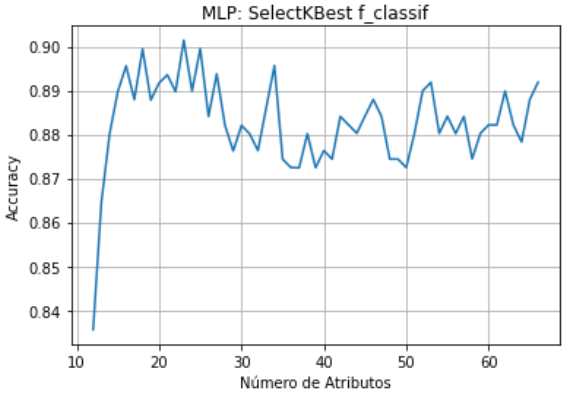
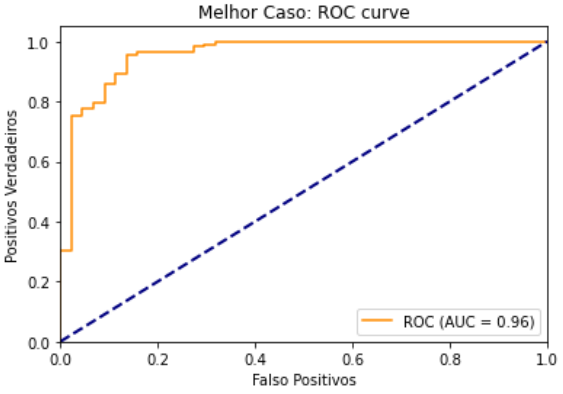
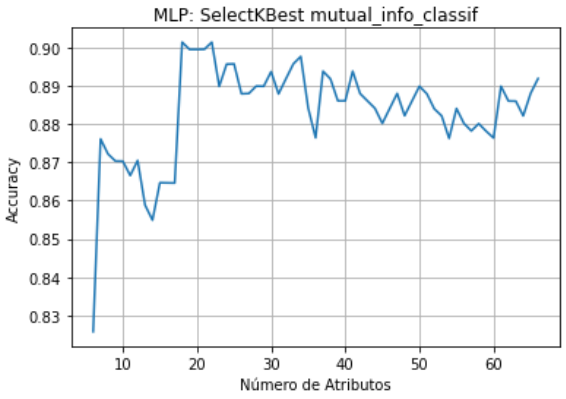
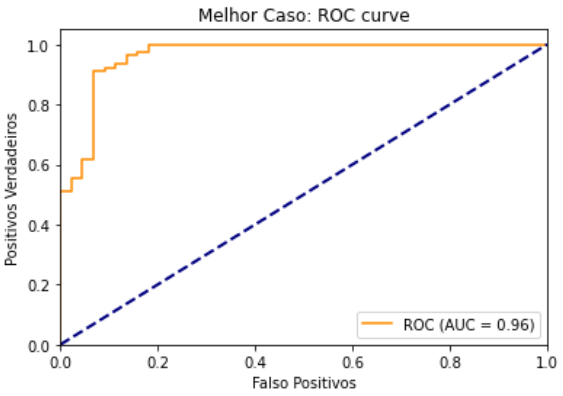


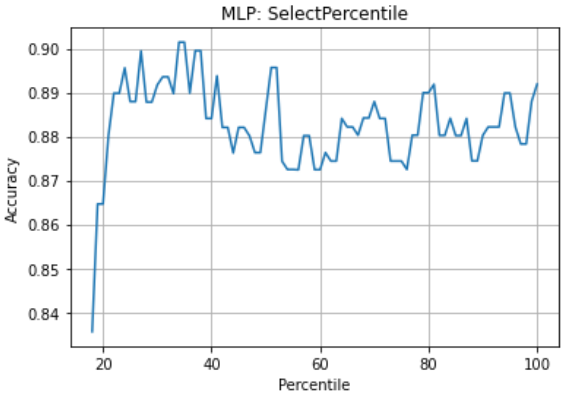
Melhor Caso: 9 Atributos



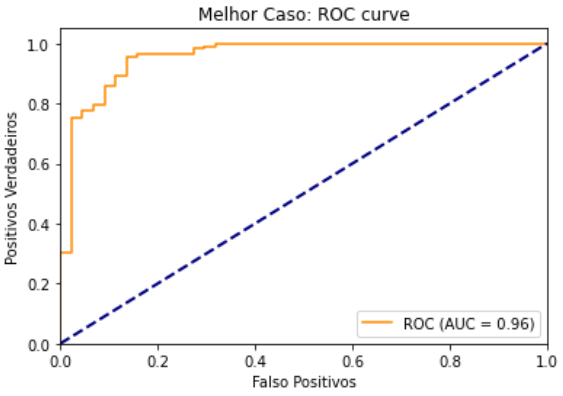
(Fonte: Elaborado pelo Autor)

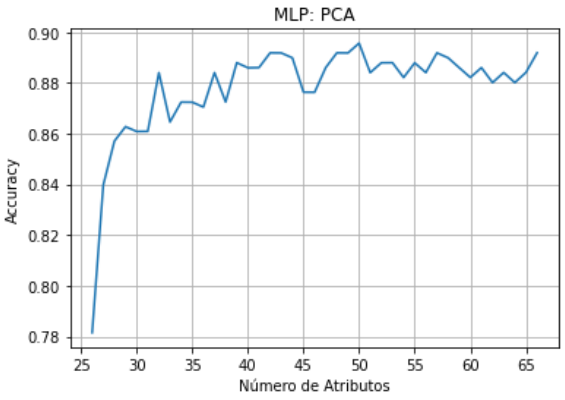
Tabela 25: Resultados da classificação MLP durante a análise exploratória utilizando RobustScaler.

Resultados - RobustScaler	
 Melhor Caso: 57 Atributos (x = 0.55)	
 Melhor Caso: 18 Atributos	
 Melhor Caso: 19 Atributos	

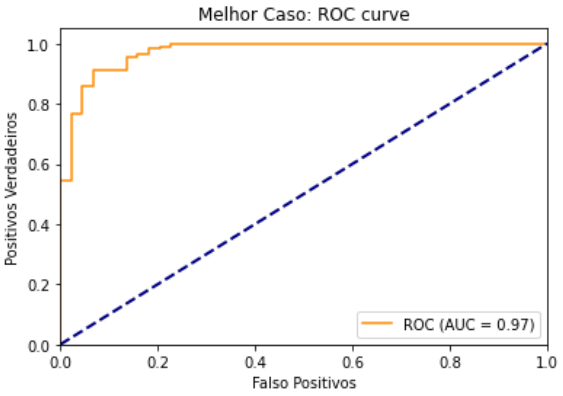


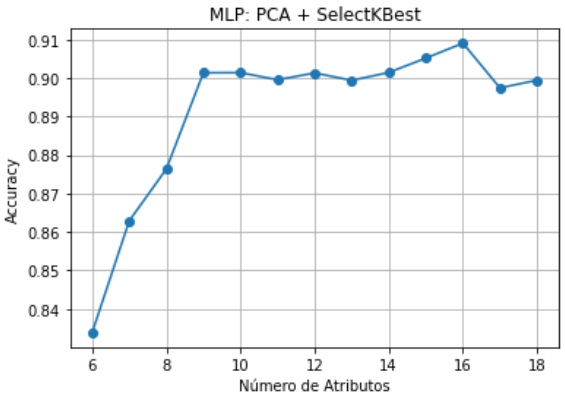
Melhor Caso: 18 Atributos (x = 27)



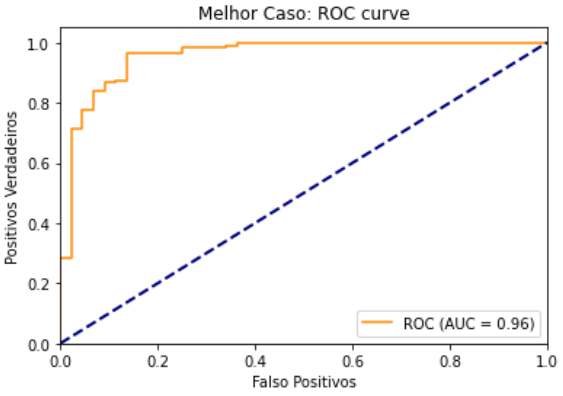


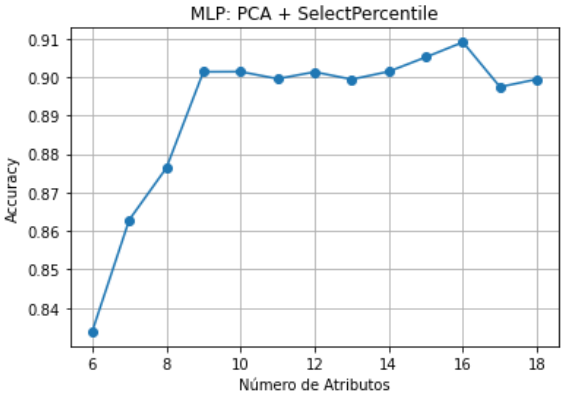
Melhor Caso: 42 Atributos

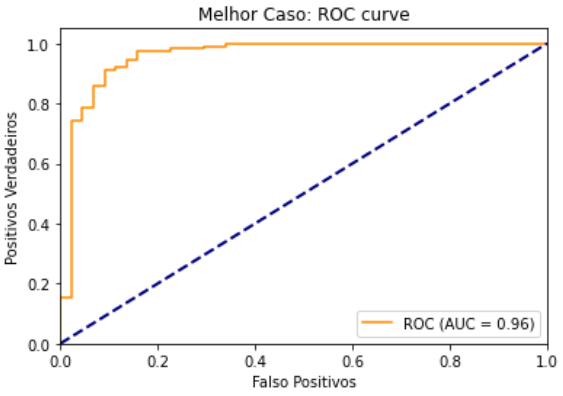




Melhor Caso: 16 Atributos







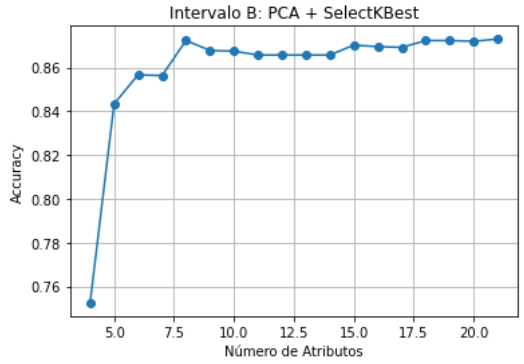
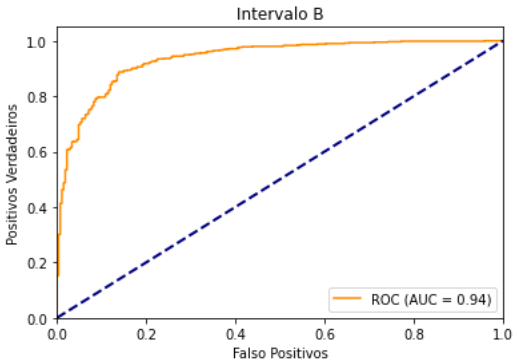
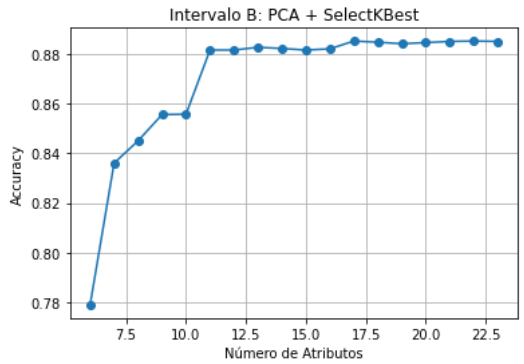
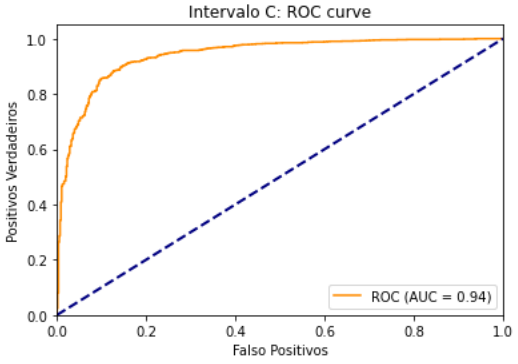
Melhor Caso: 16 Atributos

(Fonte: Elaborado pelo Autor)

B. Experimento 2

B.1 SVC

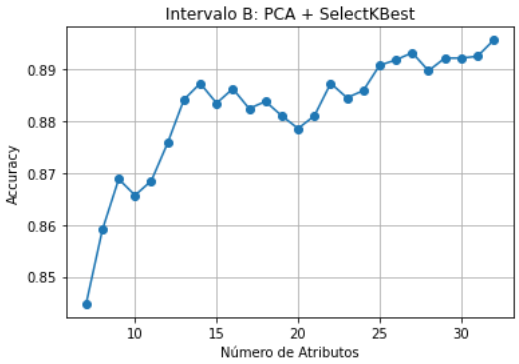
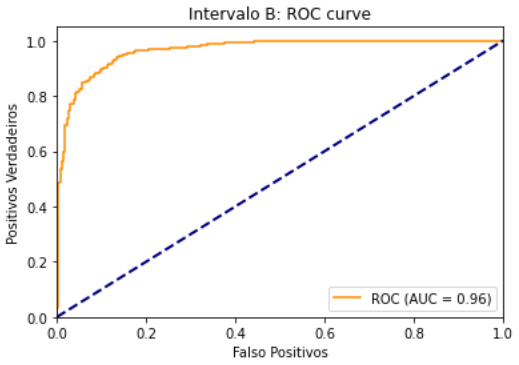
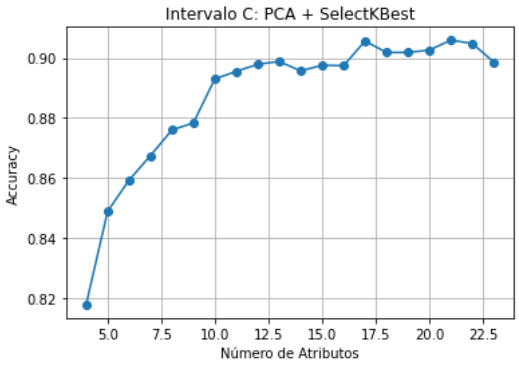
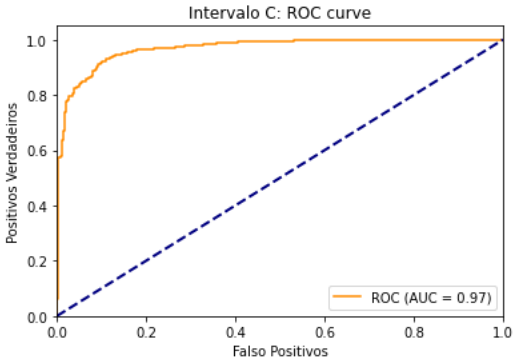
Tabela 26: Resultados da classificação SVC durante o Experimento 2.

Resultados	
 Melhor Caso: 8 Atributos	
 Melhor Caso: 11 Atributos	

(Fonte: Elaborado pelo Autor)

B.2 MLPClassifier

Tabela 27: Resultados da classificação SVC durante o Experimento 2.

Resultados	
 Melhor Caso: 27 Atributos	
 Melhor Caso: 17 Atributos	

(Fonte: Elaborado pelo Autor)