PREDIÇÃO DE DIABETES ATRAVÉS DE APRENDIZADO DE MÁQUINA

BERTO, Matheus Vargas Volpon [1]

RESUMO

Este documento apresenta a descrição e análise de um projeto referente às áreas Data Science e Machine Learning. O projeto contempla a visualização de dados sobre a diabetes tipo 2 e treinamento de modelos simples de aprendizado para prevê-la. É importante destacar também que este projeto tem caráter experimental e apenas propósitos de estudo.

Palavras-chave: Data Science; Machine Learning; Visualização; Modelos.

[1]Graduando do Curso de Ciência da Computação da UFSCar - Sorocaba, matheusvvb@hotmail.com;

1. CONCEITO

O projeto desenvolvido tem como base um conjunto de dados referentes à diabetes tipo 2 e disponível pubicamente na internet. As informações presentes no mesmo foram coletadas através de questionários feitos à pacientes do “Sylhet Diabetes Hospital”, em Bangladesh.

Os objetivos do projeto em questão são extrair e visualizar informações do conjunto de dados por meio de imagens, selecionar e manipular características importantes presentes no conjunto e, por fim, aplicação das mesmas em algoritmos simples de aprendizado, avaliando seus resultados.

2. RECURSOS UTILIZADOS

Para realização deste projeto foram utilizados: o conjunto de dados já citado - Early stage diabetes risk prediction dataset [6] -, além do ambiente de execução Google Colaboratory configurado para linguagem Python e algumas bibliotecas padrão para análise de dados e aprendizado de máquina, como Numpy, Pandas, Matplotlib, Seaborn e Scikit Learn (vejas-as nas referências).

3. ANÁLISE EXPLORATÓRIA

Inicialmente, os dados foram importados para o script em Python, por meio da função *pd.read_csv()* e suas primeiras instâncias (linhas) apresentadas abaixo:

	Age	Gender	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class
0	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
1	58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
2	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
3	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
4	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive

Figura 1 – Conjunto de dados original.
Fonte: Google Colaboratory.

Como se pode observar, o quadro de dados (*data frame)* apresenta dezessete colunas, sendo dezesseis delas de variáveis independentes (*features*) e, a última, o

resultado a ser previsto – variável dependente ou *target*. Dentre todas as dezesseis primeiras colunas, apenas a primeira, referente à idade do paciente, apresenta dados numéricos e discretos, enquanto todo o restante das informações são definidas categoricamente.

Continuando, é possível sabermos também o tamanho desse conjunto de dados, em outras palavras, a quantidade de linhas que o compõe, além de uma breve descrição estatística de suas colunas – no caso, apenas da coluna *Age*, visto que que é a única variável numérica. Tal processo é executado através do atributo *shape* e do método *describe()*.

```
(520, 17)
                Age
count   520.000000
mean     48.028846
std      12.151466
min      16.000000
25%      39.000000
50%      47.500000
75%      57.000000
max      90.000000
```

Figura 2 – Tamanho do conjunto e descrição estatística da coluna *Age*.
Fonte: Google Colaboratory.

Por fim, foram utilizados também gráficos e imagens, listados e descritos a seguir.

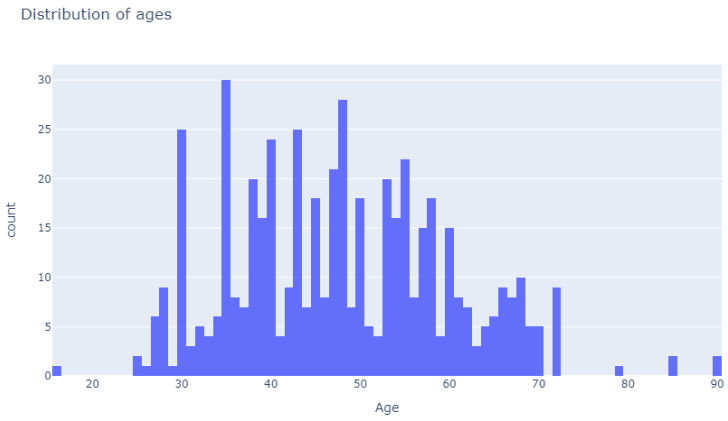

Figura 3 – Histograma de distribuição de idades.
Fonte: Google Colaboratory.

O histograma acima nos mostra a frequência das idades dos pacientes dentro do conjunto de dados, revelando a maior concentração das mesmas entre os 30 e 55 anos, com picos de frequência, aproximadamente, nos 30, 35 e 48 anos.

Já neste segundo histograma, foi relacionada a distribuição das idades em relação ao gênero dos entrevistados. O resultado está exposto abaixo:

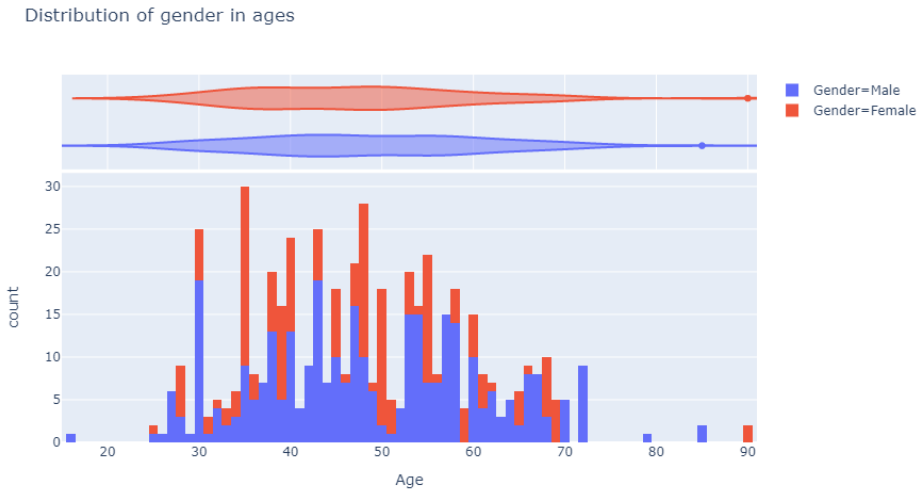

Figura 4 – Histograma de distribuição de idades em relação ao gênero.
Fonte: Google Colaboratory.

Como se pode observar, os homens são predominantes em grande parte das ocorrências. Além disso, a pessoa mais velha entrevistada – 90 anos – é uma mulher, enquanto o mais jovens – com 16 anos – é um homem. Acima do histograma, foi construído anexado também um subgráfico do tipo *violin*, mostrando a concentração de cada gênero.

Semelhantemente à imagem acima, foi criado mais um último histograma, o qual, dessa vez, relaciona a frequência das idades dos entrevistados com o resultado dos mesmos para diabetes (positivo ou negativo).

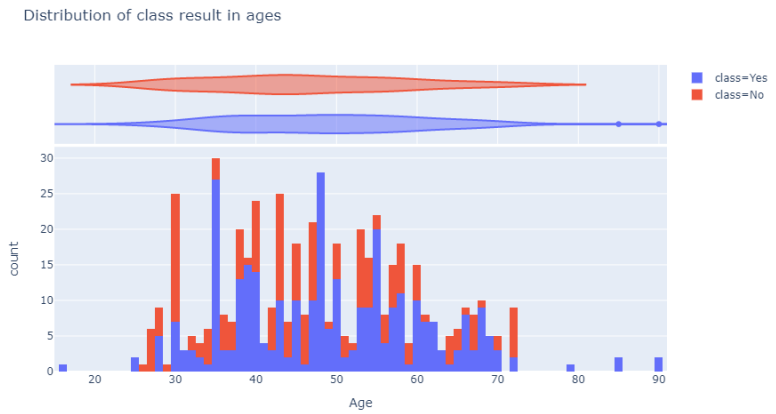

Figura 5 – Histograma de distribuição de idades em relação à diabetes.
Fonte: Google Colaboratory.

Ademais, foram plotados também gráficos de barras que relacionam a frequência de determinadas colunas – no caso, *Obesity*, *Polyuria*, *Polydipsia* e *Alopecia* – em relação à diabetes.

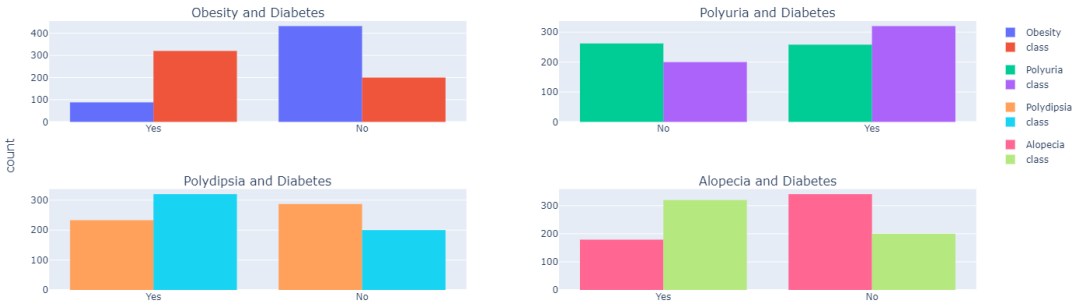

Figura 6 – Gráfico de barras de certas colunas em relação à diabetes
Fonte: Google Colaboratory.

A análise dos gráficos acima já demonstra um indício de níveis variados de correlação entre as colunas citadas anteriormente e o resultado (*class*). Por exemplo, no gráfico mais superior à esquerda, perceba que a quantidade de pessoas obesas é cerca de três vezes menor que a quantidade de pessoas diabéticas. Já na relação contrária – pessoas não obesas e não diabéticas – o número de casos de não obesidade é mais que o dobro de adiabéticas. Portanto, a variável *Obesity* não tem uma correlação positiva com o resultado final a ser previsto. O mesmo vale, mesmo que em menores porporções, para a coluna *Alopecia.*

Por outro lado, variáveis como *Polyuria* e *Polydipsia* apresentam uma correlação positiva à diabetes, havendo pouca diferença entre suas frequências. Isso será mais detalhado posteriormente.

4. PRÉ-PROCESSAMENTO E SELEÇÃO DE DADOS

Nesta etapa do projeto, o conjunto de dados será manipulado, de forma com que seja possível sua utilização pelos modelos de aprendizado, que em muitos casos não lidam corretamente com variáveis categóricas. Posteriormente, mediante certos critérios e adequações, apenas algumas das colunas originalmente existentes serão aplicadas ao algoritmos em Python.

Em primeiro lugar, foi feita uma codificação binária em cada variável categórica - exceto o gênero - do *data frame*, substituindo-se “Yes” por 1 e “No” por 0. O mesmo foi feito para a coluna alvo. Essa forma de codificação foi aplicada, nesse caso, pois as características da tabela se referem à apresentação ou não de sintomas, passíveis a essa aplicação.

Já na coluna referente ao gênero, a codificação binária não é recomendada. Embora essa variável também receba dois possíveis valores (“*Male*” e “*Female*”), não há uma relação numérica ou linear antre ambos. Portanto, a codificação utilizada para essa variável foi a OneHotEncoding.

A *OneHotEncoding* é uma técnica voltada para variáveis categóricas ou nominais, a qual transforma os diferentes valores presentes em vetores binários únicos, de tamanho proporcional aos tipos de valores presentes. Dessa forma, não há riscos de que o algoritmo de aprendizado encontre uma relação de ordem ou grandeza entre tais valores.

Logo, após aplicação dos dois tipos de codificação acima, tem-se a seguinte tabela:

	Age	Polyuria	Polydipsia	sudden weight loss	weakness	Polyphagia	Genital thrush	visual blurring	Itching	Irritability	delayed healing	partial paresis	muscle stiffness	Alopecia	Obesity	class	Female	Male
515	39	1	1	1	0	1	0	0	1	0	1	1	0	0	0	1	1	0
516	48	1	1	1	1	1	0	0	1	1	1	1	0	0	0	1	1	0
517	58	1	1	1	1	1	0	1	0	0	0	1	1	0	1	1	1	0
518	32	0	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	0
519	42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figura 7 – Conjunto de dados após codificações iniciais.
Fonte: Google Colaboratory.

Neste momento, todas as colunas do conjunto de dados foram transformadas em valores numéricos e podemos, então, vizualizar mais precisamente a correlação entre os dados. Para isso, a matriz de correlação abaixo foi gerada:

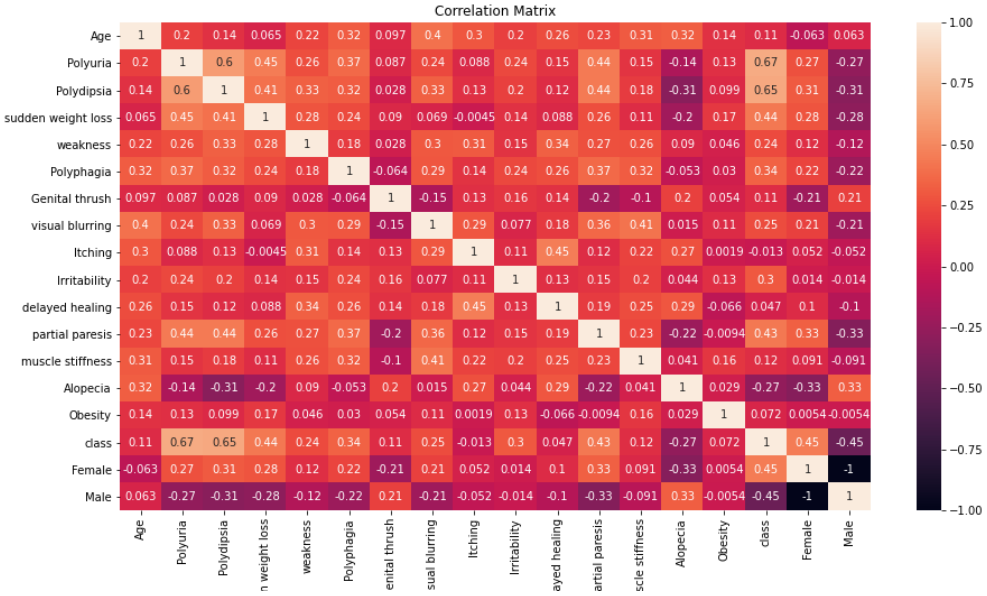

Figura 8 – Matriz de correlação do conjunto de dados após codificações iniciais.
Fonte: Google Colaboratory.

A matriz de correlação é uma matriz quadrada (*nxn*) e simétrica onde cada elemento representa a correlação entre as variáveis daquela coordenada. Elementos próximos de 1 e, na imagem, de tom mais claro, apresentam alta correlação direta, enquanto valores próximos de 0 quase não apresentam correlação e valores negativos, relação inversa. Assim, a partir da imagem acima podemos iniciar a escolha das variáveis que farão parte do modelo de aprendizagem.

Entretanto, apenas a matriz de correlação pode não ser o suficiente determinar essa escolha, é preciso também analisar a importância ou o peso de cada variável na predição do resultado final. Para isso, a classe ExtraTreesClassifier foi utilizada.

Tal classe implementa um metaestimador que ajusta uma série de árvores de decisão aleatórias (também conhecidas como árvores extras) em várias subamostras do conjunto de dados e usa a média para melhorar a precisão preditiva e o sobreajuste de controle. É comumente utilizado como modelo base para análise de importância de cada característica (*feature*). Assim, utilizaremos seu resultado como mais um meio de escolha para quais colunas efetivamente serão utilizadas nos modelos a serem treinados.

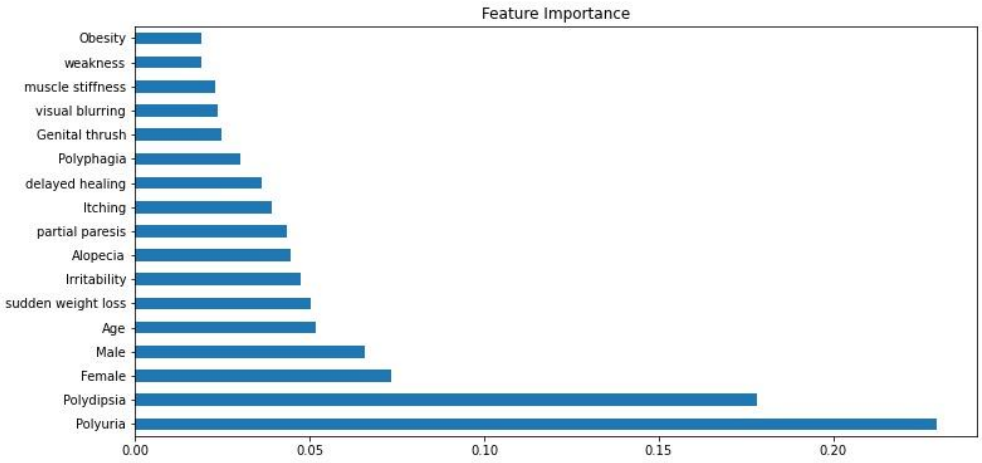

Figura 9 – Gráfico de importância de características.
Fonte: Google Colaboratory.

Ademais, como últimos critérios de escolha para as colunas, foi verificado também a quantidade de dados faltantes (NaN) ou de valores não nulos presentes em cada coluna, o resultado está exposto abaixo:

```
Age                 0
Polyuria            0
Polydipsia          0
sudden weight loss  0
weakness            0
Polyphagia          0
Genital thrush      0
visual blurring     0
Itching             0
Irritability        0
delayed healing     0
partial paresis     0
muscle stiffness    0
Alopecia            0
Obesity             0
class               0
Female              0
Male                0
dtype: int64
```

```
Age                 520
Polyuria            258
Polydipsia          233
sudden weight loss  217
weakness            305
Polyphagia          237
Genital thrush      116
visual blurring     233
Itching             253
Irritability        126
delayed healing     239
partial paresis     224
muscle stiffness    195
Alopecia            179
Obesity              88
class               320
Female              192
Male                328
dtype: int64
```

Figura 10 – Contagem de NaN e valores não nulos do conjunto de dados após codificações iniciais.
Fonte: Google Colaboratory.

Dessa forma, as variáveis escolhidas para serem aplicadas no modelos foram: *Age*, *Polyuria*, *Polydipsia*, *Female*, *Male* e, é claro, *class*.

Por fim, como processamento final no conjunto de dados, todas colunas escolhidas foram submetidas à regularização pelo operador MinMaxScaler. Observe que, na prática, essa regularização alterou somente a coluna de idades, visto que as demais já estavam representadas por 0's e 1's (novo intervalo da coluna *Age*).

Portanto, a tabela de dados final é a seguinte:

	Age	Polyuria	Polydipsia	Female	Male	class
0	0.324324	0.0	1.0	0.0	1.0	1.0
1	0.567568	0.0	0.0	0.0	1.0	1.0
2	0.337838	1.0	0.0	0.0	1.0	1.0
3	0.391892	0.0	0.0	0.0	1.0	1.0
4	0.594595	1.0	1.0	0.0	1.0	1.0

Figura 11 – Conjunto de dados final.
Fonte: Google Colaboratory.

5. MODELOS DE APRENDIZAGEM

Nesta última etapa, foram treinados e avaliados os seguintes algoritmos de aprendizado: Support Vector Machine (SVC), KNN e Decision Tree. Em todos os modelos as etapas realizadas foram idênticas, incluindo a comparação dos resultados a um classificador fictício - *Dummy Classifier*.

Todas as métricas dos classificadores foram impressas no arquivo *.ipynb* e apenas citadas neste relatório.

5.1. SUPPORT VECTOR MACHINE

Neste caso, o modelo utilizado apresenta parâmetros padrão (*C* = 1 e *kernel* = rbf). Um classificador fictício foi criado para classificar as entradas a partir da estratégia "*startified*", a qual se baseia na distribuição dos rótulos presentes nos dados de treinamentos. A partir das métricas – como acurácia, precisão e revocação (recall) – do classificador fictício, é possível saber os resultados mínimos esperados para o modelo SVC (0.49, 0.61 e 0.66, respectivamente).

A acurácia do modelo SVC foi impressa na tela, é possível perceber que tal métrica obteve maior valor no conjunto de teste, comparada ao conjunto de treinamento. Isso nos mostra que o modelo consegue generalizar devidamente os dados, sem indícios de sobreajuste.

Entretanto, somente a acurácia não é o suficiente para validar o modelo, visto que outras taxas, como falsos positivos (FP) e falsos negativos (FN) também devem ser levadas em consideração. Portanto, foi plotada a matriz de confusão do modelo, além de gráficos que mostram a variação das métricas de precisão e revocação de acordo com o tamanho do conjunto de dados.

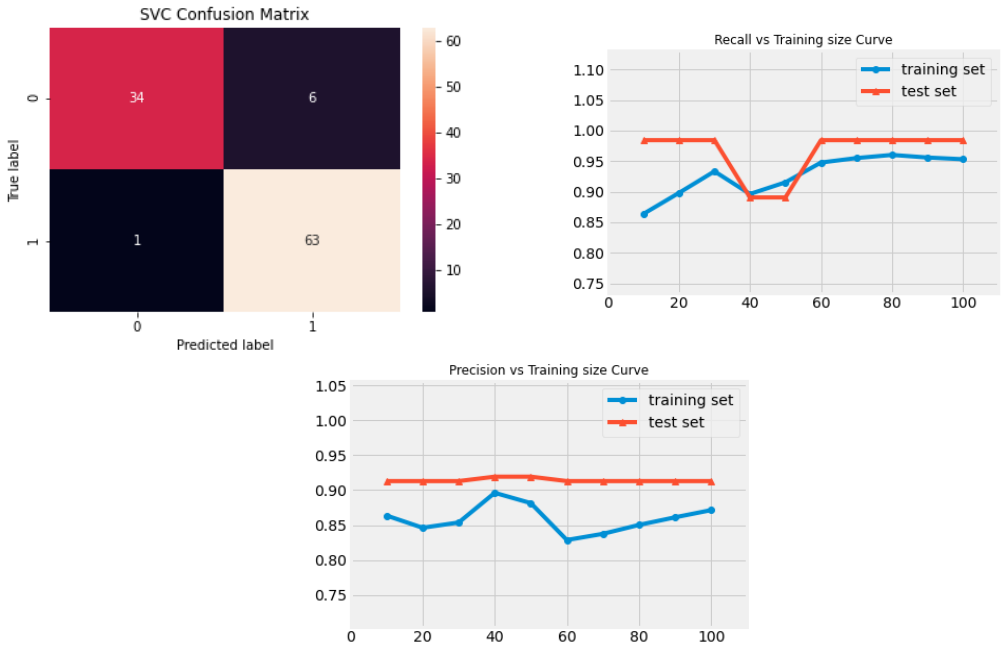

Figura 12 – Matriz de confusão e curvas de métricas modelo SVC.
Fonte: Google Colaboratory.

Analisando as imagens, é possível perceber mais detalhadamente quais são as quantidades de acertos e erros, assim como seus tipos, atingidas pelo modelo, e a variação das métricas no conjunto de treinamento e de teste.

Como última forma de avaliação, foi realizada também a validação cruzada e novo cálculo de precisão e revocação, juntamente com o desvio padrão, obtendo, respectivamente, os valores: 87.62 e 95.33.

Finalmente, conclui-se que o modelo SVC funciona corretamente e de maneira positiva, apresentando métricas superiores aos classificador fictício, além de generalização dos dados - resultados melhores no conjunto de testes em grande parte do tempo – e resultados dentro do intervalo da validação cruzada.

5.2. KNN

Inicialmente, foi testado e verificado qual o número de vizinhos (K) que proporcionasse a melhor acurácia, obtendo-se o gráfico abaixo:

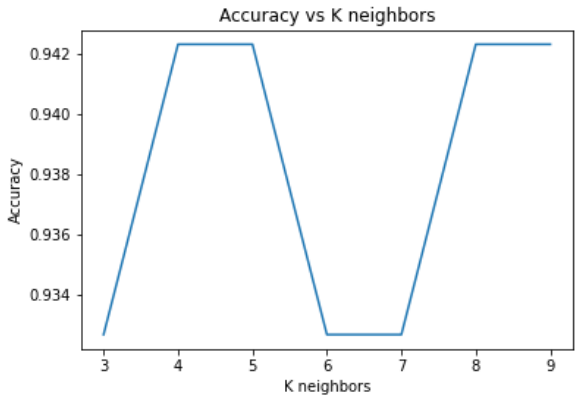

Figura 13 – Gráfico de acurácia do modelo KNN em relação a K.
Fonte: Google Colaboratory.

Neste caso, o classificador fictício criado usa a regra "*most_frequent*" e sua precisão foi 1.0 (100%), o que é compreesível, visto que, como todos os resultados foram previstos como 1, a quantidade de positivos prevista pelo modelo é exatamente a quantidade de verdadeiros positivos. Enquanto isso, sua acurácia e revocação foram ambas de 0.62.

Novamente, foram impressos as mesmas métricas e imagens utilizadas na validação do modelo anterior.

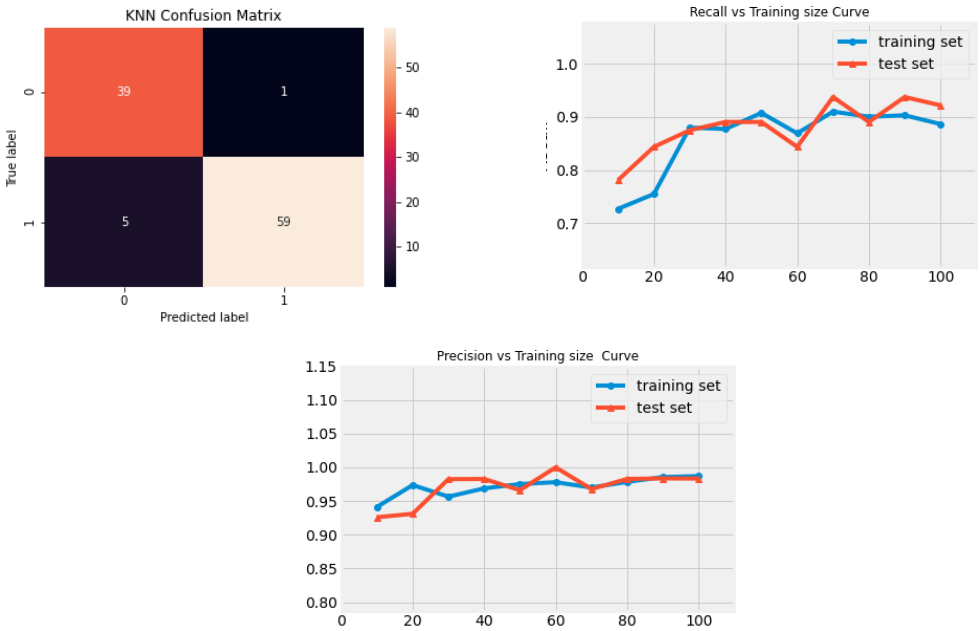

Figura 14 – Matriz de confusão e curvas de métricas modelo KNN.
Fonte: Google Colaboratory.

Aqui, é possível perceber que o modelo de KNN apresenta maior taxa de falsos negativos e menor de falsos positivos, mas, por outro lado, também apresenta valores menores de verdadeiros positivos e negativos. Ademais, este modelo apresenta também variação de valores de precisão e revocação não tão distintos nos conjuntos de treino e teste.

Por fim, a validação cruzada também foi feita e os resultados estão dentro do intervalo calculado.

5.3. DECISION TREE

E, por último, um modelo baseado no algortimo de árvores de decisão foi criado. Semelhantemente ao KNN, foi plotado um gráfico mostrando a variação da acurácia de acordo com a quantidade máxima de nós folhas (*max_leaf_nodes*).

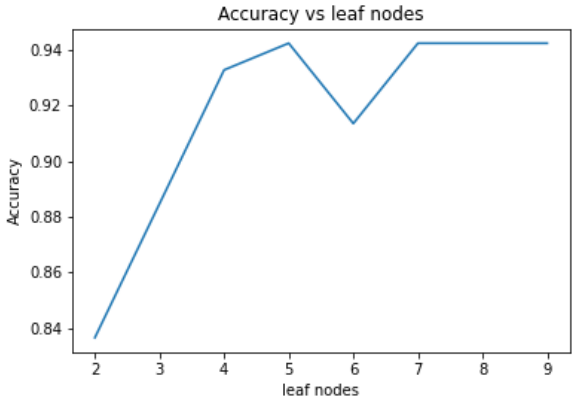

Figura 15 – Gráfico de acurácia do modelo Decision Tree em relação aos nós folha.
Fonte: Google Colaboratory.

Então, o modelo foi criado com o número máximo de nós folhas igual a cinco, além de altura máxima igual a três e critério de ganho de ingormação como Gini, os hiperparâmetros foram escolhidos de forma a evitar o sobreajuste (pré-poda). Abaixo, foi plotada também a própria árvore de decisão:

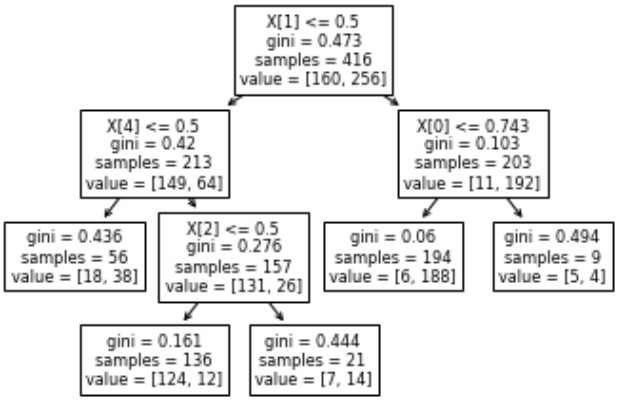

Figura 16 – Construção da Árvore de Decisão treinada.
Fonte: Google Colaboratory.

Novamente, um classificador fictício foi criado, dessa vez com a regra “*uniform*”, que classifica os resultados aleatóriamente e uniformemente, suas métricas foram impressas (0.5, 0.69 e 0.73, aproximadamente).

A impressão da acurácia do modelo de árvore de decisão também mostrou boa generalização no conjunto de testes, 2% maior que no treinamento. As imagens a seguir apresentam mais detalhes:

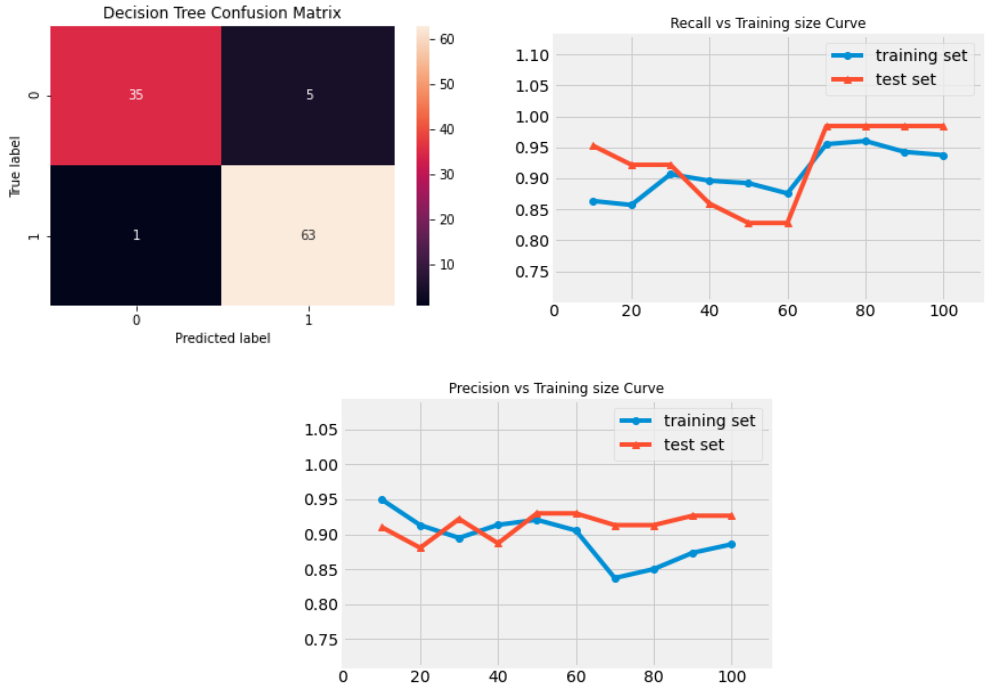

Figura 17 – Matriz de confusão e curvas de métricas modelo Decision Tree.
Fonte: Google Colaboratory.

A validação cruzada mostrou bons resultados e as métricas do modelo se encaixam neste intervalo.

6. CONSIDERAÇÕES FINAIS

Todos os três modelos de aprendizado apresentaram bons resultados, pois apresentaram generalização dos dados, métricas maiores que as dos classificadores fictícios e resultados dentro do intervalo de validação cruzada.

Particularmente, os modelos de Árvore de Decisão e SVC obtiveram resultados semelhantes, nível de generalização maior que o KNN e matrizes de confusão com

poucas diferenças. Mas, se obervarmos mais atentamente, é possível perceber que o modelo de Árvore de Decisão, embora apresente curvas de maior variação, consegue uma matriz de confusão ligeiramente melhor, com mais um verdadeiro negativo e menos um falso positivo, além de menor desvio padrão na validação cruzada.

REFERÊNCIAS

[1] NUMPY. **Numpy Reference Guide.** Release 1.19.0, NumPy Community, 2020. Disponível em <https://numpy.org/doc/stable/numpy-ref.pdf>. Acesso em: 06 de novembro de 2020.

[2] MLXTEND. **Mlxtend: User Guide Index**. Disponível em <http://rasbt.github.io/mlxtend/USER_GUIDE_INDEX/>. Acesso em: 06 de novembro de 2020.

[3] PLOTLY. **Plotly Python Open Source Graphing Library**. Disponível em <https://plotly.com/python/>. Acesso em: 06 de novembro de 2020.

[4] SCICKIT LEARN. **Scikit Learn: Machine Learning in Python**. Disponível em <https://scikit-learn.org/stable/>. Acesso em: 06 de novembro de 2020.

[5] SEABORN. **Seaborn: Statistical Data Visualization.** Disponível em: <https://seaborn.pydata.org/>. Acesso em: 06 de novembro de 2020.

[6] UCI REPOSITORY. **Early stage diabetes risk prediction dataset.** Disponível em <https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset.>. Acesso em: 06 de novembro de 2020.

[7] UDEMY. **Machine Learning A-Z™: Hands-On Python & R In Data Science.** Curso disponível em <https://www.udemy.com/course/machinelearning/>. Acesso em: 06 de novembro de 2020.