Influência de diferentes técnicas de extração e seleção de atributos na acurácia de sistemas preditivos

1st Tiago Pereira de Faria
Faculdade de Computação (FACOM)
Universidade Federal de Uberlândia (UFU)
Uberlândia, Brasil
tiago.pereira.faria@gmail.com

2nd Márcio Antônio de Freitas Junior
Faculdade de Computação (FACOM)
Universidade Federal de Uberlândia (UFU)
Uberlândia, Brasil
marcio55afr@gmail.com

***Abstract*—Este trabalho utiliza técnicas de extração, seleção de atributos e transformação do espaço de características a fim de melhorar os resultados obtidos pelo modelo proposto por Olivier Grellier na competição PLAsTiCC sitiada no Kaggle. Avaliou-se também outros modelos preditivos da literatura em conjunto com estas técnicas de pré-processamento para compará-las e utilizar a que for mais conveniente. A extração que obteve melhor desempenho alcançou um Score de 1.22910 enquanto a solução original apenas 1.42560. É possível observar variações na acurácia de alguns outros modelos classificadores ao mudar o numero de características utilizadas porém, a maioria dos classificadores tiverem resultados melhores ou próximo do seu melhor quando utilizado todas as características extraídas.**

***Index Terms*—Extração de atributos, Seleção de Atributos, Aprendizado de Máquina, PLAsTiCC, Astronomia**

I. Introduction

Em 2020 terá início às primeiras imagens do telescópio que irá revolucionar o entendimento dos homens sobre o espaço, descobrindo e mensurando milhões de objetos variantes no tempo, o *Large Synoptic Survey Telescope* (LSST) [1]. O LSST está sendo construído ao norte do deserto do Atacama, na montanha Cerro Panchon situada no Chile. O objetivo principal deste telescópio é conduzir uma pesquisa de 10 anos sobre o céu, que irá gerar um conjunto de 200 *petabytes* de imagens e informações, direcionado a responder as questões mais proeminentes sobre a estrutura e a evolução do universo e dos objetos neles existentes [2].

Um dos maiores desafios provenientes do LSST é construir o *software* que irá processar os dados gerados por ele. São mais de 30 *terabytes* de medições, todas as noites, que serão processados e armazenados para produzir o maior *dataset* não-proprietário do mundo [2]. Uma das tarefas desse *software* é classificar cada objeto observado pelo telescópio em um tempo hábil utilizando de recursos factíveis. Para isto, o *LSST Dark Energy Science Collaboration* [3] e o *LSST Transients and Variable Stars Science Collaboration* [4] desenvolveram o *Photometric LSST Astronomical Time-series Classification Challenge* (PLAsTiCC), um desafio publicado no *site* de competições Kaggle que premiou o criador do melhor algoritmo de classificação sobre o *dataset* de séries temporais astronômicas que simulam as futuras observações do telescópio LSST.

A partir deste ponto, este trabalho utiliza técnicas de extração e seleção de atributos e transformação do espaço de características a fim de melhorar os resultados obtidos pelo modelo proposto pelo [5]. Além disso, avaliamos também outros modelos preditivos da literatura em conjunto com estas técnicas de pré-processamento.

II. Extração de características

A extração de características foi baseada no trabalho [6] que utiliza características citadas pelos melhores competidores além de criar separar a extração para cada *passband* diferente, um atributo categórico e nominal do *dataset*, de cada amostra. Assim, o número de características extraídas foi de cerca de 30, como no trabalho do Grellier, para 109. Após isso ainda foi aplicado a extração PCA (*Principal Component Analysis*) contudo os resultados foram piores.

III. Seleção de características

Como de modo geral o desempenho de um classificador tende a se degradar a partir de um determinado número de atributos, e a quantidade de atributos foi engradecida, o problema da maldição da dimensionalidade precisou ser cogitado. Para verificar e contornar este problema, os atributos foram ordenados pela sua relevância de acordo com 3 algoritmos de seleção diferentes, cada um aplicado em outros 4 algoritmos classificadores. Cada algoritmo de seleção selecionou os atributos mais relevantes para testar a acurácia em cada classificador para compararmos os seletores. Além disso, o número de atributos selecionados foi aumentado gradativamente até usar todos para observar o número ideal das características extraídas.

Os 3 tipos de seleções utilizados foram: seleção dos atributos que mais influenciam os componentes criados pelo PCA, a seleção baseada na Análise da Variância e a seleção baseada na Estimação de Informação Mútua do pacote SKlearn no Python e por fim utilizando todos os atributos. Os algoritmos de classificação testados neste processo foram: O KNN, a

SVM, a Random Forest e uma rede Multi-Layer Perceptron. O algoritmo de classificação utilizado pelo Grellier é o LGBM.

A. Seleção baseada na relevância de atributos nos componentes do PCA

Como mostrado por [7], os componentes criados pelo algoritmo de análise de componentes principais (do inglês Principal Component Analises, PCA) podem ser utilizados para determinar a relevância de cada atributo da base de dados. O algoritmo de [7] mede a influência que cada atributo tem em cada componente criado pelo PCA, e então, os ordena com base nessa métrica. O nosso algoritmo, então, seleciona os k primeiros elementos dessa lista para compor as características usadas para treinar o modelo e classificar as amostras.

B. Seleção baseada na Analise da Variância (ANOVA do inglês Analysis of Variance)

A análise de variância é um teste estatístico para medir a diferença da média de duas ou mais populações baseado em um conjunto de amostras. Para cada atributo, o algoritmo realiza um teste de hipótese de que a média de seus valores para cada classe da base de dados é igual. O grau de certeza com que podemos rejeitar esta hipótese nos indica a diferença que este atributo tem entre as classes. Este grau de certeza então, é utilizado para avaliar a importância do atributo na classificação da base de dados.

C. Seleção baseada na estimação de Informação Mútua

Informação Mútua mede a quantidade de informação que se pode obter de uma variável baseado no valor de outra variável. Ou seja, ela mede o quão dependente duas variáveis são. A técnica de seleção utiliza de um método não paramétrico (o *KNN*) para selecionar outras amostras próximas que são utilizadas para calcular o grau de dependência de cada atributo. O algoritmo assume que quanto mais dependente um atributo for de sua classe, maior é a sua importância para um classificador. Então são selecionados os atributos de maior informação mútua com a classe da amostra.

IV. PCA

PCA ou *Principal Component Analysis* é um procedimento matemático que utiliza autovetores e autovalores para fazer uma converter um conjunto de observações de variáveis possivelmente correlacionadas num conjunto de variáveis não correlacionadas chamadas de componentes principais. O resultado são novos atributos de mesma quantidade de atributos iniciais.

V. Modelos Preditivos

Além dos diferentes métodos de seleção de características, também foram avaliados 5 diferentes métodos de aprendizado supervisionado: Random Forest, Máquina de vetores de suporte, Perceptron multi camadas, *KNN* e *LGBM*. Nesta sessão é apresentado de forma geral, o funcionamento de cada um destes algoritmos.

A. Random Forest

Random Forest é um algoritmo de *ensemble learning*, que utiliza vários classificadores juntos para melhorar a performance, construindo um conjunto de árvores de decisões em tempo de treinamento e utiliza a moda das classificações como resposta ou média das regressões dependendo do problema. Esse algoritmo evita o *overfitting* observado em árvores de decisões.

B. Máquina de vetores de suporte

A máquina de vetores de suporte ou SVM utiliza da distribuição dos objetos classificados para predizer os novos. A partir dos conjuntos de dados de duas classes é encontrado um hiperplano cuja margem, criada pelos elementos mais próximos dentre as duas classes, separe linearmente as duas classes. Os coeficientes do hiperplano são encontrados através da diferença entre os vetores dos objetos mais próximos e a definição de que a margem tenha módulo igual a 1.

Existem duas abordagens para que a SVM classifique diversas classes. Uma é criar um hiperplano que separa uma classe de todas as outras, para todas as classes. Ou então, cada classificador classifica uma amostra dentre um par de classes possíveis, assim é necessário n*(n-1)/2 classificadores para **n** classes enquanto o primeiro utiliza **n** classificadores.

C. Perceptron Multi Camadas

A rede Perceptron Multi Camadas combinam vários perceptrons em várias camadas formando estruturas mais complexas para abordar problemas que um único perceptron (neurônio) não consegue resolver. Os objetos são passados para a primeira camada que processa e retorna valores para a segunda camada até a última camada que retorna a saída do algoritmo. O treinamento se inicia com uma rede de neurônios com pesos aleatórios se de acordo com a resposta da rede e a classe correta de um dado objeto é executado o chamado *backpropagation*. Atualizando os pesos da ultima camada para a primeira.

D. KNN

O KNN (*K-Nearest Neighbours*) é um algoritmo de classificação supervisionada não paramétrico. Ele precisa de uma base de dados de treino e de teste. Após a leitura dos dados de treino, a classificação de novos objetos é feita através da distância, observando os K objetos já classificados mais próximos. A classe do novo objeto será a classe com maior número dentre os K objetos mais próximos a distância calculada geralmente é a euclidiana.

E. LGBM

Light Boosting Gradient Model (LGBM) é um algoritmo baseado em árvores de decisão que aplica um *boosting* gradiente. Foi designado a ser distribuído e eficiente com as seguintes vantagens: alta velocidade e eficiência no treinamento; baixo uso de memória; melhor acurácia; suporte para aprendizado paralelo e em GPU; e capaz de ligar com grandes escalas de dados [8].

O *boosting* presente no LGBM consiste em utilizar várias cópias do classificador para aprenderem sobre o conjunto de dados de forma interativa. Para classificar um resultado, é dado um peso para cada predição desses classificadores e então é a calculada a predição final. Para o aprendizado ocorre algo parecido. Cada tupla de treinamento, formado por um dado e sua classe, possui um peso. Assim que o primeiro classificador executa o processo de aprendizado, os pesos são alterados de modo a aumentar a atenção para as tuplas classificadas erroneamente por esse classificador. Assim, o próximo terá maiores chances de acertar o que o primeiro classificador e assim sucessivamente. O último classificador impulsionado (*boosted*), combina os votos de cada classificador individualmente, onde os pesos de cada classificador é uma função de sua acurácia [9].

Quando voltado para classificar múltiplas classes, o LGBM pode utilizar algumas métricas, e uma delas é através da função objetivo *softmax* [10]. Essa função retorna um vetor que representa a distribuição de probabilidades de uma lista de potenciais resultados. A soma das probabilidades desse vetor é igual a 1.

VI. Metodologia

Para criar um sistema que classifique observações de objetos astronômicos entre 14 diferentes classes, avaliamos diferentes técnicas de extração de características, seleção de características e modelos de predição. foi criado um sistema de extração de características que constrói 109 atributos e este foi comparado ao método de [6], que constrói 33 características no total. para fazer a seleção de atributos, avaliamos 3 técnicas diferentes presentes na biblioteca SkLearn do Python. Também avaliamos a utilização do PCA na base de dados antes do submetimento àos modelos preditivos. Por fim, avaliamos 5 modelos preditivos diferentes e comparamos os seus resultados.

Foi realizado um pré-processamento dos dados para viabilizar a utilização de alguns dos modelos preditivos propostos. Uma normalização foi realizada nos atributos utilizando o algoritmo z-score. Além disso, eventuais valores que não estavam presentes em algumas amostras foram substituidos pela média daquela amostra na base de dados.

A. Parâmetros da Random Forest

Para construir o *ensemble*, foram utilizadas 150 árvores. Não foi realizada nenhuma técnica de póda.

B. Parâmetros da MLP

Foi utilizado uma rede com 2 camadas ocultas, com 5 *perceptrons* em cada. A função de ativação utilizada é a *sigmoide*:

$$f(x) = \frac{1}{1 - e^{-x}} \tag{1}$$

A taxa de aprendizado utilizada foi de 0.001 e o número máximo de épocas foi de 500.

C. Parâmetros da SVM

Para a criação da SVM, utlizamos o *kernel* linear, com C = 0.025 e um número máximo de iterações de 800.

D. Parâmetros do KNN

O KNN foi relizado considerando os 20 vizinhos mais próximos. Nenhuma espécie de peso foi utilizada, então cada vizinho tem o mesmo peso na votação para decidir a classe.

VII. Experimentos

De modo geral, foram feitos dois tipos de experimentos. Um tipo para analisar a seleção de atributos e verificar a existência do problema da maldição da dimensionalidade e outro tipo para analisar a extração de atributos e o quanto ela melhora o desempenho do classificador.

A solução do Grellier foi utilizada para avaliar a extração de características de modo que utilizando o mesmo classificador e parâmetros, ao mudar as características de cada amostra, é possível observar se determinada extração melhora ou piora os resultados da classificação. Foi comparado os resultados da extração de características da solução do Olivier Grellier, as características originais mais um acréscimo de 5 atributos ainda não utilizados, esse novo conjunto de características aplicado ao PCA e o conjunto novo separando as características para cada *passband* diferente (não aplicando o PCA). O resultados é mostrado na Tabela I

Para avaliar o desempenho de cada combinação de seleção de atributo mais um modelo preditivo, realizamos testes utilizando o *K-folds*, uma técnica utilizada para separar a base de dados em conjuntos de treino e teste. O algoritmo foi executado com k = 10. A base de dados utilizada para os experimentos continha 7848 amostras, então, para cada iteração do *K-fold*, eram designadas cerca de 785 amostras para testes e cerca de 7063 amostras para treino.

primeiramente, avaliamos as técnicas de seleção de atributos e modelos preditivos utilizando a base de dados de 33 características, e então, refizemos os experimentos utilizando as 109 características extraídas pelo modelo de extração proposto neste trabalho. Os resultados desses experimentos para 15, 25 e 30 características selecionadas foram apresentados nas Tabelas de 1 a 4. Como pode ser observado, para os 4 modelos preditivos avaliados, a melhor configuração encontrada foi utilizando um sub-conjunto de atributos selecionado por uma das técnicas escolhidas.

VIII. Resultados

Os resultados dos experimentos utilizando a base de dados de 33 características podem ser encontrados nas tabelas de 1 a 4. foram mostradas as acurácias para cada técnica de seleção de atributos quando utilizadas as 15, 25 e 30 melhores características.

Na Tabela II, são apresentados os valores da acurácia obtida com o modelo de *Random Forest*. Pode-se notar que ao utilizar o modelo de seleção de atributos baseado em estimação de informação mútua, conseguimos, com 25 características, um

TABLE I
RESULTADOS DA CLASSIFICAÇÃO OBTIDO NO SITE DO KAGGLE

Extração de Características	Score
Características do Grellier	1.42560
Novas características	1.42526
Novas características PCA	2.90960
Novas características por *passband*	1.22910

TABLE II
RESULTADOS OBTIDOS COM A RANDOM FOREST

Técnica de seleção	número de features		
	15	25	30
Seleção com PCA	0.591	0.699	0.724
ANOVA	0.688	0.723	0.718
Estimação de Informação Mutua	0.726	**0.732**	0.728
Sem Seleção (33 atributos)	0.726		
Sem Seleção (Vetores do PCA)	0.657		

valor de acurácia maior, de 0.732, que quando utilizadas todas as 33, de 0.726.

Na Tabela III, são apresentados os valores de acurácia para a SVM, e também conseguimos melhores resultados aplicando a seleção de atributos. Desta vez, o método utilizando ANOVA conseguiu 0.585 de acurácia utilizando 25 atributos, enquanto sem a seleção de atributos, a SVM obteve 0.565.

Na Tabela IV, são mostrados os resultados utilizando a MLP. Para este método preditivo, os 30 melhores atributos selecionados pelo método baseado em estimação de informação mútua conseguiu 0.646 de acurácia, valor maior que o obtido ao utilizar todas as 33, 0.637.

Na Tabela V, é possível perceber que a diminuição no número de dimensões do espaço de amostras tem uma grande contribuição com a performance do KNN. Utilizando apenas as 15 características mais relevantes segundo o método de estimação de informação mútua, o algoritmo consegue uma acurácia de 0.634 contra os 0.565 utilizando os 33 atributos.

Para ter uma melhor visualização do impacto da quantidade

TABLE III
RESULTADOS OBTIDOS COM A SUPORT VECTOR MACHINE

Técnica de seleção	número de features		
	15	25	30
Seleção com PCA	0.328	0.412	0.559
ANOVA	0.571	**0.585**	0.576
Estimação de Informação Mutua	0.569	0.572	0.577
Sem Seleção (33 atributos)	0.565		
Sem Seleção (Vetores do PCA)	0.565		

TABLE IV
RESULTADOS OBTIDOS COM A MLP

Técnica de seleção	número de features		
	15	25	30
Seleção com PCA	0.452	0.574	0.633
ANOVA	0.629	0.636	0.644
Estimação de Informação Mutua	0.629	0.644	**0.648**
Sem Seleção (33 atributos)	0.637		
Sem Seleção (Vetores do PCA)	0.630		

TABLE V
RESULTADOS OBTIDOS COM O KNN

Técnica de seleção	número de features		
	15	25	30
Seleção com PCA	0.395	0.501	0.560
ANOVA	0.602	0.577	0.567
Estimação de Informação Mutua	**0.634**	0.612	0.575
Sem Seleção (33 atributos)	0.565		
Sem Seleção (Vetores do PCA)	0.565		

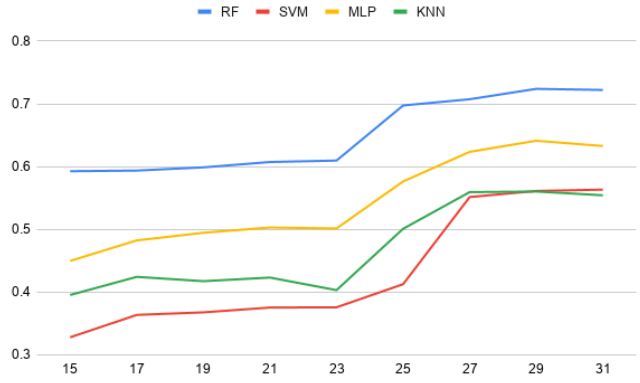

Fig. 1. Acurácia obtida por cada modelo preditivo utilizando diversas quantidades de características selecionadas pela técnica baseada no PCA na base de 33 características

de características no resultado de um modelo preditivo, montamos os gráficos das Figuras 1 a 6. estes gráficos mostram para cada modelo preditivo, a acurácia obtida nos experimentos para as quantidades de atributos utilizadas. No eixo horizontal, estão anotadas as quantidades de atributos, enquanto no eixo vertical são mostradas as acurácias do modelo. Nas Figuras de 1 a 3 são mostrados os valores para a base de dados de 33 características, enquanto nas Figuras de 4 a 6 são mostrados os resultados para a base de 109 características. É possível notar que os resultados são melhores quando utilizamos o método de extração proposto de 109 atributos. A melhor acurácia obtida foi de 0.775, quando utilizadas 89 atributos selecionados pelo método baseado em informação mútua e a *Random Forest* como método preditivo. Para a base de 33 características, o melhor método encontrado conseguiu acurácia de 0.732, utilizando também a *Random Forest*, mas com 25 características extraídas também pelo método de seleção baseado em informação mútua.

Na Figura 1, são mostradas as acurácias obtidas com a base de 33 características e a técnica de seleção baseada nos componentes principais resultantes do PCA.

Na Figura 2, são mostradas as acurácias obtidas com a base de 33 características e a técnica de seleção baseada em informação mútua.

Na Figura 3, são mostradas as acurácias obtidas com a base de 33 características e a técnica de seleção baseada em ANOVA

Na Figura 4, são mostradas as acurácias obtidas com a base de 109 características e a técnica de seleção baseada nos

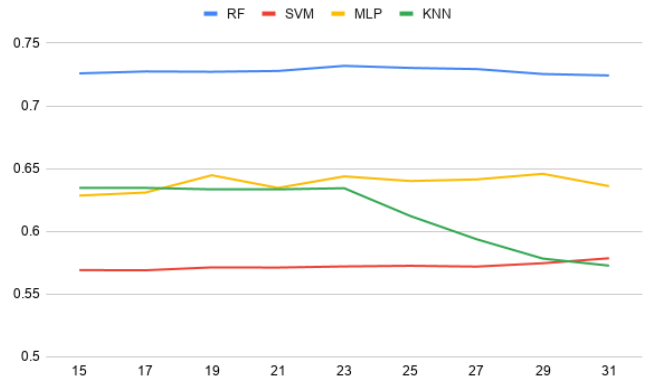

Fig. 2. Acurácia obtida por cada modelo preditivo utilizando diversas quantidades de características selecionadas pela técnica baseada em Estimação de Informação Mútua na base de 33 características

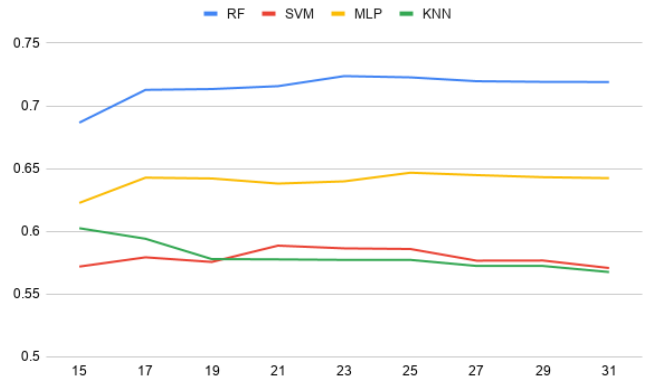

Fig. 3. Acurácia obtida por cada modelo preditivo utilizando diversas quantidades de características selecionadas pelo ANOVA na base de 33 características

componentes principais resultantes do PCA.

Na Figura 5, são mostradas as acurácias obtidas com a base de 109 características e a técnica de seleção baseada em informação mútua.

Na Figura 6, são mostradas as acurácias obtidas com a base de 109 características e a técnica de seleção baseada em ANOVA

References

[1] T. Allam Jr, A. Bahmanyar, R. Biswas, M. Dai, L. Galbany, R. Hlozek, E. E. Ishida, S. W. Jha, D. O. Jones, R. Kessler *et al.*, "The photometric lsst astronomical time-series classification challenge (plasticc): Data set," *arXiv preprint arXiv:1810.00001*, vol. 1, 2018.

[2] LSSTC, "Lsst project mission statement," Site for the LSST created by LSST Corporation, 2019. [Online]. Available: www.lsst.org/about

[3] ——, "Lsst dark energy science collaboration," Web page hosted on GitHub, 2019. [Online]. Available: https://lsstdesc.org/

[4] F. Bianco and R. Street, "Transients and variable stars lsst science collaboration," Web page hosted on GitHub, 2019. [Online]. Available: https://lsst-tvssc.github.io/

[5] O. Grellier, "Plasticc in a kernel meta and data," PLAsTiCC discussion on Kaggle, 2019. [Online]. Available: https://www.kaggle.com/ogrellier/plasticc-in-a-kernel-meta-and-data

[6] M. A. d. Freitas Junior, "Classificador lgbm aplicado na resolução do desafio plasticc do telescópio lsst," 2019.

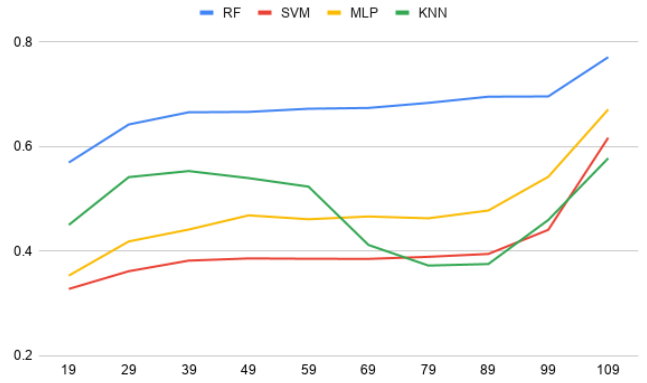

Fig. 4. Acurácia obtida por cada modelo preditivo utilizando diversas quantidades de características selecionadas pela técnica baseada no PCA na base de 109 características

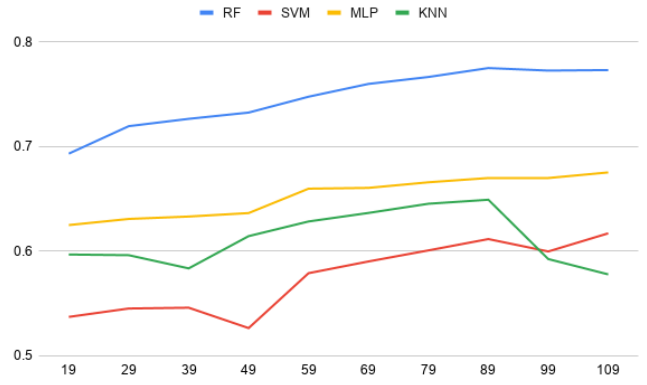

Fig. 5. Acurácia obtida por cada modelo preditivo utilizando diversas quantidades de características selecionadas pela técnica baseada em Estimação de Informação Mútua na base de 109 características

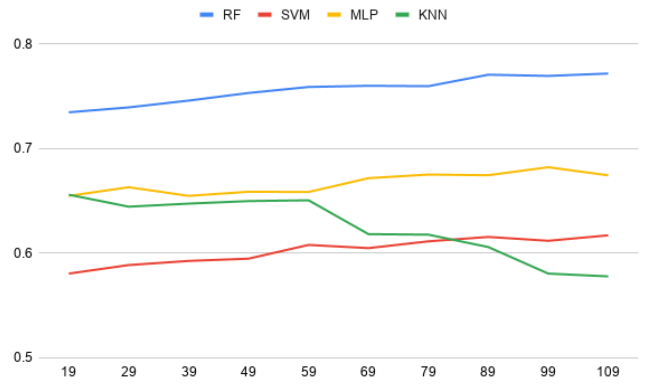

Fig. 6. Acurácia obtida por cada modelo preditivo utilizando diversas quantidades de características selecionadas pelo ANOVA na base de 109 características

[7] G. Kaushik, "Visualization tools for feature importance and principal component analysis," Mar 2019. [Online]. Available: https://medium.com/cascade-bio-blog/creating-visualizations-to-better-understand-your-data-and-models-part-1-a51e7e5af9c0

[8] Microsoft, "Lgbm documentation," Microsoft Open Source Code of Conduct, 2019. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/index.html

[9] J. P. Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques, third edition*, 3rd ed. Morgan Kaufmann Publishers, 2012.

[10] Microsoft, "Lgbm parameters," Microsoft Open Source Code of Conduct, 2019. [Online]. Available: https://lightgbm.readthedocs.io/en/latest/Parameters.html