Neural Computing and Applications (2022) 34:7523–7536
https://doi.org/10.1007/s00521-021-06884-w

ORIGINAL ARTICLE

Multi-criteria text mining model for COVID-19 testing reasons and symptoms and temporal predictive model for COVID-19 test results in rural communities

Laith Abu Lekham[1,2] · Yong Wang[1] · Ellen Hey[2] · Mohammad T. Khasawneh[1]

Received: 31 July 2021 / Accepted: 19 December 2021 / Published online: 5 January 2022

Abstract

This study is conducted to build a multi-criteria text mining model for COVID-19 testing reasons and symptoms. The model is integrated with a temporal predictive classification model for COVID-19 test results in rural underserved areas. A dataset of 6895 testing appointments and 14 features is used in this study. The text mining model classifies the notes related to the testing reasons and reported symptoms into one or more categories using look-up wordlists and a multi-criteria mapping process. The model converts an unstructured feature to a categorical feature that is used in building the temporal predictive classification model for COVID-19 test results and conducting some population analytics. The classification model is a temporal model (ordered and indexed by testing date) that uses machine learning classifiers to predict test results that are either positive or negative. Two types of classifiers and performance measures that include balanced and regular methods are used: (1) balanced random forest and (2) balanced bagged decision tree. The balanced or weighted methods are used to address and account for the biased and imbalanced dataset and to ensure correct detection of patients with COVID-19 (minority class). The model is tested in two stages using validation and testing sets to ensure robustness and reliability. The balanced classifiers outperformed regular classifiers using the balanced performance measures (balanced accuracy and G-score), which means the balanced classifiers are better at detecting patients with positive COVID-19 results. The balanced random forest achieved the best average balanced accuracy (86.1%) and G-score (86.1%) using the validation set. The balanced bagged decision tree achieved the best average balanced accuracy (83.0%) and G-score (82.8%) using the testing set. Also, it was found that the patient history, age, testing reasons, and time are the key features to classify the testing results.

Keywords Classification · Community health · Machine learning · Population health analytics · Primary care · Text mining

1 Introduction

The pandemic of the novel SARS-CoV-2 Coronavirus (COVID-19) has made big changes in the world. The World Health Organization (WHO) declared a global emergency on January 30, 2020, after the outbreak of COVID-19 in China [1]. COVID-19 is recognized as an acute respiratory virus because it attacks mainly the upper respiratory system in the human body [2]. Many people have died due to this highly contagious and deadly airborne virus. The virus can spread very fast, and it can be transferred very easily from one person to another with a high infection rate [3]. The outbreak of COVID-19 has created many critical challenges and issues for public health, research, and medical communities.

COVID-19 forced many countries to shut down their operations as the virus put the lives of millions of people around the world at high risk. Numerous studies continue to be conducted in several areas to combat, control, reduce, and contain the impacts and consequences of the COVID-

✉ Yong Wang
yongwang@binghamton.edu

[1] Systems Science and Industrial Engineering Department, State University of New York at Binghamton, Binghamton, USA

[2] Finger Lakes Community Health, Geneva, USA

19 virus. Many studies discussed different solutions and issues related to COVID-19 in many areas such as transportation, education, supply chain, economy, trade, and politics. The next section is focused on the review of papers that are relevant to our research topic in text mining and machine learning applications.

2 Literature review

In the recent literature, numerous studies discussed various predictive and text mining models related to COVID-19. A large number of the discussed models focused on the prediction of future cases of COVID-19 and hospitalization rate for planning purposes. Countless predictive methods and techniques were used to predict different variables associated with the COVID-19 virus, such as machine learning algorithms, logistic regression, correlation analysis, and time-series analysis.

Several studies have focused on investigating the association between the mobility of people and the spread of COVID-19. The authors of [4] studied the ramification of control measures and human mobility on the COVID-19 pandemic in China using time-series correlation analysis. They used real-time human mobility data that represents the daily population travel from Wuhan to other cities in China. They found that there is a clear association between the growth rate of COVID-19 cases and the mobility (travel) before the control measures took place. The direction and values of the correlation between the growth rate and mobility changed after the implementation of containment strategies. They concluded that the control measures adopted by the government of China played a key role in mitigating and reducing the spread of COVID-19. Another study analyzed the effect of mobility on COVID-19 cases [5]. They found that the density of cars, pedestrians, and transit traffic impacted the spread of the COVID-19 with a time lag of 15–20 days. This finding was obtained based on a comparison of the global COVID-19 cases with changes in the number of cars, pedestrians, and transit traffic.

Many studies have used machine learning and data science methods and techniques to predict different factors associated with COVID-19, such as number of cases, number of required beds, hospitalization rate, and healthcare workers needed. Most models are developed using SEIR (susceptible, exposed, infectious, and recovered) and SIR (susceptible, infectious, and recovered) models, curve-fitting models, agent-based simulation models, and machine learning models [6]. One study did a comprehensive review of many papers that used machine learning methods to predict the confirmed cases of COVID-19 [7]. They grouped the methods used into three categories: (1) traditional and deep learning regression techniques, (2) network analysis, and (3) social media and search-based techniques. Also, they discussed the challenges of using machine learning methods in this area. Another study used machine learning to predict the trends of the COVID-19 in both some individual countries and around the globe [8]. They used an integrated model of logistic regression and the FbProphet time-series predictive model (non-linear time-series forecasting model). The logistic regression model is used to fit the cap of epidemic trend, and the FbProphet model is used to predict the epidemic curve and trend. Another paper used machine learning algorithms to produce a 10-day projection of the number of new confirmed cases, number of recoveries, and number of deaths [9]. They used multiple regression models including support vector machine (SVM), least absolute shrinkage and selection operator (LASSO), and exponential smoothing (ES).

Study [10] developed an integrated model using cloud computing and machine learning to predict COVID-19 growth rate and trends. They developed an improved machine learning mathematical model by fitting generalized inverse Weibull distribution using iterative weighting. Also, they used a cloud computing platform for real-time data to anticipate COVID-19 growth rates in countries around the world. Study [11] developed a hybrid model of two algorithms: (1) adaptive and network-based fuzzy inference system algorithm and (2) perceptron-imperialist multi-layered competitive algorithm to predict COVID-19 confirmed cases and death rates. They used data from Hungary for their model. In study [6], the authors discussed the effect of crucial and unprecedented factors and uncertainties on predictive models for COVID-19 related numbers. They addressed and discussed the effect of hospital settings and capacity, daily test capacity and rate, population density, demographics, vulnerable people, and income versus commodities (poverty). Also, they discussed the development and importance of auto-tuned and dynamic data-driven predictive models that are mathematically proven.

A considerable number of studies discussed the use of social media to extract the reported symptoms of COVID-19 by people using text mining techniques. CDC identified nine main symptoms for COVID-19 that include shortness of breath (difficulty breathing), cough, fever, chills, repeated shaking with chills, muscle pain, headache, sore throat, and loss of taste or smell [12, 13]. One study analyzed the symptoms reported by people on Twitter using a typical text mining framework [14]. They found that taste disturbance, anosmia, and psychiatric issues are the most common reported symptoms. Another research discussed the use of tweets to extract the most common symptoms reported by people using a standard text mining model [14]. Also, the authors compared their results with the

identified symptoms in other studies. They found that fever, pyrexia, and cough are the most frequently reported symptoms. Besides, they found that other mild symptoms, such as anosmia and ageusia, are frequently reported. Paper [15] used a symptomatology text mining approach to analyze the symptoms reported by people on social networks in Colombia (Bogota). Also, the results of the study can help in understanding the spread of COVID-19 and any new variants.

In the literature, most of the machine learning studies concerned with COVID-19 are focused on predicting the key pandemic trends such as the number of confirmed cases, the mortality and hospitalization rates, and the number of recoveries. Also, the majority of the studies used regression algorithms to predict those variables. Limited research discussed the prediction of patients' testing results (either positive or negative) for COVID-19. Few studies talked about using machine learning predictive models for COVID-19 in rural areas. Besides, most of the text mining models used standard techniques to analyze COVID-19 symptoms and test reasons. Limited research used text mining to conduct population health analysis related to COVID-19 in rural areas.

In summary, this research has two primary research objectives: (1) develop a multi-criteria detection text mining model to categorize COVID-19 testing reasons and symptoms and conduct population health analysis and (2) develop a temporal machine learning classification predictive model for COVID-19 patients testing results (positive/negative). The main contribution of this study is the application of the models developed. Two different machine learning techniques (text mining and classification) were utilized to develop an integrated model that can be preventative tool for COVID-19 in rural areas. The results and models are tailored for primary care that works with underserved populations. Also, the developed text mining model can detect multiple testing reasons and symptoms for COVID-19 in underserved communities. Besides, the predictive model is designed to accurately anticipate the test results of COVID-19 tests. This can help medical centers to be proactive and educate patients if there is a possibility that their test results are positive especially when the laboratory results are delayed. To the best of the authors' knowledge, no paper has discussed or developed such application and integrated model for primary care that works in rural areas.

3 Data and methods

The data used in this study are from an outpatient primary care healthcare provider that works with underserved populations in upstate New York, with a focus on COVID-19-related appointments. The healthcare provider is the Finger Lakes Community Health (FLCH). FLCH is a federally qualified health center, and it has eight medical centers located in the Finger Lakes area. In addition, FLCH works and collaborates with two other medical centers in upstate New York.

The dataset used in this study represents COVID-19 testing appointments at FLCH from March 2020 (i.e., start of the pandemic) to April 2021. The dataset is collected from the eCW electronic health record system (EHR) using a structured query language (SQL). The dataset contains 6,895 appointments with 14 features and one binary label representing the COVID-19 status. The features represent patient demographics and appointment attributes such as age, gender, race, appointment date, testing site, and testing reason. The label represents COVID-19 testing results as either positive or negative. The features and label details are shown in Fig. 1. The 'testing reason' feature is the unstructured text corpus used to develop the multi-criteria detection text mining model for reasons and symptoms categorization. This feature represents the notes written by the schedulers about the test reasons conveyed by patients. The unstructured feature will be replaced with the structured outcome of the text mining model to be used in the predictive classification model for the test results.

This research was driven by the need to derive and categorize COVID-19 testing reasons and symptoms from an unstructured clinical text dataset and to predict COVID-19 patients testing results. This will allow the decision makers and clinical staff to get accurate insights into COVID-19 in rural underserved communities and to better assemble resources needed to serve these communities. This study is carried out in two main phases. The first phase focuses on building the multi-criteria text mining model for COVID-19 testing reasons and symptoms. The second phase focuses on building the temporal classification model for COVID-19 test results. Figure 2 illustrates the methodology flow chart for building both the text mining and predictive models. Python 3.7 data science packages

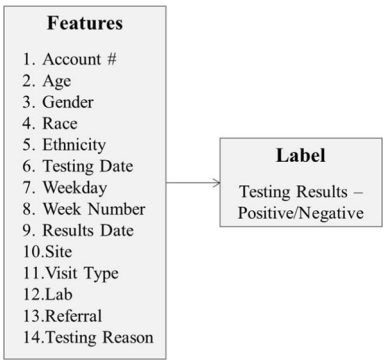

Fig. 1 Features and label avalaibale in the dataset

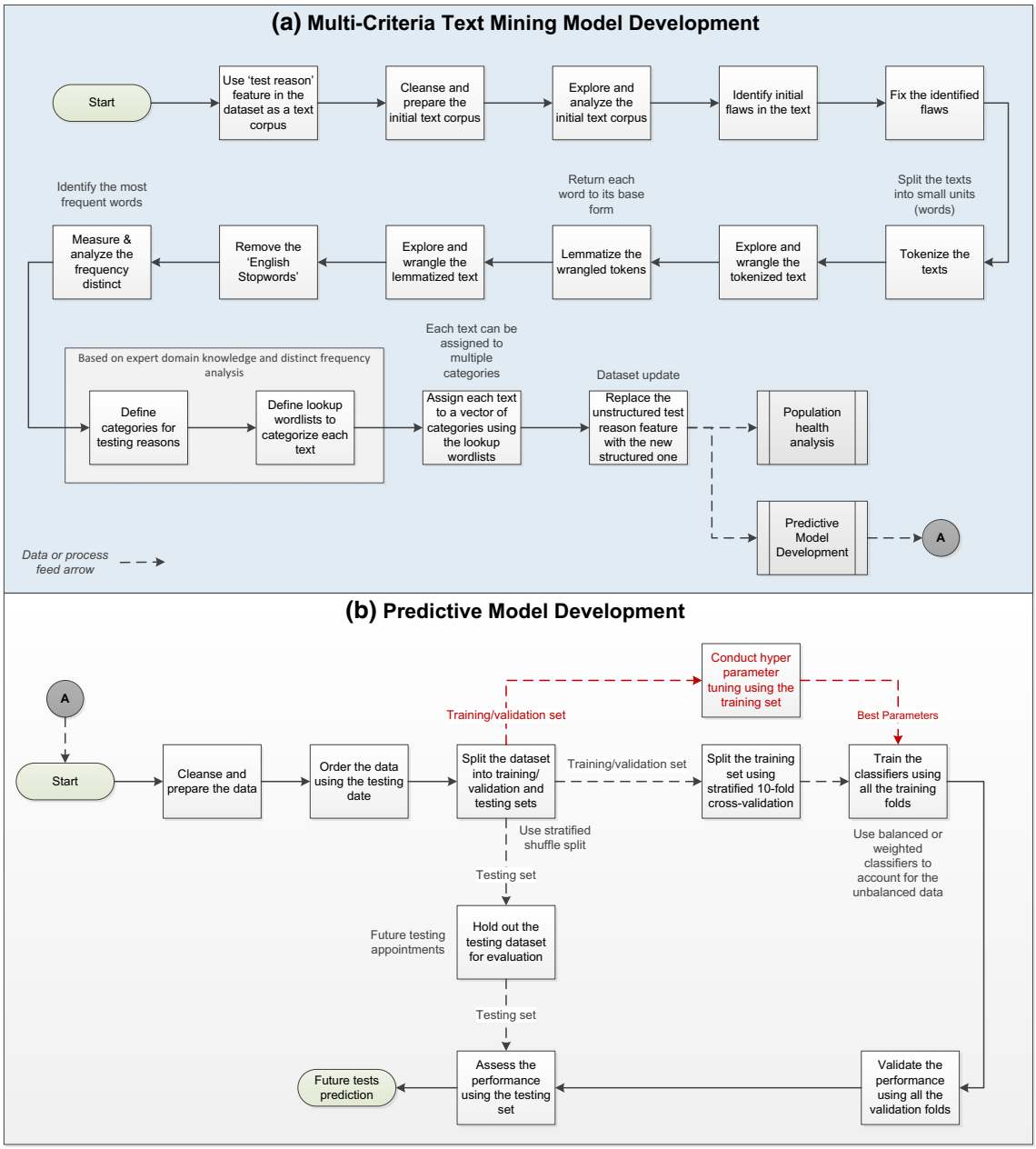

Fig. 2 Methodology flow chart for building the multi-criteria text mining model (**a**) and the temporal classification predictive model (**b**)

that include Scikit-Learn [16], Imbalanced-Learn [17], NLTK [18], Pandas [19], Numpy [20], Category-Encoders [21], Seaborn [22], Yellowbrick [23], and Matplotlib [24] were used in this study.

The first model (first phase) is developed in three main sub-phases that include the following: (1) initial text preparation and exploration; (2) deep text wrangling using text mining; and (3) text tagging and categorization. Each sub-phase is carried out in several steps as explained later in the manuscript. The model development flow chart is explained and shown in Fig. 2a.

The *first sub-phase* is carried out in five steps. First, the 'testing reason' is used as the text corpus to develop the text mining model. Second, the corpus is initially prepared and cleansed. Third, the corpus is initially explored and analyzed to look for trends and data flaws. Fourth, some major misspellings and grammatical mistakes are identified. Finally, the identified mistakes are corrected to remove serious text issues.

The *second sub-phase* is carried out in five main steps. First, the corpus is tokenized by splitting each text into individual words (smaller units of language). Second, the tokenized texts are explored for patterns and flaws. After

that, the tokenized corpus is wrangled based on the findings of text exploration and analysis. Third, the wrangled corpus is lemmatized where each word is returned to its original form or root [18]. The lemmatization is conducted by considering the parts of speech of each text. Fourth, the lemmatized corpus is explored and wrangled as in the previous step. Fifth, the English stopwords were removed from the prepared corpus. Stopwords are commonly used words, such as pronouns and auxiliary verbs, that search engines ignore in their search inquiries [18]. The stopwords are removed because they are not favorable; therefore, space and processing time can be saved. This sub-phase is critical for the categorization of test reasons and identification of the look-up wordlists.

The *third sub-phase* is conducted in five steps. First, the frequency of the words in the corpus is measured in which the most common words are identified. After that, the frequency distribution of the words is explored looking for trends and insights. Second, the findings are reviewed by the clinical team, which is a critical step to defining the categories for test reasons and symptoms and their associated wordlists. Nine categories are identified to classify the test reasons and symptoms for COVID-19. Third, a look-up wordlist is created for each category. The look-up wordlists are used to tag each text to a vector of categories as each appointment can be assigned or tagged to one or multiple categories. For example, a patient can be tested for multiple reasons, such as experiencing symptoms and contacted by the health authorities. Fourth, the text mining algorithm scans the corpus looking for the look-up words to classify each appointment to one or more of the test reasons categories. The mapping is done using a multi-criteria tagging technique where each text is assigned to a binary nine-digit vector and each digit represents a testing category. If the text is tagged to a category in which the corresponding digit will have a value of one; zero otherwise. After finishing the tagging process, the vector will be transformed to a multi-class one-digit vector similar to the 'Label Powerset Transformation' method [25, 26]. Finally, the tagging and categorization outcome are used to create a new structured categorical feature for the 'testing reason' that will be used in the population health analysis and development of the predictive model in the second phase. Table 1 provides the pseudocode for the multi-criteria text mining algorithm.

The second model (second phase) is built on multiple steps. The model development flow chart is illustrated in Fig. 2b. The first step in building the model is preparing the dataset where the missing data are handled, data errors are corrected, outliers are evaluated, and data are converted and encoded. After that, the dataset is ordered using the testing date (converted to a numerical variable) and the week number of the testing date to make the model temporal. The wrangled dataset is split into two sets for training, validation, and testing. The split is done using the 'Stratified Shuffle Split' approach (90%:10%). Also, the training dataset is divided using the 'Stratified K-fold Cross-Validation' approach (tenfold) to validate the performance of the model. The two-way dataset division allows the model to be tested in a two-stage manner. This approach ensures the robustness and reliability of the model.

The Stratified K-fold method is a robust and popular validation and testing approach in machine learning because it infuses the K-fold Cross-Validation and the Stratified Shuffle Split approaches [16, 27–29]. This method is well-founded to account for unbalanced labels and biased data issues. The stratified folds are generated by maintaining the percentage of samples of each class in each fold as in the whole set [16, 30]. In this study, the division was carried out using the percentage of each class to ensure that the model addresses both classes properly (positive and negative). The testing results label is unbalanced because most patients tested negative (4.21% positivity rate).

Two kinds of classifiers are trained: (1) balanced or weighted classifiers and (2) regular classifiers (unbalanced and unweighted). The balanced or weighted classifiers work by resampling or giving weights to the classes. These classifiers are designed to mitigate the issues of unbalanced datasets [17]. Three balanced machine learning classifiers are used: include random forest (RF), bagging decision tree (BDT), and gradient boosting (GB). In addition, three regular machine learning classifiers that included linear discriminant analysis (LDA), K-nearest neighbor (K-NN), and Gaussian process (GP) are used. Before the training process, cross-validated hyperparameter tuning is done using the 'Random Search Optimization' approach [31] to find the best parameters for each classifier, if any. The tuning process is done independently from training, validation, and testing. The tuning was conducted on the training dataset. Table 2 provides the pseudocode for the temporal predictive model algorithm.

The performance of the text mining model was evaluated manually by the relevant stakeholders. For the performance evaluation of the predictive model, six measures are used: (1) accuracy, (2) F1-score, (3) recall (sensitivity), (4) precision (positive predictive value), (5) balanced accuracy, and (6) geometric mean score (G-score). The first four measures are common in practice, so they were not described in details in this paper. For more information about the first four metrics, the reader is advised to refer to [16, 32]. The last two measures are used to evaluate the performance of classifiers on unbalanced datasets (balanced and geometry performance measures). These metrics can reveal any prediction issues of the minority class in

Table 1 A pseudo code for the multi-criteria text mining algorithm

Text Feature Preparation

Collect feature $\boldsymbol{D}$ (text documents) $\rightarrow (\boldsymbol{d_1}, \boldsymbol{d_2}, \ldots, \boldsymbol{d_n})$
For $\boldsymbol{d_n}$ in $\boldsymbol{D}$:
    Tokenize $\boldsymbol{d_n}$
    Lemmatize $\boldsymbol{d_n}$
    Remove English stopwords from $\boldsymbol{d_n}$
    Return $\rightarrow$ *converted & deconstructed* $\boldsymbol{d_n} = \boldsymbol{k} \rightarrow (\boldsymbol{k_1}, \boldsymbol{k_2}, \ldots, \boldsymbol{k_j})$
Define *test reasons and symptoms categories* $\boldsymbol{C} \rightarrow (c_1, c_2, \ldots, c_m)$
Define *look-up wordlists* $\boldsymbol{W} \rightarrow (\boldsymbol{w_1}, \boldsymbol{w_2}, \ldots, \boldsymbol{w_m})$
Define a *binary nine-digit mapping vector* $V \rightarrow (\boldsymbol{v_{1,n}} = 0, \boldsymbol{v_{2,n}} = 0, \ldots, \boldsymbol{v_{9,n}} = 0)$

Tagging and Categorizing

For $\boldsymbol{d_n}$ in D:
    For $\boldsymbol{k_j}$ in $\boldsymbol{d_n}$:
        For $\boldsymbol{w_m}$ in $\boldsymbol{W}$:
            if $\boldsymbol{k_j}$ is in $\boldsymbol{w_m}$:
                Tag $\boldsymbol{k_j}$ to $\boldsymbol{c_m}$
                Assign $\boldsymbol{v_{m,n}} = 1$
            else:
                Tag $\boldsymbol{k_j}$ to '$\boldsymbol{c_6}$ – 'Other/Unreported/Referral category'
                Assign $\boldsymbol{v_{6,n}} = 1$
For $\boldsymbol{v_n}$ in $\boldsymbol{V}$:
    Transform $\boldsymbol{v_n}$ to a multi-class one-digit vector $\rightarrow \boldsymbol{vm_n}$
Return $\rightarrow$ structured categorical feature $\boldsymbol{DS}$

favor of the majority class. The balanced accuracy is the average recall attained for each class [33]. The G-score is the root of the product of class-wise sensitivity, and it equals the squared root of the product of the specificity and sensitivity (recall) for binary classification [17, 34]. This measure maximizes the accuracy of each class while balancing these accuracies. The final judgment and evaluation of the model performance are based on the balanced and geometry measures as they allow us to see the ability of the model of detecting the minority class, which represents patients with positive results. The focus is on avoiding misclassifying positive results as negative (high risk). The first four measures give more focus to the majority class (negative patients). In the real world, the misclassification of negative patients as positive (low risk) is tolerable. Equations (1) and (2) are used to calculate the balanced accuracy and G-score, respectively, as follows:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{1}$$

$$G - \text{Score} = \left(\prod_{i=1}^{n} \text{Sensitivity}_i\right)^{\frac{1}{n}} = \sqrt{\text{Sensitivity}_{\text{class1}} \times \text{Sensitivity}_{\text{class2}}} \tag{2}$$

where *TP* is the true positive, *FN* is the false negative, *TN* is the true negative, and *FP* is the false positive. All of them are calculated from the confusion matrix. The reader is advised to refer to [35] for more details about these terms. Also, the receiver operating characteristics (ROC) analysis is used to evaluate the performance of one classifier (well-performing classifier) to provide better insights into the reliability and robustness of the study results [35]. The ROC curve can ensure that the classifier is not randomly predicting each class by measuring the true positive rate (sensitivity or recall) against the false positive rate (the complement of specificity). The closer the value of the area under the ROC curve (ROC AUC) to one the better the performance of the assessed classifier. The reader is advised to refer to (Fawcett, 2006) for more details about the ROC analysis and associated terms.

Table 2 Pseudo code for the temporal predictive model algorithm

Dataset Preparation	
1	Collect full dataset $\boldsymbol{S} \rightarrow (\boldsymbol{x_n}, \boldsymbol{y_n})$
2	Replace $\boldsymbol{D}$ in $\boldsymbol{x}$ with $\boldsymbol{DS} \rightarrow \boldsymbol{SU}$ (updated dataset)
3	Prepare $\boldsymbol{SU}$
4	Make $\boldsymbol{SU}$ temporal - Order and index using testing date feature $\boldsymbol{x_t}$
5	Split ($\boldsymbol{SU}$) $\rightarrow$ ($\boldsymbol{SU_{train}} = 90\%$, $\boldsymbol{SU_{test}} = 10\%$) - using stratified shuffle split
Training and Validation	
6	Define a classifier $\boldsymbol{clf}$
7	For $\boldsymbol{SU_{train}} \rightarrow (\boldsymbol{x_{train}}, \boldsymbol{y_{train}})$:
8	Define stratified K-fold $\rightarrow \boldsymbol{K} = 10$
9	For $\boldsymbol{K}$ in stratified 10 folds:
10	Split $\boldsymbol{SU_{train}} \rightarrow (\boldsymbol{SU_{tr}} = 9\ folds,\ \boldsymbol{SU_{te}} = 1\ fold\ )$
11	Train $\boldsymbol{clf}\ (x_{tr}, y_{tr}) \rightarrow \boldsymbol{clf_{fitted}}$
12	Predict and validate $\boldsymbol{clf_{fitted}}(x_{te}, y_{te})$
13	Return $\boldsymbol{PM_{va}}$ – validation performance measures
Testing	
14	For $\boldsymbol{SU_{test}} \rightarrow (\boldsymbol{x_{test}}, \boldsymbol{y_{test}})$:
15	Predict $\boldsymbol{clf_{fitted}}\ (\boldsymbol{x_{test}}) \rightarrow\ \hat{\boldsymbol{y}}_{\boldsymbol{test}}$
16	Calculate six performance measures ($\hat{\boldsymbol{y}}_{\boldsymbol{test}}, \boldsymbol{y_{test}}$)
17	Return $\boldsymbol{PM_{test}}$ – testing performance measures

4 Results and discussion

4.1 Phase 1: multi-criteria text mining model

This section discusses the findings of the first phase of the study: building the multi-criteria text mining model for COVID-19 testing reasons and symptoms. As discussed in the data and methods section, the development of the model was done in three sub-phases as follows: (1) initial preparation and exploration of the corpus, (2) deep wrangling of the corpus using text mining, and (3) text tagging and categorization.

In the first sub-phase, missing data were replaced with a new category called 'Unreported.' Also, obvious errors and issues in the data and text were corrected and validated with the relevant stakeholders. For example, the entire text corpus was converted to lower case. Another instance is that numbers and special characters such as exclamation and question marks were removed. Also, misspellings and grammatical mistakes were checked and corrected. Additionally, all the COVD-19-related terms such as corona and covid were standardized to one-term COVID in the corpus. Another example is correcting commonly misspelled words such as 'fatigue,' 'asymptomatic,' 'sore throat,' 'diarrhea,' and 'dyspnea.'

In the second sub-phase, the unstructured corpus was gradually deconstructed and converted to a dataset ready for categorization and tagging. As explained in the data and methods section, the text deconstruction and conversion were carried out in three main steps: tokenization, lemmatization, and removal of English stopwords. The corpus was deeply explored and wrangled after each deconstruction step in this phase to make sure that the corpus was properly prepared for tagging. For example, some acronyms and slang words were identified and amended based on the exploration of the corpus. Another instance is that less common and frequently misspelled words, such as 'malaise' and 'tightness,' were detected and corrected during the exploration and wrangling.

In the last (third) sub-phase, the testing reasons and symptoms texts were categorized and tagged using look-up wordlists. The prepared text corpus was analyzed using the frequency distribution calculation. The frequency distribution calculates the most frequent words in the corpus. Figure 3 shows the top 25 common and frequent words in the text corpus before (a) and after (b) cleaning the corpus. It can be seen in Fig. 3a and b that the word 'asymptomatic' is the most common. In addition, the frequency and order changed before and after cleaning. For example, the frequency of the word 'asymptomatic' increased after

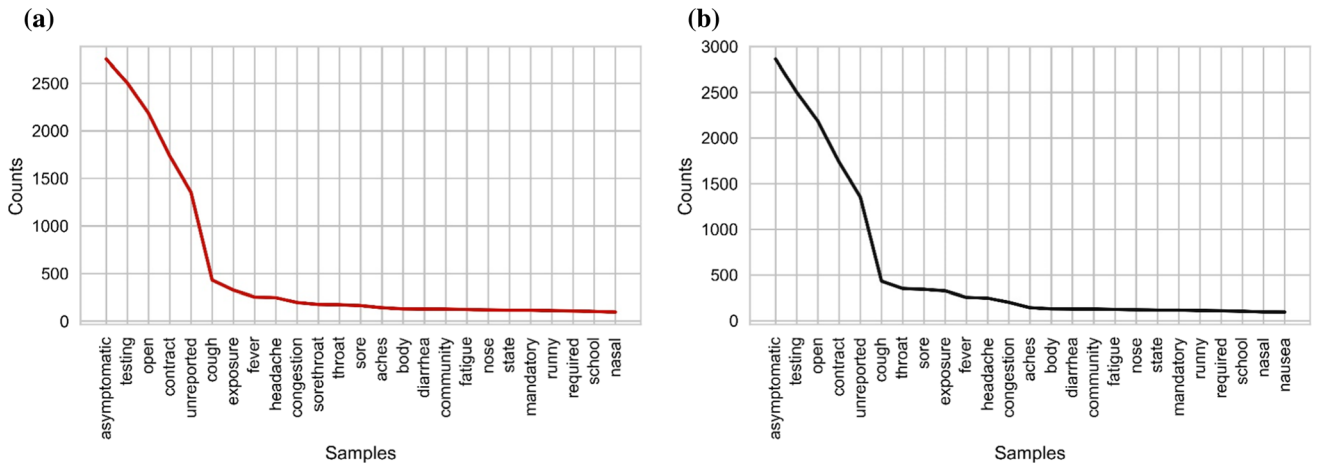

Fig. 3 Most frequent and common words (top 25) in the text corpus. **a** before cleaning the corpus; **b** after cleaning the corpus

cleaning the corpus. This step, along with the expertise of the clinical staff, was critical to identify the categories and wordlists for testing reasons and symptoms in the next steps.

Nine categories were identified for the testing reasons and symptoms according to the findings of the frequency distribution analysis and discussion with the clinical team. Each category is linked to a look-up wordlist that is used to tag each testing reason and symptom in the corpus. Table 3 presents the main look-up words in the wordlists of the nine categories. The complete wordlists were not included due to space limitations and privacy concerns.

The 'asymptomatic' category represents patients with no symptoms. The 'constitutional' category represents patients with general fever, body, and flu-like symptoms. The 'contract' category represents patients who test frequently based on a contract such as nursing homes. The 'gastro' category represents patients with gastrointestinal symptoms such as vomiting, nausea, diarrhea, stomach aches, and abdominal pain. The 'neurology' category represents patients with symptoms related to the nervous system such as headache, dizziness, and loss of smell and taste. The 'other/unreported' category represents patients who either did not report their symptoms, experienced uncommon symptoms, such as rash and mental issues, or were referred by other healthcare providers. The 'PHC' category represents patients who were exposed or had contact with other patients with COVID-19. The 'required by the state, county, or work' category represents patients who are mandated to be tested to do a certain job. The 'respiratory and ENT' category represents patients with respiratory, ear, nose, and throat symptoms such as shortness of breath, sinus, sore throat, and congestion.

The last step in the third sub-phase is tagging each testing appointment to a category or multiple categories by mapping each text to the categories using the wordlists. Each text can be assigned to more than one category because testing appointments can be scheduled for several reasons or patients can experience multiple symptoms. Figure 4 illustrates the total frequency of each reason category in the dataset where it measures how many times each category appeared in the dataset. Figure 5 shows the total frequency of the top 10 category combinations where it measures how many times each category occurred alone or with other categories (itemsets frequency). In Fig. 4, the reasons and symptoms were counted individually. Figure 5

Table 3 Main look-up words in the wordlists of the nine testing reasons and symptoms categories

#	Testing Reason Category	Main Look-up Words
1	Asymptomatic	Asymptomatic
2	Constitutional	Body, aches, fever, fatigue, flu, weakness
3	Contract	Contract
4	Gastro	Nausea, vomiting, diarrhea, stomach, abdominal
5	Neurology	Headache, smell, taste, dizziness
6	Other/Unreported/Referral	Unreported, referral, mental, allergy, rash
7	Public Health Contact (PHC)	Exposure, contact
8	Required by state, county, or work	Mandatory, state, county, work, requirement
9	Respiratory/ENT	Cough, congestion, sinus, breathing, lungs, throat

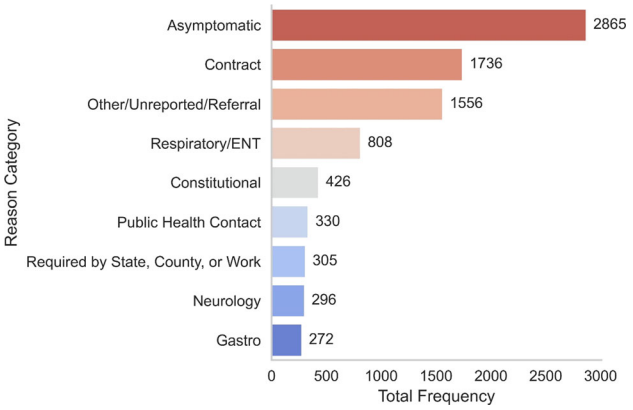

Fig. 4 Total frequency of each testing reasons and symptoms category

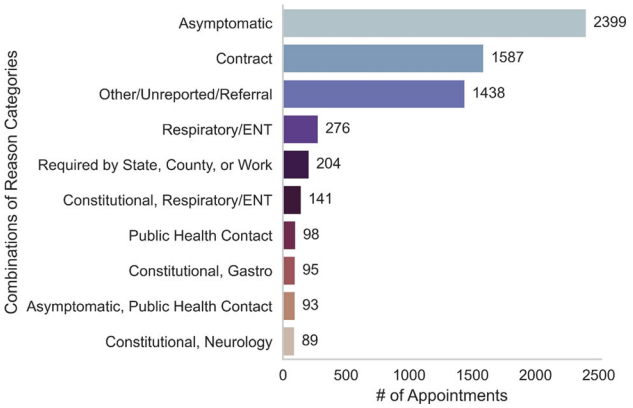

Fig. 5 Total frequency of the top 10 testing reasons combinations

is used to show what groups happen together more frequently as some testing reasons and symptoms have some association and correlation. In Fig. 5, the reasons and symptoms were counted in itemsets like in the 'Association Rule mining' method to check if there are any frequent itemsets or combinations of reasons and symptoms. The main difference between the two figures is that Fig. 5 is developed to show frequency of itemsets that considers the relationship and association between the groups of testing reasons and symptoms.

As it can be noticed from Fig. 4, most patients experienced no symptoms at all before testing. Besides, most patients who experienced symptoms suffered from respiratory and ENT issues more than any other type of symptoms. This means that patients with respiratory and ENT issues think they have COVID-19, but it does not mean that they have it. Figure 5 shows that the constitutional symptoms happen frequently with the ENT/respiratory symptoms where you cannot derive this insight from Fig. 4 because Fig. 4 shows the frequency of each category without considering the association with other groups. Besides, Fig. 5 shows that many patients, who suffer from symptoms before testing, are more likely to have ENT/respiratory symptoms only. It should be kept in mind that Figs. 4 and 5 do not imply that patients have COVID-19. These figures show only the reasons why patients want to be tested.

Figure 6 shows the positivity and negativity rates for each testing reason and symptom category. The patients who suffered from neurology and constitutional symptoms are the most susceptible to COVID-19. Also, patients with respiratory and ENT symptoms combined with constitutional symptoms have a high positivity rate. Patients who were contacted by the public health authorities because they were exposed to other COVID-19 patients have a high possibility to be positive for the virus. Patients who were mandated (by contract, state, work, etc.) to get the test are less likely to have the virus, which may mean that they are very careful about being exposed to the virus.

Figure 7 shows the average age (years) of the top 10 combinations of testing reasons and symptoms categories. It can be seen from the figure that age varies greatly with the categories. This may indicate an association between the variables.

Figure 8 shows the frequency of the top 10 combinations of testing reasons and symptoms categories by ethnicity (a) and gender (b). It can be noted from Fig. 8a that ethnicity does not largely differ among the categories. It can be noted from Fig. 8b that the female gender dominates all the categories except for the patients who were contacted by public health authorities, which may indicate that male patients are most likely to be exposed to COVID-19 patients (e.g., potentially less careful about the social distancing guidelines). The results are representative in the rural populations in Upstate NY.

4.2 Phase 2: classification predictive model

This section discusses the results and findings of the second phase of the study: building the classification predictive model of the COVID-19 testing results. As discussed in the data and methods section, the development of the model is carried out in several steps: data preparation and splitting, classifiers tuning, training and validation, and model testing.

First, the dataset was cleaned and prepared to build the classification model. The missing data in the categorical features were replaced with a new class called 'Unreported.' There were no missing data in the numerical features. Data entry error mistakes, such as errors found in the 'Race' field (e.g., some patients entered their race as Hispanic although it is considered ethnicity), were also corrected. The date features (date of testing and results) were converted to be numerical features using the Pandas package available in Python. The categorical features were

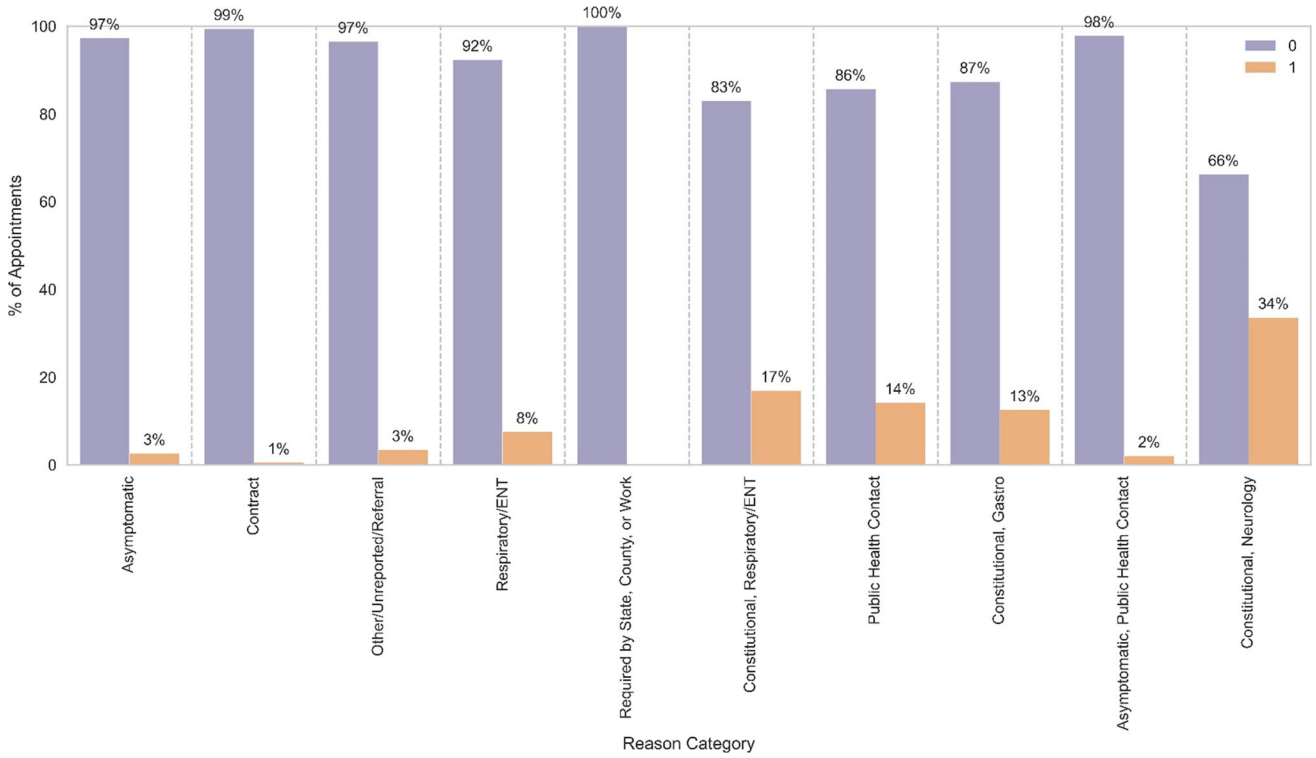

Fig. 6 Positivity and negativity rates for the top 10 combinations testing reasons and symptoms categories

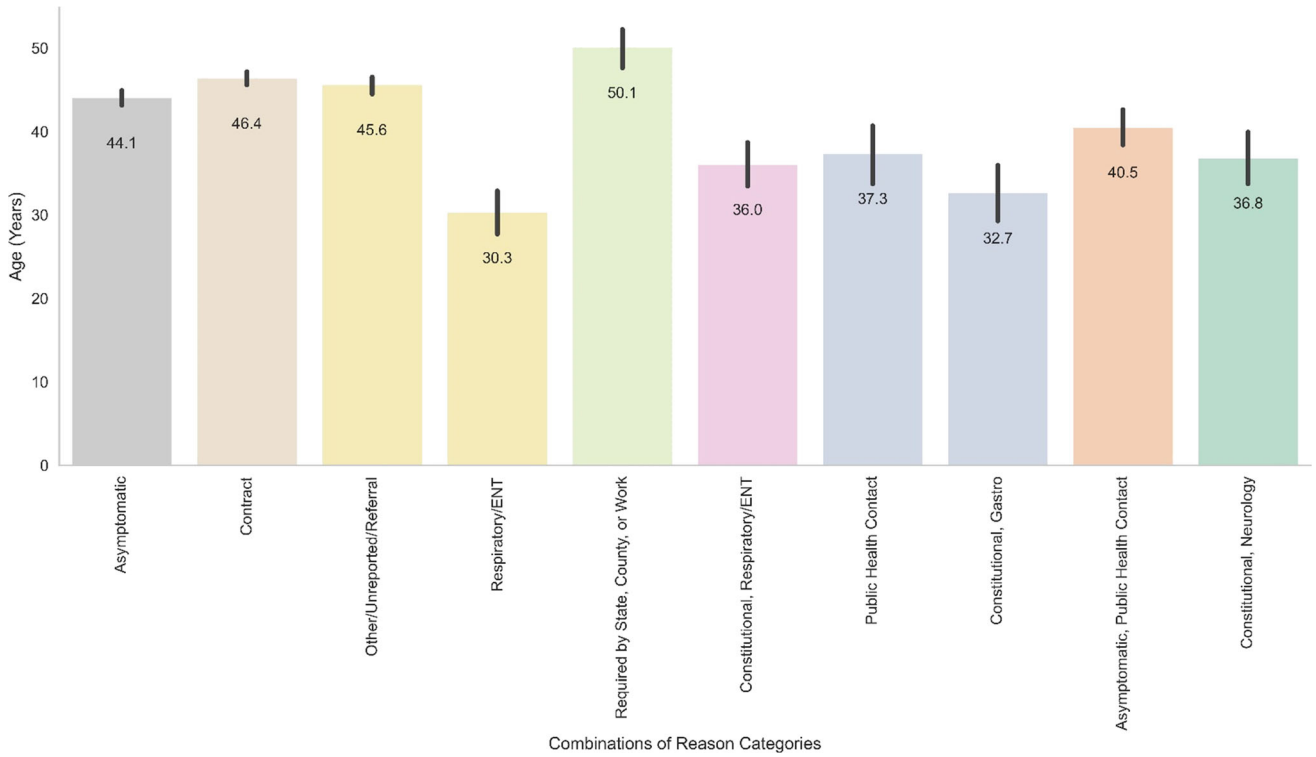

Fig. 7 Average age (years) of the top 10 combinations of testing reasons and symptoms categories

converted and encoded to numerical variables using the 'Target Encoding' method [21, 36]. The encoder works by using a blend of the posterior probability of the target label given a specific categorical class and the prior probability

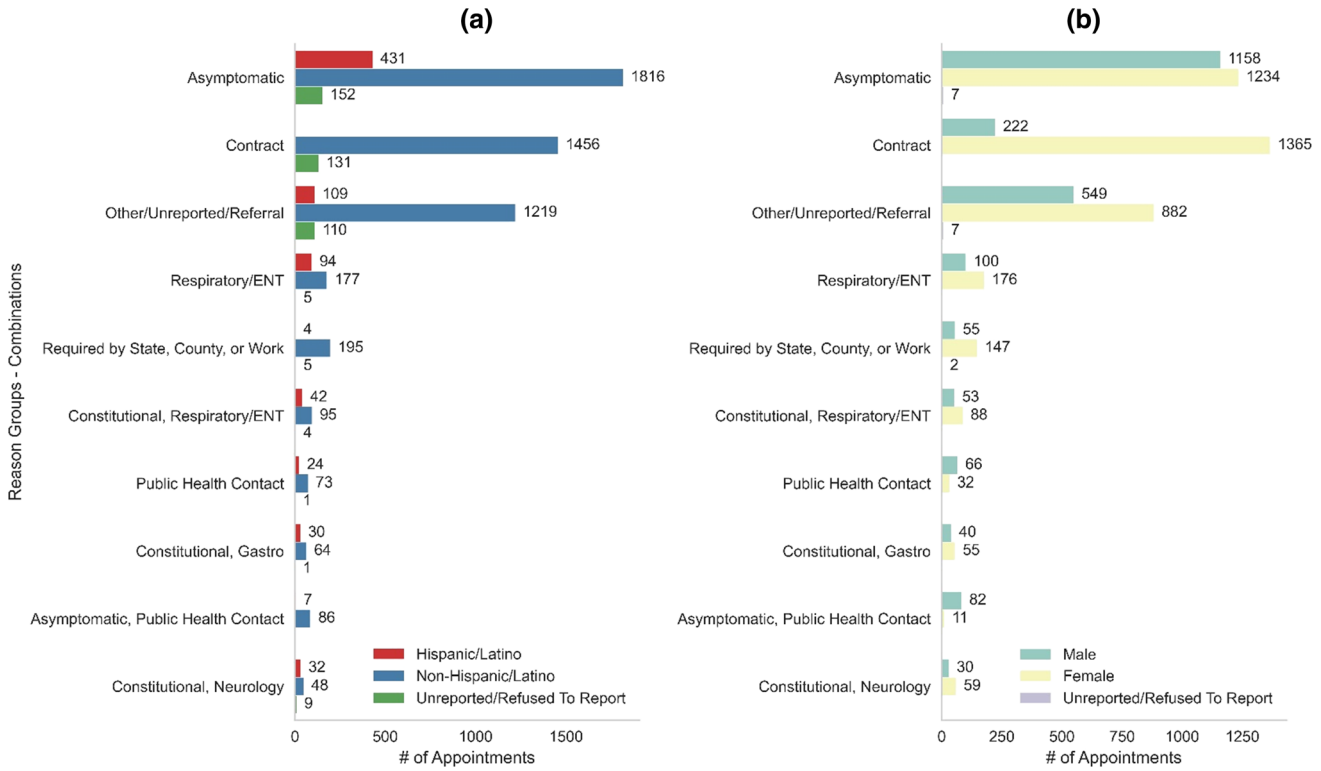

Fig. 8 Frequency of the top 10 combinations of testing reasons and symptoms categories by **a** ethnicity and **b** gender

of the target label over all the data. The numerical features were checked for outliers; however, there were no outliers in the numerical features such as the age field. The data were then ordered using the testing date and the week number to make it temporal. Dataset splitting for training, validation, and testing is discussed earlier in the data and methods section.

Second, the classifiers were tuned, trained, and validated. As discussed earlier, three balanced or weighted classifiers (RF, BDT, GP) and three regular classifiers (K-NN, LDA, and GP) were used in building the model. After tuning the hyperparameter of each classifier, the parameters were used to build and train the classifier to be validated and tested. The classifiers were first validated using the Stratified K-fold method and then tested using the holdout testing set. Table 4 summarizes the six performance measures result for each classifier used using the validation and testing sets. (Lowest numbers are highlighted in red.) As shown, the weighted and balanced classifiers outperformed regular classifiers using the balanced and geometry performance measures (balanced accuracy and G-score). The regular classifier performed poorly using the balanced and geometry measures. On the other hand, the regular classifier outperformed the weighted and balanced classifiers using the remaining performance measures. This means that the regular classifier can detect only the majority class without the ability to see the minority class effectively. The weighted and balanced classifiers can detect both classes effectively and accurately. Another thing that can be noticed is that all the classifiers performed with higher consistency using the testing set (lower STD) than the validation set. However, the performance measures are slightly higher using the validation set than the testing set.

For the balanced or weighted classifiers, all three classifiers showed satisfactory stable performance using the validation and testing sets. The standard deviation of the classifiers is less than 4%, which indicates consistent and stable performance. BDT classifier has the best overall performance among all balanced or weighted classifiers, as discussed earlier. The GB classifier has the worst performance using the balanced and geometry performance measures, which means the classifier is less effective in classifying the minority class. On the other hand, the RF classifier has the best balanced and geometry performance measures using the validation set (85.8 and 85.8%, respectively), and the BDT classifier has the best balanced and geometry performance measures using the testing set (83.0 and 82.8%, respectively). The RF classifier achieved a balanced accuracy 13.9% higher than the best standard classifier (LDA) using the testing set. Also, the RF classifier achieved a G-score 18.5% higher than the best standard classifier (LDA) using the validation set. The BDT classifier achieved a balanced accuracy 9.1% higher than the best standard classifier (K-NN) using the validation set.

Table 4 Performance measures result for each classifier. STD stands for standard deviation

		Balanced & Weighted Classifiers			Standard Classifiers		
	Measure Value (STD)	RF	BDT	GB	K-NN	LDA	GP
Validation	Accuracy	86.8% (1.0%)	89.2% (1.6%)	94.7% (0.8%)	96.7% (0.6%)	96.6% (0.5%)	95.0% (0.7%)
	Balanced Accuracy	86.1% (2.3%)	85.8% (2.4%)	77.2% (3.1%)	70.8% (3.9%)	71.9% (3.7%)	63.5% (4.7%)
	F1-Score	90.2% (0.6%)	91.7% (0.8%)	95.1% (0.6%)	95.5% (0.6%)	96.4% (0.6%)	94.6% (0.8%)
	Recall	86.8% (0.3%)	89.1% (1.6%)	94.7% (0.8%)	95.7% (0.6%)	96.6% (0.5%)	95.0% (0.7%)
	Precision	96.0% (0.3%)	96.0% (0.2%)	95.6% (0.5%)	95.4% (0.6%)	96.3% (0.5%)	94.3% (0.9%)
	G-Score	86.1% (2.4%)	85.8% (2.4%)	75.1% (3.9%)	66.1% (5.4%)	67.3% (5.1%)	54.7% (8.3%)
Testing	Accuracy	86.8% (0.4%)	89.3% (0.6%)	94.7% (0.8%)	96.4% (0.3%)	96.3% (0.0%)	96.1% (0.3%)
	Balanced Accuracy	82.1% (0.2%)	83.0% (0.7%)	75.0% (2.4%)	73.9% (1.7%)	68.7% (0.0%)	65.8% (2.7%)
	F1-Score	90.8% (0.3%)	91.7% (0.4%)	95.0% (0.6%)	96.2% (0.3%)	95.9% (0.0%)	95.5% (0.4%)
	Recall	86.8% (0.4%)	89.3% (0.6%)	94.7% (0.8%)	96.4% (0.3%)	96.3% (0.0%)	96.1% (0.3%)
	Precision	95.4% (0.0%)	95.6% (0.1%)	95.4% (0.4%)	96.1% (0.3%)	95.7% (0.0%)	95.3% (0.5%)
	G-Score	81.9% (0.2%)	82.8% (0.8%)	72.3% (3.1%)	70.3% (2.3%)	62.9% (0.0%)	58.4% (4.2%)

Also, the BDT classifier achieved a G-score 13.1% higher than the best standard classifier (K-NN) using the testing set. Overall, RF and BDT have a great ability to accurately detect the minority class.

For the standard classifier, they showed satisfactory and stable performance using the regular measures and performed poorly using the balanced and weighted measures. The GP classifier had the worst measures using both the validation and testing sets among all the regular classifiers. K-NN and LDA classifiers have almost similar performance using both sets and all measures. As aforementioned in the data and method section, the final model evaluation is based on the balanced and geometry measures because they can show us the ability of the classifiers to accurately detect the minority class. Those measures can reduce the risk of misclassifying positive results as negative, which means patients with COVID-19 are effectively detected. Accordingly, the RF and BDT are chosen as the best performing classifiers and, consequently, are now being used in the real-time model at the medical center (FLCH).

Figure 9 shows the ROC curve for the BRF classifier. It can be noticed that the performance of BRF is highly satisfactory. The BRF classifier can detect both classes with high accuracy. The average, micro-, and macro-AUC values for both classes are higher than 92%, which indicates that the classifier can accurately predict both classes (perfect diagnostic ability). Figure 10 shows the probability distribution of the predicted values in the testing set using the BRF classifier. It can be seen that a large proportion of the probabilities of the predicted values are closer to one, which indicates that the classifier has high confidence in predicting the cases in the testing set. There are few cases where the probability is almost equal for both classes, which indicates that the classifier is unsure about them (gray area).

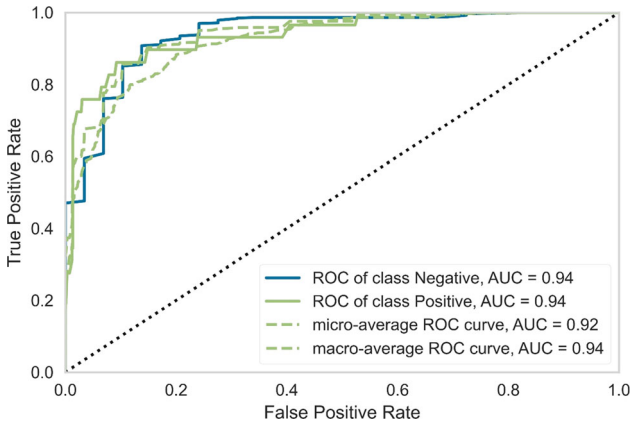

Fig. 9 The ROC curve for the BRF classifier

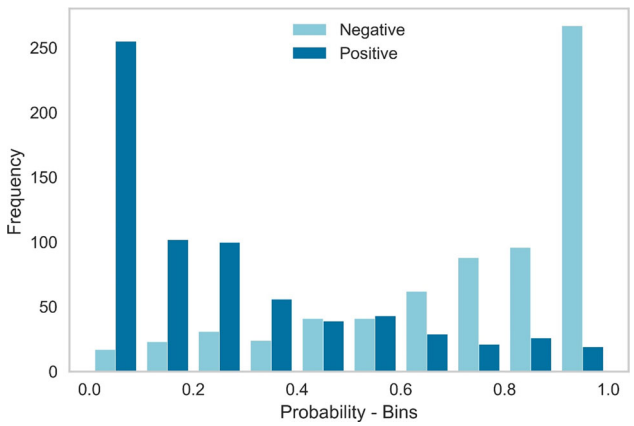

Fig. 10 Predictions probability distribution for each class in the testing set using the BRF classifier

Figure 11 shows the importance of each feature in the dataset relative to the target label (testing results) using the BRF classifier. BRF was used because it performed well with great stability and it has the feature importance characteristic [37]. It can be seen from the figure that the most important features for the detection and classification of the testing results are patient history, age, and testing reasons and symptoms (higher than 10% relative importance compared to other features). Also, it is worth mentioning that the date features can influence the prediction of the testing results, which may indicate that there is some association or correlation between the time and testing results. Besides, the laboratory feature can be considered an important feature to the BRF classifier.

5 Conclusions and future work

The COVID-19 pandemic has put many lives in danger and affected the world negatively in many ways. There is indeed a need to fight this pandemic by all means and in all areas. Most of the relevant studies are focused on predicting the key pandemic trends such as the number of confirmed cases, the mortality and hospitalization rates, and the number of recoveries. This study is conducted to build a multi-criteria text mining model for COVID-19 testing reasons and symptoms integrated with a temporal predictive classification model for COVID-19 test results in rural underserved areas. A dataset of 6,895 testing appointments and 14 features is used in this study.

The text mining model classifies the notes related to the testing reasons and reported symptoms into one or more categories using look-up wordlists and a multi-criteria mapping process. The model converts an unstructured feature to a categorical feature that is used in building the temporal predictive classification model for COVID-19 test results and conducting some population analytics. The classification model is a temporal model (ordered and indexed by testing date) that uses machine learning classifiers to predict test results either positive or negative. Two types of classifiers are used, which include balanced classifiers (random forest, bagged decision tree, and gradient boosting) and regular classifiers (K-nearest neighbor, linear discriminant analysis, and Gaussian process). The balanced or weighted methods are used to address and account for the biased dataset and imbalanced target label and to ensure correct detection of patients with COVID-19 (minority class). The model is tested in a two-stage manner using validation and testing sets to ensure robustness and reliability.

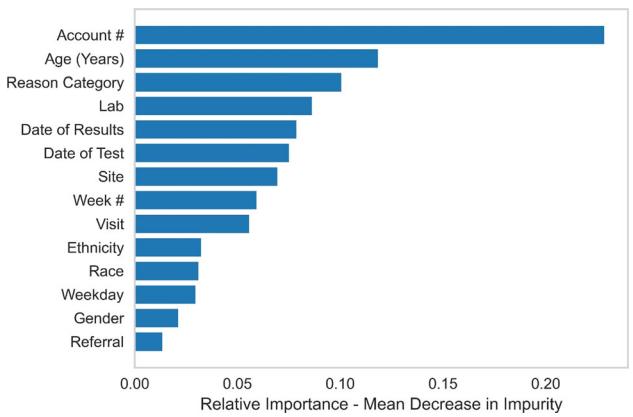

Fig. 11 Feature Importance of the 14 features using the BRF classifier

All classifiers showed stable consistent performance with low standard deviation: Most of the measures' standard deviations were less than 5%. Nevertheless, the balanced classifiers outperformed regular classifiers using the balanced performance measures (Balanced accuracy and G-score), which means that the balanced classifiers are better at detecting patients with positive COVID-19 results. The balanced random forest achieved the best average balanced accuracy (86.1%) and G-score (86.1%) using the validation set. The balanced bagged decision tree achieved the best average balanced accuracy (83.0%) and G-score (82.8%) using the testing set. Also, it was found that patient history, age, testing reasons, and time are the most important features to predict the testing results.

The findings of this research are promising for outpatient primary care providers working with underserved communities. One major limitation of this study is the reliability of the used data. Some of the patients reported inaccurate or incomplete data. For future work, other machine learning algorithms such as convolutional deep learning neural networks will be considered. Also, the look-up tables will be continuously updated for accurate text tagging. Finally, the model will be real-time, and it will be connected to the EHR system.

Funding This research was not supported by any grant or funding agencies.

Declarations

Conflicts of interest There is no competing or conflict of interest associated with this study.

References

1. Velavan TP, Meyer CG (2020) The COVID-19 epidemic. Trop Med Int Health 25(3):278–280. https://doi.org/10.1111/tmi.13383
2. Fauci AS, Lane HC, Redfield RR (2020) Covid-19—navigating the uncharted. N Engl J Med 382:1268–1269. https://doi.org/10.1056/NEJMe2002387

3. Shereen MA, Khan S, Kazmi A, Bashir N, Siddique R (2020) COVID-19 infection: emergence, transmission, and characteristics of human coronaviruses. J Adv Res 24:91–98. https://doi.org/10.1016/j.jare.2020.03.005
4. Kraemer MU, Yang CH, Gutierrez B, Wu CH, Klein B, Pigott DM, Du Plessis L, Faria NR, Li R, Hanage WP, Brownstein JS (2020) The effect of human mobility and control measures on the COVID-19 epidemic in China. Sci 368(6490):493–497. https://doi.org/10.1126/science.abb4218
5. Gondauri D, Batiashvili M (2020) The study of the effects of mobility trends on the statistical models of the COVID-19 virus spreading. Electron J General Med 17(6):1–4
6. Santosh KC (2020) COVID-19 prediction models and unexploited data. J Med Syst 44:170. https://doi.org/10.1007/s10916-020-01645-z
7. Ahmad A, Garhwal S, Ray SK, Kumar G, Malebary SJ, Barukab OM (2020) The number of confirmed cases of covid-19 by using machine learning: methods and challenges. Arch Computat Methods Eng 28(4):2645–2653. https://doi.org/10.1007/s11831-020-09472-8
8. Wang P, Zheng X, Li J, Zhu B (2020) Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics. Chaos, Solitons Fractals 139:110058. https://doi.org/10.1016/j.chaos.2020.110058
9. Rustam F, Reshi AA, Mehmood A, Ullah S, On BW, Aslam W, Choi GS (2020) COVID-19 future forecasting using supervised machine learning models. IEEE Access 8:101489–101499. https://doi.org/10.1109/ACCESS.2020.2997311
10. Tuli S, Tuli S, Tuli R, Gill SS (2020) Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing. Internet of Things 11:100222. https://doi.org/10.1016/j.iot.2020.100222
11. Pinter G, Felde I, Mosavi A, Ghamisi P, Gloaguen R (2020) COVID-19 pandemic prediction for Hungary; a hybrid machine learning approach. Math 8(6):890. https://doi.org/10.3390/math8060890
12. CDC (2021) Things to Know about the COVID-19 Pandemic. CDC COVID-19 https://www.cdc.gov/coronavirus/2019-ncov/your-health/need-to-know.html#print. Accessed June 18, 2021
13. Neuman S (2020) CDC Adds 6 Symptoms To Its COVID-19 List. NPR Coronavirus Updates. https://www.npr.org/sections/coronavirus-live-updates/2020/04/27/845321155/cdc-adds-6-symptoms-to-its-covid-19-list. Accessed June 18, 2021
14. Krittanawong C, Narasimhan B, Virk HU, Narasimhan H, Wang Z, Tang WW (2020) Insights from Twitter about novel COVID-19 symptoms. Eur Heart J–Dig Health 1(1):4–5. https://doi.org/10.1093/ehjdh/ztaa003
15. Sarker A, Lakamana S, Hogg-Bremer W, Xie A, Al-Garadi MA, Yang YC (2020) Self-reported COVID-19 symptoms on Twitter: an analysis and a research resource. J Am Med Inform Assoc 27(8):1310–1315. https://doi.org/10.1093/jamia/ocaa116
16. Saire JE, Navarro RC (2020) What is the people posting about symptoms related to Coronavirus in Bogota, Colombia?. arXiv preprint
17. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 2:2825–2830
18. Lemaître G, Nogueira F, Aridas CK (2017) Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. J Mach Learn Res 18(1):1–5
19. Bird S, Klein E, Loper E (2009) Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.
20. McKinney W (2010) Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference 445:51–56
21. Oliphant TE (2006) A guide to NumPy. Trelgol Publishing, USA
22. W. McGinnis (2016) Category Encoders. https://contrib.scikit-learn.org/category_encoders/. Accessed June 18, 2021
23. Waskom M, Botvinnik O, Ostblom J, Gelbart M, Lukauskas S, Hobson P, Gemperline DC, Augspurger T, Halchenko Y, Cole JB, Warmenhoven J (2020) mwaskom/seaborn: v0. 10.1. Zenodo. 10.5281/zenodo.3767070
24. Bengfort B, Bilbro R (2019) Yellowbrick: visualizing the Scikit-Learn model selection process. J Open Sour Softw 4(35):1075
25. Hunter JD (2007) Matplotlib: A 2D graphics environment. IEEE Ann Hist Comput 9(3):90–95. https://doi.org/10.1109/MCSE.2007.55
26. Spolaôr N, Cherman EA, Monard MC, Lee HD (2013) A comparison of multi-label feature selection methods using the problem transformation approach. Electron Notes Theor Comp Sci 292:135–151. https://doi.org/10.1016/j.entcs.2013.02.010
27. Szymański P, Kajdanowicz T (2017) A scikit-based Python environment for performing multi-label classification. arXiv preprint
28. Raschka S (2018) Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint
29. Twomey JM, Smith AE (1998) Bias and variance of validation methods for function approximation neural networks under conditions of sparse data. IEEE Trans Syst, Man, Cybern Part C (Appl Rev) 28(3):417–430
30. Ojala M, Garriga GC (2010) Permutation tests for studying classifier performance. J Mach Learn Res 11(6):1833–1863
31. Abu Lekham L, Wang Y, Hey E, Lam SS, Khasawneh MT (2020) A multi-stage predictive model for missed appointments at outpatient primary care settings serving rural areas. IISE Trans Healthcare Syst Eng 1:79–94. https://doi.org/10.1080/24725579.2020.1858210
32. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(2):281–305
33. Powers DM (2020) Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint
34. Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. 2010 20th IEEE International Conference on Pattern Recognition 3121–3124
35. Kubat M, Matwin S (1997) Addressing the curse of imbalanced training sets: one-sided selection. ICML 97:179–186
36. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27(8):861–874. https://doi.org/10.1016/j.patrec.2005.10.010
37. Micci-Barreca D (2001) A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. ACM SIGKDD Explor Newsl 3(1):27–32. https://doi.org/10.1145/507533.507538
38. Scikit-yb (2019) Feature Importances. https://www.scikit-yb.org/en/latest/api/model_selection/importances.html. Accessed June 18, 2021