COMPARAÇÃO ENTRE MODELOS DE CLASSIFICAÇÃO DA SARS-COV-2

Autores

João Victor Mantuan Oliveira
Francisco Cláudio Costa Moraes Júnior

2021

1 Introdução

Visto que a Sars-Cov-2 é um vírus de alto contágio e ao mesmo tempo de alta semelhança com outros vírus já existentes, pois faz parte da classe dos Coronavirus, torna-se necessário uma classificação de alta assertividade para que se tenha uma real noção da disseminação do vírus e a partir dai criar estratégias de combates adequadas e que sejam verdadeiramente efetivas. Portanto, este projeto tem como base o estudo de diferentes modelos de classificação aplicados no contexto da Sars-Cov-2, analisando o desempenho e eficácia dos tais. Por fim, tem-se como objetivo determinar o modelo de classificação que retorne o melhor resultado classificatório ao menor tempo de execução possível.

2 Fundamentação Teórica

2.1 Virus Sars-Cov-2

O primeiro caso identificado ocorreu em dezembro de 2019 em Wuhan, China. Mais conhecido como Covid-19, o vírus faz parte da classe coronavirus da síndrome respiratória aguda grave 2 (Sars-Cov-2), vírus este que tem como principal fator sua transmissão entre seres humanos, ocasionada por gotículas de vírus que são espalhadas pelo ar através de espirros, tosses ou até mesmo pela fala, gotículas estas que são inaladas para os pulmões, causando uma nova contaminação.

Principais sintomas:

- Febre ou calafrios.
- Tosse.
- Falta de ar ou dificuldade em respirar.

- Fadiga.
- Dor de cabeça.
- Congestão nasal ou corrimento nasal.
- Dores musculares ou no corpo.
- Dor de garganta.
- Perda de olfato ou paladar ou ambos.
- Náuseas ou vômitos.
- Diarreia.

Algumas pessoas no entanto não sentem nenhum sintoma quando infectadas, os chamados assintomáticos, mas continuam sendo potenciais transmissores da doença. O Sars-Cov-2 possui um periodo de incubação entre 2 e 14 dias, com uma mediana de 5 dias.

2.2 Dados Sars-Cov-2

Os dados utilizados para treinamento e testes dos modelos implementados foram retirados do Banco de Dados Nacional, openDataSuS, onde contém todos os dados necessários sobre os casos identificados de Covid-19 em todo território nacional.

O arquivo gerado pela plataforma traz mais informações do que as necessárias para um modelo de classificação, portanto o arquivo passou por uma fase de tratamento de dados, o que resultou nos atributos abaixo.

Atributos:

- CS_SEXO (Sexo: 1-Masculino, 2-Feminino, 9-Indefinido).
- NU_IDADE_N (Idade do paciente).
- TP_IDADE (Faixa etária do paciente: 1ª, 2ª ou 3ª idade).
- CS_GESTANT (Período Gestacional: 1º, 2º, 3º trimestre gestacional, 4-Período não informado, 5-Não, 6-Não se aplica, 9-Ignorado).
- CS_RACA (Raça: 1-Branca, 2-Preta, 3-Amarela, 4-Parda, 5-Indígena 9-Indefinido).
- CS_ESCOL_N (Escolaridade: 0-Analfabeto, 1-Fundamental I, 2-Fundamental II, 3-Médio, 4-Superior, 9-Indefinido).
- CO_PAIS (Pais: 1-Brasil).
- CS_ZONA (Zona: 1-Urbana, 2-Rural, 3-Periurbana,9-Indefinido).
- SURTO_SG (Não informado).
- NOSOCOMIAL (Nosocominal: 1-Sim, 2-Não, 9-Ignorado).
- AVE_SUINO (Contato com Aves/Suínos: 1-Sim, 2-Não, 9-Ignorado).
- FEBRE (1-Sim, 2-Não, 9-Ignorado).

- TOSSE (1-Sim, 2-Não, 9-Ignorado).
- GARGANTA (1-Sim, 2-Não, 9-Ignorado).
- DISPNEIA (1-Sim, 2-Não, 9-Ignorado).
- DESC_RESP (Desconforto Respiratório: 1-Sim, 2-Não, 9-Ignorado).
- SATURACAO (1-Sim, 2-Não, 9-Ignorado).
- DIARREIA (1-Sim, 2-Não, 9-Ignorado).
- VOMITO (1-Sim, 2-Não, 9-Ignorado).
- OUTRO_SIN (1-Sim, 2-Não, 9-Ignorado).
- PUERPERA (1-Sim, 2-Não, 9-Ignorado).
- FATOR_RISC (1-Sim, 2-Não, 9-Ignorado).
- CARDIOPATI (1-Sim, 2-Não, 9-Ignorado).
- HEMATOLOGI (1-Sim, 2-Não, 9-Ignorado).
- SIND_DOWN (1-Sim, 2-Não, 9-Ignorado).
- HEPATICA (1-Sim, 2-Não, 9-Ignorado).
- ASMA (1-Sim, 2-Não, 9-Ignorado).
- DIABETES (1-Sim, 2-Não, 9-Ignorado).
- NEUROLOGIC (1-Sim, 2-Não, 9-Ignorado).
- PNEUMOPATI (1-Sim, 2-Não, 9-Ignorado).
- IMUNODEPRE (1-Sim, 2-Não, 9-Ignorado).
- RENAL (1-Sim, 2-Não, 9-Ignorado).
- OBESIDADE (1-Sim, 2-Não, 9-Ignorado).
- OBES_IMC (IMC do paciente).
- OUT_MORBI (1-Sim, 2-Não, 9-Ignorado).
- VACINA (1-Sim, 2-Não, 9-Ignorado).
- ANTIVIRAL (1-Sim, 2-Não, 9-Ignorado).
- TP_ANTIVIR (Tipo do antiviral: 1-Oseltamivir, 2-Zanamivir, 3-Outro).
- HOSPITAL (1-Sim, 2-Não, 9-Ignorado).
- UTI (1-Sim, 2-Não, 9-Ignorado).
- SUPORT_VEN (Suporte Ventosa: 1-Sim, invasivo, 2-Sim, não invasivo, 3-Não, 9-Ignorado).

- RAIOX_RES (Raio X - tórax: 1-Normal, 2-Infiltrado, 3-Consolidação, 4-Misto, 5-Outro, 6-Não realizado, 9-Ignorado).
- AMOSTRA (1-Sim, 2-Não, 9-Ignorado).
- TP_AMOSTRA (Tipo de amostra: 1-Secreção, 2-Lavado, 3-Tecido, 4-Outro, 5-LCR, 9-Ignorado).
- PCR_RESUL (REsultado PCR: 1-Detectável, 2-Não detectável, 3-Inconclusivo, 4-Não realizado, 5-Aguardando resultado, 9-Ignorado).
- POS_PCRFLU (Agente Etiológico: 1-Sim, 2-Não, 9-Ignorado).
- TP_FLU_PCR (Tipo influenza: 1-Influenza A, 2-Influenza B, 9-Ignorado).
- POS_PCROUT (1-Sim, 2-Não, 9-Ignorado).
- PCR_VSR (1-Sim, 2-Não, 9-Ignorado).
- PCR_PARA1 (1-Sim, 2-Não, 9-Ignorado).
- PCR_PARA2 (1-Sim, 9-Ignorado).
- PCR_PARA3 (1-Sim, 9-Ignorado).
- PCR_PARA4 (1-Sim, 9-Ignorado).
- PCR_ADENO (1-Sim, 9-Ignorado).
- PCR_METAP (1-Sim, 9-Ignorado).
- PCR_BOCA (1-Sim, 9-Ignorado).
- PCR_RINO (1-Sim, 9-Ignorado).
- PCR_OUTRO (1-Sim, 9-Ignorado).
- CRITERIO (1-Laboratorial, 2-Clinico Epidemiológico, 3-Clinico, 4-Clinico Imagem, 9-Ignorado).
- EVOLUCAO (1-Cura, 2-Óbito, 3-Óbito por outras causas, 9-Ignorado).
- PCR_SARS2 (1-Sim, 9-Ignorado).
- DOR_ABD (1-Sim, 2-Não, 9-Ignorado).
- FADIGA (1-Sim, 2-Não, 9-Ignorado).
- PERD_OLFT (1-Sim, 2-Não, 9-Ignorado).
- PERD_PALA (1-Sim, 2-Não, 9-Ignorado).
- TOMO_RES (Tomografia: 1-Tipico Covid, 2-Indeterminado, 3-Atipico, 4-Negativo para pneumonia, 9-Ignorado).
- TP_TES_AN (Tipo do teste antigênico: 1-Imunofluorescência, 2-Teste rápido, 9-Ignorado).

- RES_AN (Resultado do teste antigênico: 1-Positivo, 2-Negativo, 3-Inconclusivo, 4-Não realizado, 5-Aguardado resultado, 9-Ignorado).
- POS_AN_FLU (Agente Etiológico: 1-Sim, 2-Não, 9-Ignorado).
- TP_FLU_AN (Agente Etiológico: 1-Influenza A, 2-Influenza B, 9-Ignorado).
- POS_AN_OUT (Agente Etiológico: 1-Sim, 2-Não, 9-Ignorado).
- AN_SARS2 (Agente Etiológico: 1-Sim, 9-Ignorado).
- TP_AM_SOR (Tipo de amostra sorológica: 1-Sangue/Plasma/Soro, 2-Outro, 9-Ignorado).
- TP_SOR (Tipo Sorologia: 1-Teste rápido, 2-Elise, 3-Quimiluminescência, 4-Outro, 9-Ignorado).
- RES_IGG (Resultado Sorológico: 1-Sim, 9-Ignorado).
- RES_IGM (Resultado Sorológico: 1-Sim, 9-Ignorado).
- RES_IGA (Resultado Sorológico: 1-Sim, 9-Ignorado).
- VACINA_COV (1-Sim, 9-Ignorado).

Saida:

- CLASSI_FIN (1-Negativo para Covid, 5-Positivo para Covid).

3 Metodologia

Os dados referentes ao período de 01 de janeiro de 2021 à 09 de agosto de 2021 foram baixados da plataforma openDataSuS, uma plataforma do Governo Federal usada para documentar de forma oficial os dados nacionais do Sars-Cov-2.

A escolha do período dos dados citados acima decorre da possibilidade de termos dados melhores distribuídos, com menor viés e variância, visto que em 2020 ainda não se tinha começado a vacinação em território nacional, logo buscamos em 2021 dados com uma quantidade mais equilibrada entre vacinados e não vacinados. A partir disto os dados passaram pelos seguintes processos:

- Tratamento dos dados.
- Construção dos modelos de análise.
- Aplicação dos dados tratados nos modelos construídos.
- Análise dos dados e gráficos retornados pelos modelos.

3.1 Tratamento dos Dados

Os dados obtidos na openDataSuS e salvos em srag_cov contém uma quantidade alta demais de colunas, contendo dados desnecessários para um modelo de classificação, por isso foi feita uma seleção manual das colunas que se mostravam com dados mais relevantes e menos redundantes.

Mas em vez de uma seleção manual das colunas, poderia ser feito também uma seleção a partir do modelo PCA, para reduzir a dimensionalidade e obter apenas as features que mais impactariam o modelo de forma positiva. No entanto a escolha por seleção manual foi feita, para que em uma próxima versão do código se pudesse fazer uma comparação dos resultados com uma escolha PCA, vendo assim a real diferença entre escolha manual x PCA.

Após feita a escolha manual das colunas e salva as escolhas no arquivo srag_cov _columns, foram aplicados os dois arquivos, original e das colunas, no algoritmo etl.py para ocorrer o drop das colunas desnecessárias e os demais tratamentos necessários, como por exemplo, tratar linhas com dados nulos e e converter linhas com dados textuais em dados numéricos.

3.2 Construção dos modelos

Antes dos modelos em si serem construídos, foi feita a normalização dos dados utilizando a técnica min-max.

A principio a ideia era usar 4 modelos de classificação e compará-los, no entanto encontrou-se alguns empecilhos durante a execução dos treinamentos em alguns modelos.

Modelos planejados:

- Regressão Logística.
- MLP Classifier com SGD.
- SVM.
- KNN.

Nos modelos MLP Classifier com SGD e KNN ocorreram problemas de tempo de execução, neste projeto foi utilizado a plataforma Colab, devido seu poder de processamento, no entanto por varias vezes houveram problemas em que o Colab desconectava a conta, interrompendo assim a compilação dos modelos ainda no período de treinamento.

Uma alternativa para se solucionar o modelo KNN é utilizar PCA para diminuir a dimensão da matriz, ocorrendo assim em menor tempo de análise. Os demais modelos concluíram o treinamento sem maiores problemas e o os problemas enfrentados no modelo MLP foram contornados.

3.3 Aplicação dos dados nos modelos e retorno dos modelos

Como dito anteriormente, os dados foram normalizados e além disso foi utilizada a técnica K-folds, ao todo foram utilizados 5-folds em cada rodagem dos modelos e observado o retorno das métricas de cada modelo.

As métricas utilizadas:

- Acurácia.
- Precisão.
- F1-Score.
- Recall.

4 Experimentos

Após o tratamento dos dados, normalização e definição de 5-folds, os dados foram submetidos aos modelos utilizando os seguintes parâmetros:

4.1 Parâmetros, Classes e Modelos

Parâmetros Regressão Logística:

- C: [1, 10, 100, 1000, 10000]

`Classe Regressão Logística:`

```python
class LogisticReg:
    def __init__(self, n_iter=200):
        self.n_iter = n_iter

    def fit(self, X_train, y_train, params, cv):
        self.estimator = LogisticRegression(max_iter=self.n_iter)
        gs = GridSearchCV(estimator=self.estimator, param_grid=params, cv=cv)
        gs.fit(X_train, y_train)
        self.best_params = gs.best_params_

        self.estimator = LogisticRegression(max_iter=self.n_iter, **self.best_params)
        self.estimator.fit(X_train, y_train)

    def predict(self, X_test):
        pred = self.estimator.predict(X_test)
        self.best_estimator = self.estimator
        return pred

    def calc_metrics(self, y_test, pred):
        self.metrics = compute_metrics(y_test, pred)
        self.metrics.rename(columns={0: 'Reg. Log.'})

    def plot_metrics(self):
        plt.title('Métricas do Modelo')
        self.metrics.rename(columns={0: 'Reg. Log.'})
        sns.barplot(x=self.metrics.index.to_list(), y=self.metrics[0])
        plt.show()
```

Figura 1 – Modelo Regressão Logística

Parâmetros SVM:

- gamma_range = [2e-3, 2e-1, 2e1]
- c_range = [2e-1, 2e1, 2e3]

`Classe SVM:`

```python
class SVM:
    def __init__(self, n_iter=200):
        self.n_iter = n_iter

    def fit(self, X_train, y_train, params, cv):
        self.estimator = SVC(kernel='rbf', max_iter=self.n_iter)
        gs = GridSearchCV(self.estimator, param_grid=params, cv=cv)
        gs.fit(X_train, y_train)
        self.best_params = gs.best_params_

        self.estimator = SVC(kernel='rbf', max_iter=self.n_iter, **self.best_params)
        self.estimator.fit(X_train, y_train)

    def predict(self, X_test):
        pred = self.estimator.predict(X_test)
        self.best_estimator = self.estimator
        return pred

    def calc_metrics(self, y_test, pred):
        self.metrics = compute_metrics(y_test, pred)
        self.metrics.rename(columns={0: 'SVM'})

    def plot_metrics(self):
        plt.title('Métricas do Modelo')
        self.metrics.rename(columns={0: 'SVM'})
        sns.barplot(x=self.metrics.index.to_list(), y=self.metrics[0])
        plt.show()
```

Figura 2 – Modelo SVM

Parâmetros MLP:

- hidden_layer_sizes: [4, 16, 32, 64, 128],
- alpha: [1e-5, 1e-4, 1e-3, 1e-2, 1e-1],
- batch_size: [8, 16, 32, 64, 128],
- learning_rate_init: [1e-4, 1e-3, 1e-2, 1e-1]

`Classe MLP:`

```python
class MLP:
    def __init__(self, n_iter=200):
        self.n_iter = n_iter

    def fit(self, X_train, y_train, params, cv):
        self.estimator = MLPClassifier(solver='sgd', max_iter=self.n_iter, verbose=2)

        gs = GridSearchCV(estimator=self.estimator, param_grid=params, cv=cv)
        gs.fit(X_train, y_train)
        self.best_params = gs.best_params_

        self.estimator = MLPClassifier(solver='sgd', max_iter=self.n_iter, **self.best_params)
        self.estimator.fit(X_train, y_train)

    def predict(self, X_test):
        pred = self.estimator.predict(X_test)
        self.best_estimator = self.estimator
        return pred

    def calc_metrics(y_test, pred):
        self.metrics = compute_metrics(y_test, pred)
        self.metrics.rename(columns={0: 'MLP'})

    def plot_metrics(self):
        plt.title('Métricas do Modelo')
        self.metrics.rename(columns={0: 'MLP'})
        sns.barplot(x=self.metrics.index.to_list(), y=self.metrics[0])
        plt.show()
```

Figura 3 – Modelo MLP com SGD

É importante ressaltar que os parâmetros foram testados através do GridSearch e logo após os modelos foram treinados com os parâmetros que retornaram os melhores percentuais nas métricas estabelecidas, sendo elas, Acurácia, Precisão, F1-Score, Recall.

4.2 Resultados

Resultados de todas as métricas em cada modelo testado, embora todos estejam próximos em taxas percentuais é possível verificar que a Rede Neural obteve melhores resultados, em uma quantidade de iterações consideravelmente baixa (3 iterações) e um bom tempo de execução do modelo em treinamento (40min). Logo após tem-se bons resultados da Regressão Logística, com número de iterações maior (100 iterações) e tempo de execução um pouco melhor que o modelo MLP (35min). Em último lugar tem-se o modelo SVM em uma quantidade de iterações mais alta (200), embora ainda não seja tão alta em um contexto de SVM, além disto teve o maior tempo de execução (1hora).

`Resultados Regressão Logística:`

```
Melhores parâmetros: {'C': 1}

Métricas:
Acurácia    0.894227
Precisão    1.000000
Recall      0.894227
F1-Score    0.944160
Name: 0, dtype: float64
```

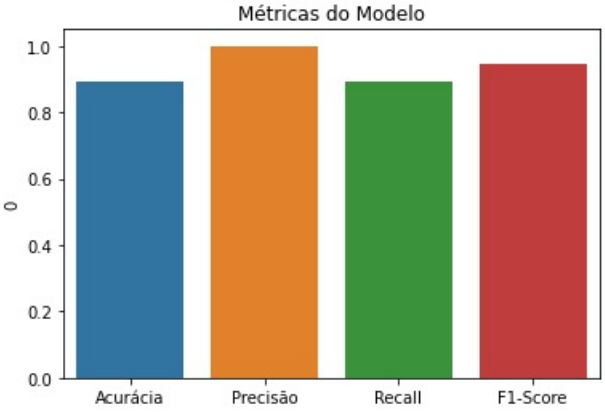

Figura 4 – Métricas Regressão Logística

Resultados SVM:

```
Melhores parâmetros: {'C': 20.0, 'gamma': 20.0}

Métricas:
Acurácia    0.725242
Precisão    1.000000
Recall      0.725242
F1-Score    0.840742
Name: 0, dtype: float64
```

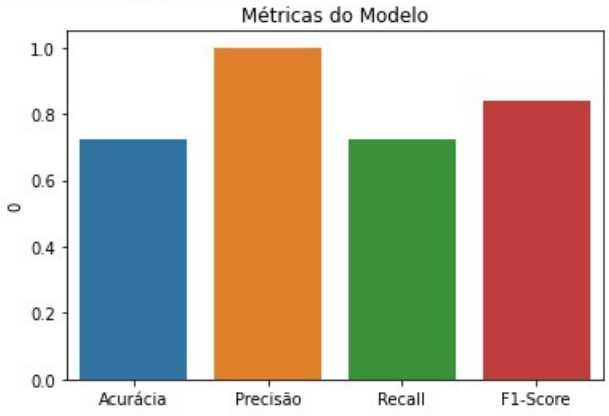

Figura 5 – Métricas SVM

Resultados MLP:

```
Melhores parâmetros: {'batch_size': 8, 'hidden_layer_sizes': 32}

Métricas:
                MLP
Acurácia  0.934400
Precisão  1.000000
Recall    0.934400
F1-Score  0.966088
```

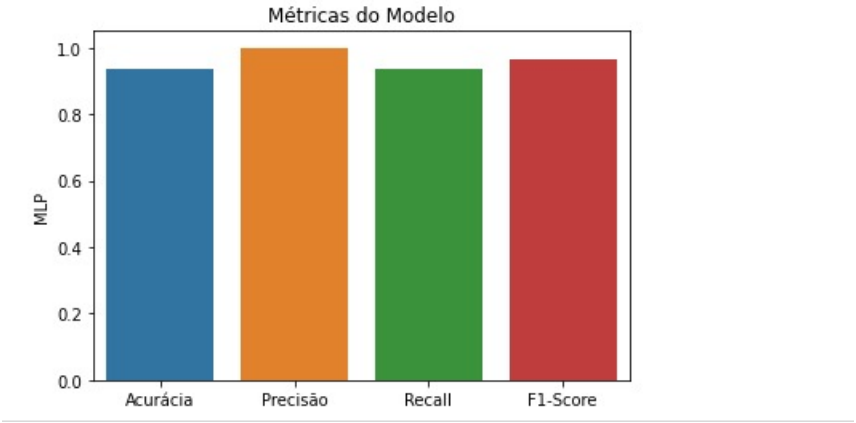

Figura 6 – Métricas MLP

`Resultados Finais:`

	Reg. Logística	SVM	MLP
Acurácia	0.894227	0.72643	0.934400
Precisão	1.000000	1.00000	1.000000
Recall	0.894227	0.72643	0.934400
F1-Score	0.944160	0.84154	0.966088

Figura 7 – Métricas Finais

5 Considerações finais

Este projeto inicial conseguiu alcançar os resultados esperados, desde o inicio houve uma preocupação em seguir uma metodologia a risca para evitar possíveis erros pelo caminho. Logo chega-se ao final do projeto com dados tratados, modelos implementados corretamente e métricas retornadas com alto nível de precisão. É importante ressaltar que muito ainda pode ser feito para alcançar resultados ainda mais precisos, como a implementação do PCA, verificação de novos parâmetros e novos modelos e também utilizar um acelerador de processamento como por exemplo o Numba, podem ajudar a conseguir resultados ainda melhores em menos tempo de execução.

Além disto, é possível usar os dados na forma atual para se fazer uma análise de séries temporais, para prever a evolução do Sars-Cov-2 ao longo do tempo em território nacional, como projeto de investigação futura.

6 Divisão das tarefas

- João Victor Mantuan Oliveira: Implementação de todos os modelos.
- Francisco Cláudio Costa Moraes Júnior: Implementação do ETL e escrita do artigo.

Referências

openDataSuS:
`https://opendatasus.saude.gov.br/` Nenhuma citação no texto.

Sklearn SVM:
`https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
Nenhuma citação no texto.

Sklearn MLP:
`https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html` Nenhuma citação no texto.

Sklearn Regressão Logística:
`https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html` Nenhuma citação no texto.