Universidade Federal de Viçosa, *Campus* Florestal

Disciplina: TÓPICOS ESPECIAIS III

Professor: Fabrício A. Silva

Aluno: João Marcos Alves Modesto Ramos **Matrícula:** 3506

Trabalho Prático II

Engenharia de Atributos e Ajustes de Modelos

Florestal, MG

2020

Sumário

Link do vídeo: O vídeo se encontra neste link:
https://www.youtube.com/watch?v=50eC1LHbDZI

Link dos slides: Os slides se encontram nesse link:

Modelos de Redes Neurais

1ª Etapa - Limpeza

Em relação ao desafio do Kaggle “Titanic - Machine Learning from Disaster”, são passados dois datasets, um de treino e um de testes. Ambos possuem as mesmas colunas, com exceção da coluna “Survived”, onde somente o de treino possui. Abaixo, podemos ver como os dados são organizados no dataset:

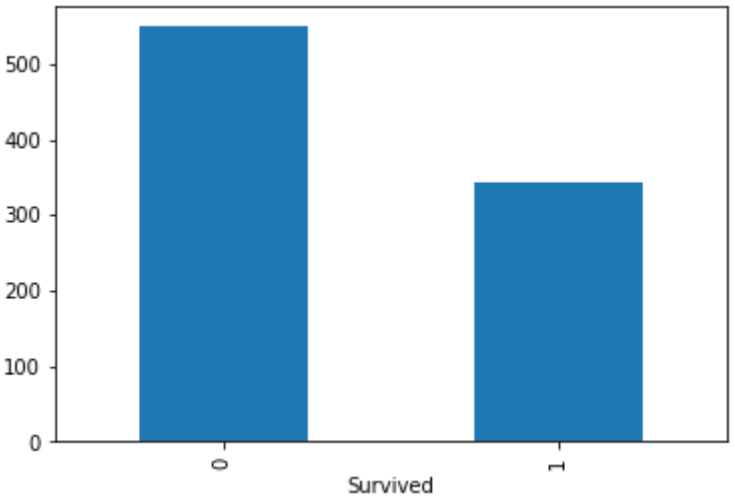
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

Podemos observar que, diversas destas colunas necessitam de limpeza. Para isto, criamos duas funções, para converter a coluna “Sex” e a coluna “Cabin” para se a pessoa possui uma cabine ou não. Antes de aplicarmos estas, é necessário também, converter a coluna “Age” e “Cabin” para que os valores “NaN” sejam transformados em valores mais compreensíveis.

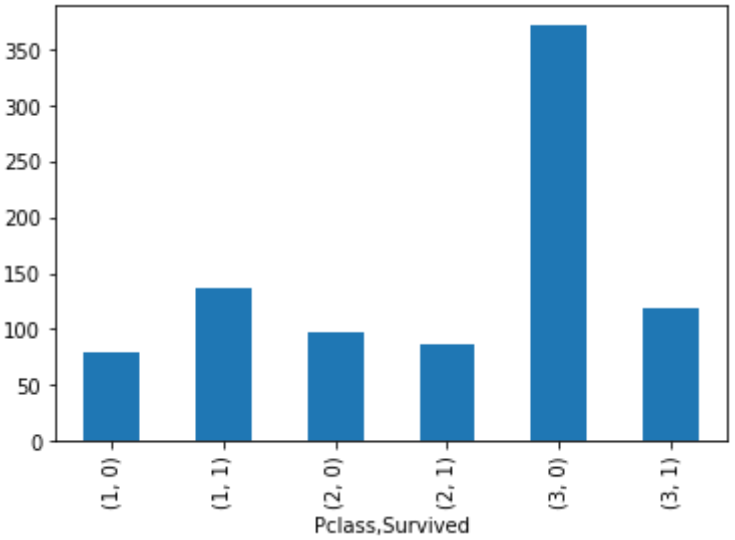
Foi também observado, que a coluna “Ticket”, “Name”, e “PassangerID” são dados de identificação do passageiro, portanto, não possui uma relevância muito alta para a aplicação do algoritmo.

2ª Etapa - Análise de colunas

Nesta etapa, iremos analisar alguns gráficos relevantes para o dataset, para tentarmos identificar quais colunas são mais relevantes para a análise. Um fator importante também é analisarmos a porcentagem de sobreviventes, que neste caso, são 38% dos passageiros, como podemos ver no gráfico abaixo.

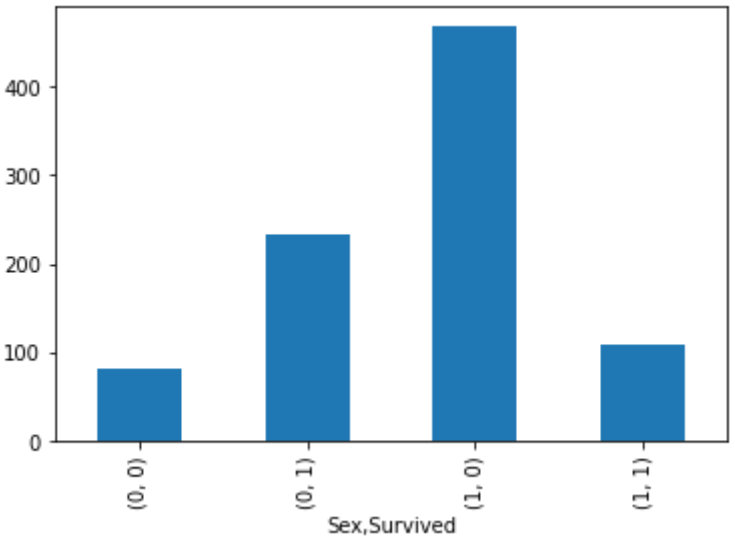


Pclass



Nesta coluna, ao plotarmos o gráfico, reparamos que grande parte dos passageiros da 3 classe não sobreviveram, e um passageiro da primeira classe tem mais chances de sobreviver do que não. Portanto, tal coluna pode ser relevante para a análise dos dados

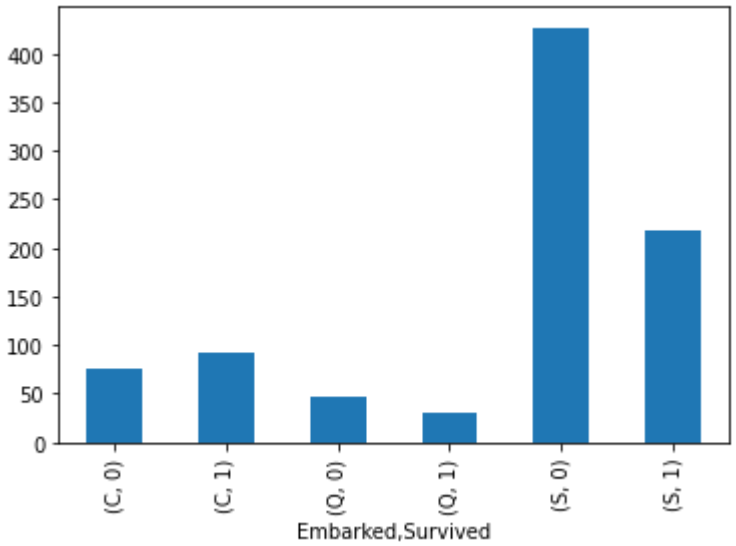
Sex



Sendo que 0 corresponde ao sexo feminino, vemos que grande parte dos sobreviventes eram do sexo feminino, e se um passageiro era do sexo masculino, ele possuia uma chance maior de não sobreviver.

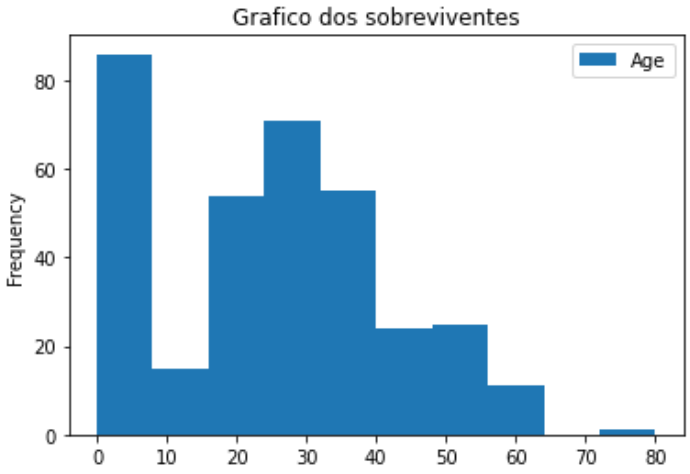
Embarked

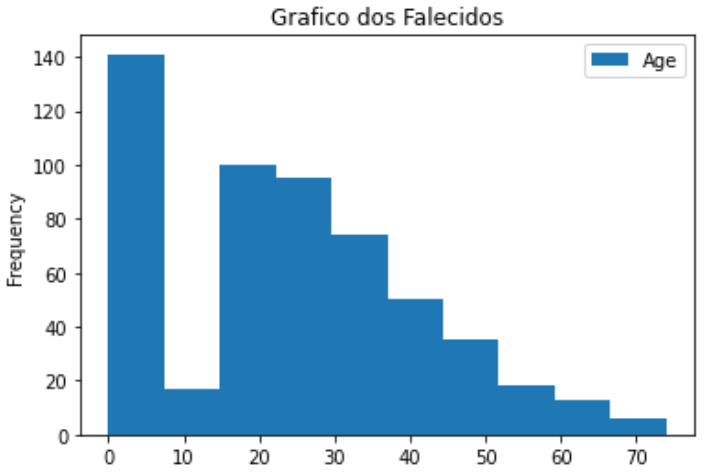
Embarked	Survived	
C	0	75
	1	93
Q	0	47
	1	30
S	0	427
	1	217



Em relação ao qual porto o passageiro embarcou, podemos ver que grande parte dos tripulantes do porto “S” de “Southampton” eram a maioria, e proporcionalmente, poucos sobreviveram, enquanto os que embarcaram no porto “C” de “Cherbourg”, a maior parte sobreviveram.

Age





Em relação a idade, é importante notarmos que grande parte dos sobreviventes tinham menos de 10 anos. Outro detalhe importante é como a curva do gráfico se comporta, percebemos padrões semelhantes tanto entre sobreviventes quanto dos falecidos. Porém, ao final da curva, a cauda dos sobreviventes cai mais drasticamente.

Sibsp

SibSp	Survived	
0	0	398
	1	210
1	0	97
	1	112
2	0	15
	1	13
3	0	12
	1	4
4	0	15
	1	3
5	0	5
8	0	7

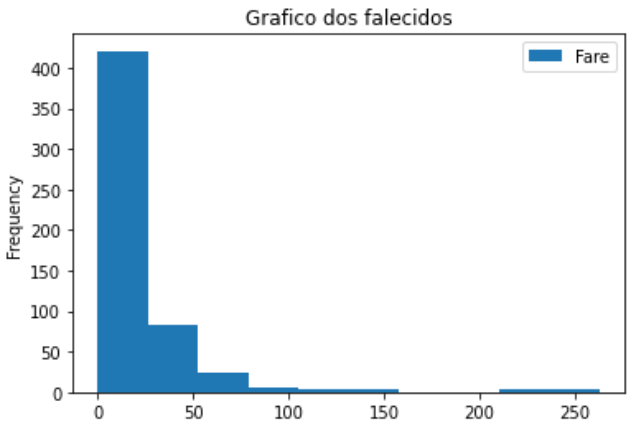
Consiste no número de irmãos/cônjuges a bordo do Titanic. Podemos perceber que pessoas com 1 cônjuge/irmão tem mais probabilidades de sobreviver do que falecer, e podemos ver que, acima de um certo número, não há sobreviventes. Portanto, é um atributo relevante.

Parch

Parch	Survived	
0	0	445
	1	233
1	0	53
	1	65
2	0	40
	1	40
3	0	2
	1	3
4	0	4
5	0	4
	1	1
6	0	1

Consiste no número de dos pais/filhos a bordo do Titanic. Podemos perceber, assim como a coluna “Sibsp” que em a taxa de sobrevivência varia de acordo com a quantidade de pais/filhos, portanto, é um atributo relevante.

Fare



Consiste no valor pago da passagem. Vemos que grande parte dos sobreviventes, pagaram entre 0 a 100 U.M (Unidades monetárias). A frequência de falecidos que pagaram valores mais altos é bem mais baixa que a de sobreviventes, mostrando a relevância dessa coluna

Cabin

Cabin	Survived	
0	0	481
	1	206
1	0	68
	1	136

Consiste, depois da limpeza, se o usuário possui ou não uma cabine. Vemos que, se um usuário possui uma cabine, a chance dele sobreviver é bem maior, portanto é um atributo relevante para a nossa análise.

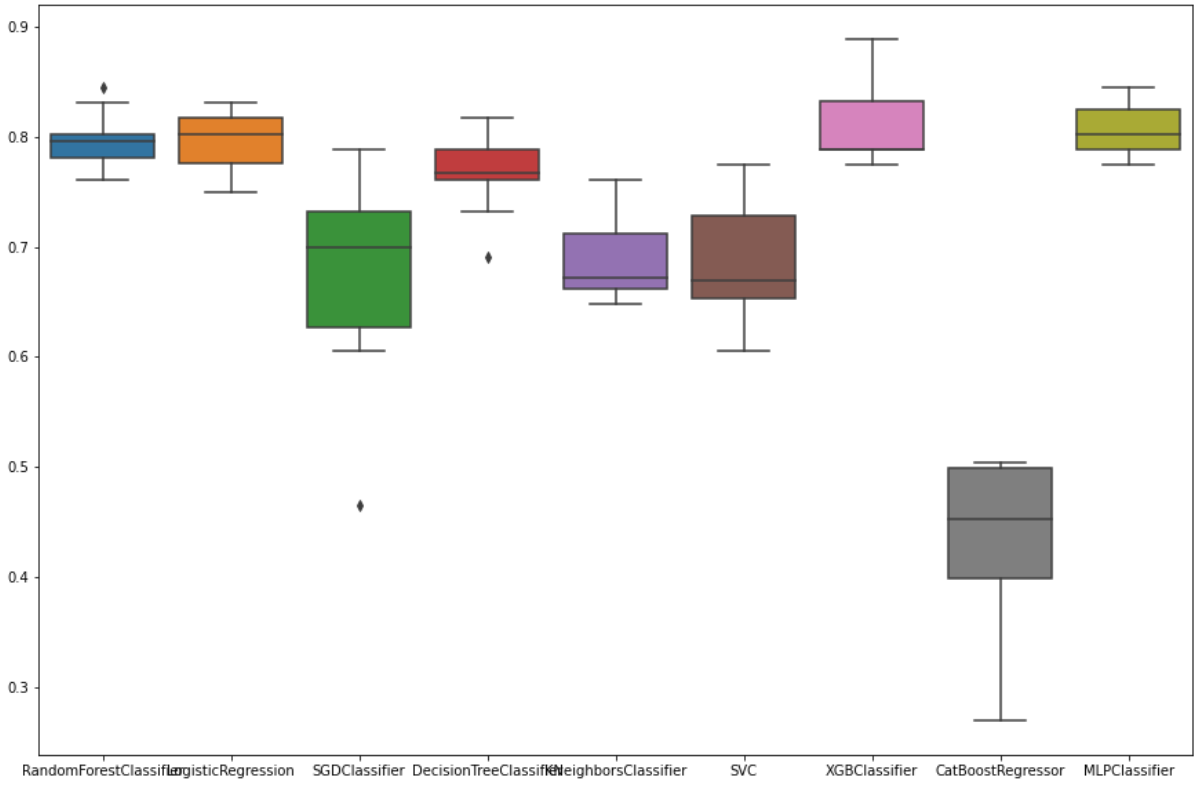
3ª Etapa - Engenharia de atributos

Nesta etapa, como já efetuamos a conversão da coluna “Cabin” e “Sex”, nos resta converter as linhas da coluna “Pclass” e “Embarked” em colunas. Depois da conversão, o dataframe encontra-se da seguinte forma:

	Survived	Sex	Age	SibSp	Parch	Fare	Cabin	Pclass_1	Pclass_2	Pclass_3	Embarked_C	Embarked_Q	Embarked_S
0	0	1	22.0	1	0	7.2500	0	0	0	1	0	0	1
1	1	0	38.0	1	0	71.2833	1	1	0	0	1	0	0
2	1	0	26.0	0	0	7.9250	0	0	0	1	0	0	1
3	1	0	35.0	1	0	53.1000	1	1	0	0	0	0	1
4	0	1	35.0	0	0	8.0500	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	1	27.0	0	0	13.0000	0	0	1	0	0	0	1
887	1	0	19.0	0	0	30.0000	1	1	0	0	0	0	1
888	0	0	0.0	1	2	23.4500	0	0	0	1	0	0	1
889	1	1	26.0	0	0	30.0000	1	1	0	0	1	0	0
890	0	1	32.0	0	0	7.7500	0	0	0	1	0	1	0

4ª Etapa - Escolha do modelo

Nesta etapa, escolhemos qual seria o melhor modelo para execução. Antes, precisamos separar os dados em Treino, Validação e Teste. Depois deste processo, temos que escolher o melhor modelo para o contexto. com isso, vamos verificar de acordo com o score do Cross Validation. Executamos para cada algoritmo com cv = 10, e com os dados de treino, e obtemos o seguinte resultado:



Vemos que o XGB apresenta um resultado excelente, junto com o MLP e o Random Forest e o Logistic Regression. Ou seja, qualquer um destes algoritmos é uma excelente escolha para a resolução do problema.

Este teste, foi feito várias vezes, e o que se mantém mais estável na colocação é o RandomForest. Apesar de apresentar excelentes resultados, o XGB em algumas ocorrências apresenta um desempenho bem inferior, por isto, iremos usar o RandomForest para ser o nosso modelo de exemplo. Além dele, vamos comparar também usando o MLP, já que foi o modelo atribuído neste trabalho.

O vídeo se encontra neste link: https://www.youtube.com/watch?v=50eC1LHbDZI

Métricas	Random Forest	MLP
Acurácia:	0.815642458100558	0.8100558659217877
Cross Val Score:	0.80 (+/- 0.05)	0.81 (+/- 0.06)

5ª Etapa - Hyperparameter tuning

Para tentar melhorar o resultado dos algoritmos, foi efetuado o Tunning dos parâmetros utilizando o Random Search. Abaixo, podemos ver os resultados de cada algoritmo:

Métricas	Random Forest	MLP
Acurácia:	0.8268156424581006	0.8044692737430168
Cross Val Score:	0.82 (+/- 0.04)	0.78 (+/- 0.07)

Além disto, os melhores parâmetros para cada algoritmo, foram:

- Random Search: {'n_estimators': 800, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 110, 'bootstrap': False}
- MLP : {'solver': 'adam', 'learning_rate': 'adaptive', 'hidden_layer_sizes': (50, 50, 50), 'alpha': 0.5, 'activation': 'relu'}

6ª Etapa - Aplicação de Auto ML

Foi também, utilizado um algoritmo de Auto Machine Learning, o AutoSKlearner. Com ele, foi passado os dados já limpos, e efetuados a engenharia de atributos. Foi obtido os seguintes resultados:

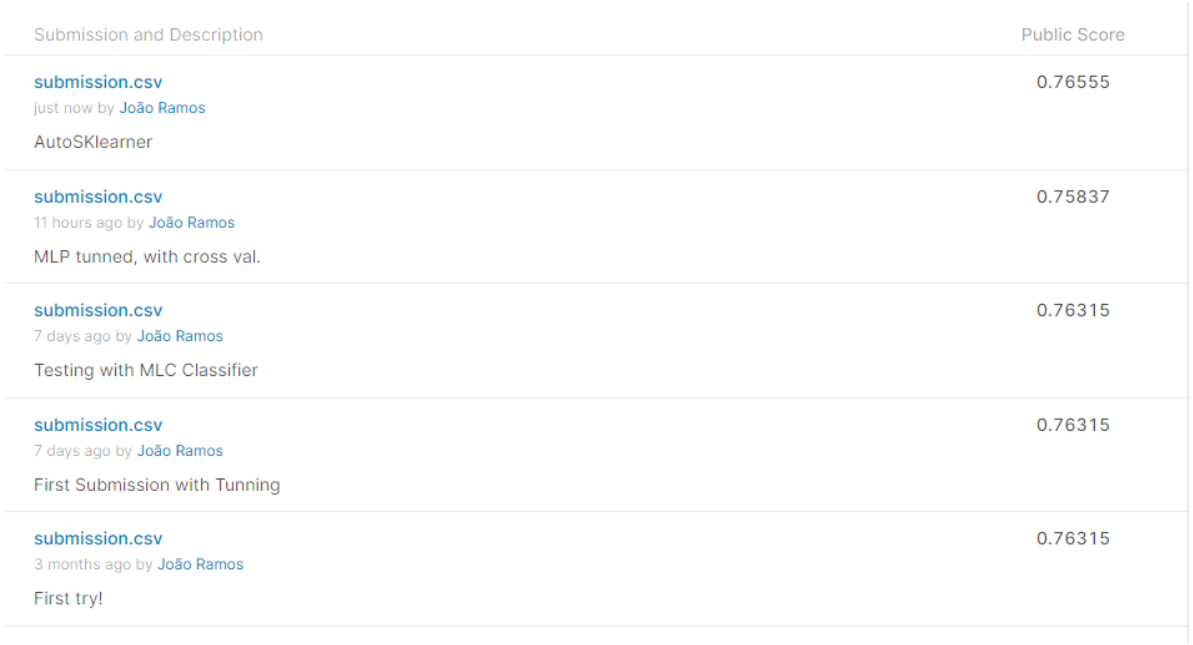
Métricas	AutoSKlearner
Acurácia:	0.8156424581005587
Cross Val Score:	0.80 (+/- 0.04)

Abaixo, podemos ver uma tabela comparando todos os algoritmos, nos seus melhores casos.

Métricas	Random Forest	MLP	AutoSKlearner
Acurácia:	0.826815642458100	0.810055865921787	0.815642458100558
Cross Val Score:	0.82 (+/- 0.04)	0.81 (+/- 0.06)	0.80 (+/- 0.04)

Resultados

Foram enviados ao Kaggle várias tentativas com cada algoritmo, e assim, podemos ver a pontuação de cada predição, como mostra a imagem abaixo:



Submission and Description	Public Score
submission.csv just now by João Ramos AutoSKlearner	0.76555
submission.csv 11 hours ago by João Ramos MLP tunned, with cross val.	0.75837
submission.csv 7 days ago by João Ramos Testing with MLC Classifier	0.76315
submission.csv 7 days ago by João Ramos First Submission with Tunning	0.76315
submission.csv 3 months ago by João Ramos First try!	0.76315

Em relação às melhoras, o primeiro trabalho foi efetuado com engenharia de atributos, portanto, a grande melhora poderia ser dada pela escolha de outros modelos ou pelo ajuste dos hiperparâmetros. Porém, como foram visto os resultados acima, não apresentaram melhoras significativas, e em alguns casos, apresentaram até uma determinada piora. Isso pode ser justificado pela aleatoriedade presente dentro dos algoritmos.

Outro resultado interessante é a utilização de métodos de Auto-ML, que apresentaram resultados semelhantes à análise feita, e demandou um esforço menor nas etapas de preparação dos modelos. Ou seja, em determinados casos, a utilização de um Auto-ML é uma escolha válida.