DOI: http://dx.doi.org/10.26483/ijarcs.v13i3.6838

ISSN No. 0976-5697

Volume 13, No. 3, May-June 2022

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

SA: SIMULATED ANNEALING FOR ANALYZING COVID-19 FEMALE RECOVERY PATIENTS MENTAL HEALTH USING DEMOGRAPHIC FACTORS

Dr. Subhani Shaik[1], Dr. K. Vijayalakshmi[2], K Gowri Priya[3], V. Vismitha[4], J. Saiteja[5]
Associate Professor, Dept. of IT, Sreenidhi Institute of science & Technology(A), Hyderabad.
Professor, Dept. of IT, Sreenidhi Institute of science & Technology(A), Hyderabad.
B.Tech Students, Dept. of IT, Sreenidhi Institute of science & Technology(A), Hyderabad.

Abstract: The COVID-19 epidemic has been causing chaos on the society, rendering it in ruins. WHO predicts that in March 2020, the world has taken a significant toll on people, leading to a lot of distortion in their lifestyle? Mental health has gone for a toss and overlooked. People in India do not have the privilege to expect support and deal with their mental health. It is high time to bridge this gap, and we have attempted to do using unconventional approaches like the booming technologies. The proposed model is validated against other baseline techniques like naive-bayes, gradient-boosting, xgboost, catboost, lightgbm and optimization techniques like simulated annealing using svm. The proposed method outperforms other baseline techniques for attaining better accuracy. They are particularly suited to predicting psychological problems. For implementation purposes, choose features like age, family history, seek_help, employment, and a few other features. The proposed model is evaluated with a COVID-19 dataset based on various performance matrices to show its effectiveness.

Keywords: pandemic, classification, model, age, psychological, health, consequences, optimization, techniques, conventional approaches.

INTRODUCTION

Nowadays, many people are suffering from Covid-19 and this has been key issue in the society. According to the World Health Organization (WHO), for many people with insaness there is decrease in their life expectancy. In many countries and also in India introduced many policies like NMHP 2014 to deal with mental health issues, But more needs to be done in the context of COVID-19, which has exacerbated mental illnesses everywhere. In this project we are going to analyze Covid-19 female recovery patient's mental health using demographic factors using features like age, ancestry, seek help, employment, educational achievement, marital status, and a few other features. Doing so we can conclude that whether a covid recovered person is mentally stressed or not.

The purpose of our research is just to explore the effectiveness of COVID-19 on psychiatric problems in terms of their medical condition and treatment adherence as reported by their main caretakers, in addition to investigate the relationship between lifestyle (years old, education levels, socioeconomic factors, support systems) and personality characteristics (diagnoses of mental problems, sickness status, and patient's insight of the COVID-19 epidemic are all factors to consider). Contrasting the normal community and various mental conditions, this population's awareness level may be modified by their careers, who play a critical role in customer satisfaction, recovery, and early relapse detection. The presumption is that those with psychiatric illness have no non-physical influence from COVID-19, and their COVID-19 consciousness will be the same as the general population. Demographic and clinical characteristics will have no effect on their level of awareness.

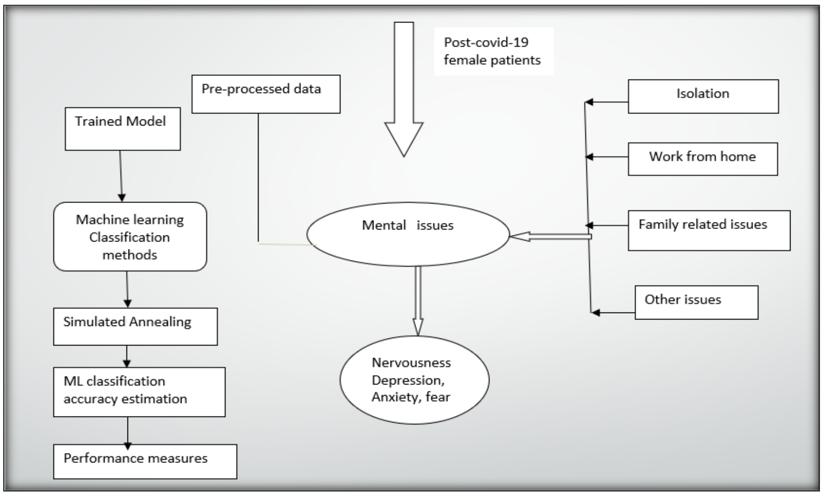

Figure 1: Framework

LITERATURESURVEY

This thesis documents the existing models and their limitations that would bring the revolution in understanding of our project proposal. Ideally, the limitations of the proposed works are taken into consideration which helps in accuracy and precision of the evaluation of the statements. This would help us in taking up various algorithms and procedural methods for data collection and their pre processing. Eight publications, including commentaries (n = 4) and correspondence (n = 5) addressed the potential mental health impact of COVID-19 on the general population, based on literature from previous disease outbreaks or specified theoretical models. There was greater geographical diversity in this group of publications, with papers originating from China, Canada, Iran, Japan, Singapore and Brazil.

Two of these papers examined the likely impact of the COVID-19 pandemic in specific countries. One of these, from Iran (Zandifar and Badrfam, 2020) highlighted the role of unpredictability, uncertainty, seriousness of the disease, misinformation and social isolation in contributing to stress and mental morbidity. The authors highlighted the need for both mental health services, particularly for vulnerable populations, and the strengthening of social capital to reduce the adverse psychological impact of the outbreak. Another, from Japan (Shigemura et al., 2020), emphasised the economic impact of COVID-19 and its effects on well-being, as well as the likely high levels of fear and panic behaviour, such as hoarding and stockpiling of resources, in the general population. This paper also identified populations at higher risk of adverse mental health outcomes, including patients with COVID-19 and their families, individuals with existing physical or psychiatric morbidity, and healthcare workers.

Of the remaining papers, one pointed out that the wide scope and spread of COVID-19 could lead to a true mental health crisis, especially in countries with high case loads (Dong and Bouey, 2020) which would require both large-scale psychosocial crisis interventions and the incorporation of mental health care in disaster management plans in the future. In a related report (Duan and Zhu, 2020) it was pointed out that while Western countries have incorporated psychological interventions into their protocols for disease outbreaks, this has not yet happened in countries such as China, leading to the emergence and persistence of stress-related disorders in affected persons.

Such methods, in their opinion, would empower society to handle the COVID-19 outbreak in an adaptive manner. Similar strategies were reiterated in a paper from Singapore (Ho et al., 2020) which also discussed the role of improved screening for mental disorders, improving links between community and hospital services, and providing accurate information to the general public in order to minimize maladaptive responses such as “panic” and paranoia regarding the disease and its transmission. In contrast to the above literature on practical considerations, two papers from Canada (Amundsen and Taylor, 2020a, b) have discussed the mental health impact of COVID-19 from the point of view of health anxiety. Health anxiety, which arises from the misinterpretation of perceived bodily sensations and changes, can be protective in everyday life. However, during an outbreak of infectious disease, particularly in the presence of inaccurate or exaggerated information from the media, health anxiety can become excessive. At an individual level, this can manifest as maladaptive behaviours (repeated medical consultations, avoiding health care even if genuinely ill, hoarding particular items); at a broader societal level, it can lead to mistrust of public authorities and scape goating of particular populations or groups. The authors underline the need for evidence-based research into health anxiety and its determinants, so that valid individual- and population-level strategies can be developed to minimize it in the face of the COVID-19 pandemic and future outbreaks of a similar nature.

FRAMEWORK

Reading the Dataset

The read csv method in pandas can be used to read a CSV file (). A CSV file can be read using pandas' read csv() function. The application reads the file Stress_Data-women.csv. We must use the to_csv() method of a DataFrame to start writing to a CSV file.

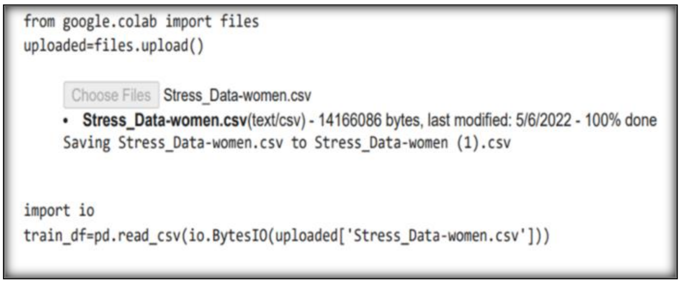

Figure 2 :Reading the dataset

Checking Null Values

- isnull() function to check null values.

In any expression, a NULL element will translate to NULL. If any column has variable, SQL will ignore UNIQUE, FOREIGN key, and CHECK restrictions. In generally, every NULL value in the database is thought to be distinct from the others. The conclusion of a comparison operation that includes a NULL is considered UNKNOWN.

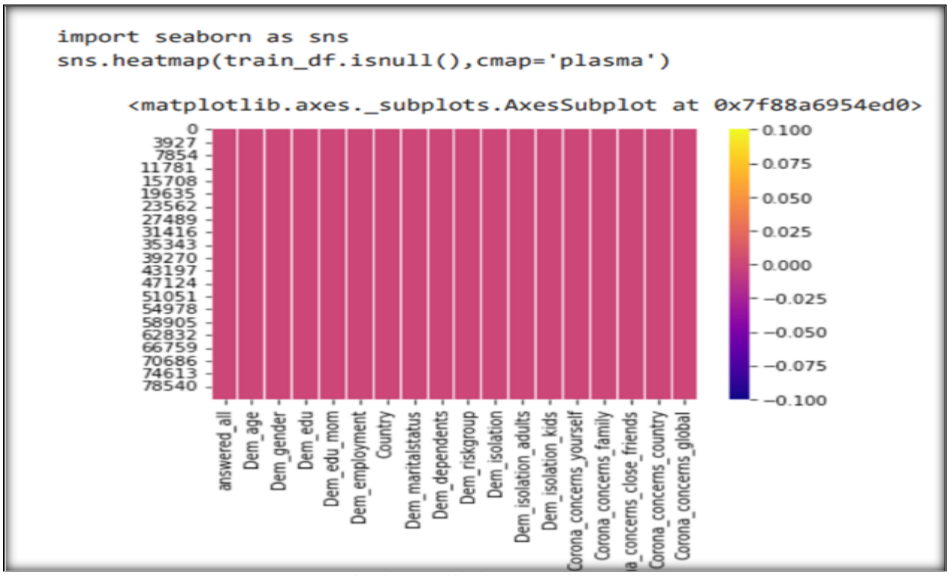

Figure 3 :Checking null values

Visualization

- We can get visualization using the seaborn library.
- Visualization is used in advanced analytics. It's important for the data scientist to show charts and graphs for proper understanding of problem and for implementation.
- The below Visualization graph shows the female ratio of covid-19 effected patients.

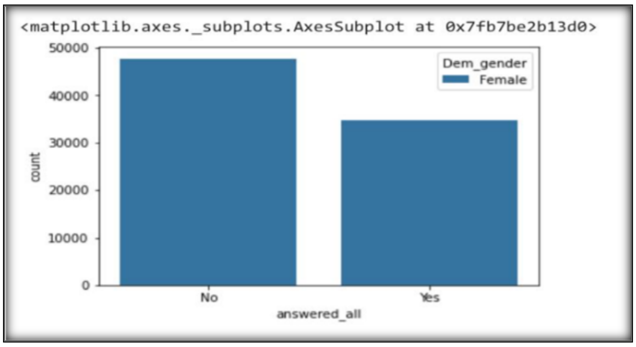

Figure 4: Gender Graph

The below graph indicates the relation of employment of covid -19 recovery patients with their mental health.

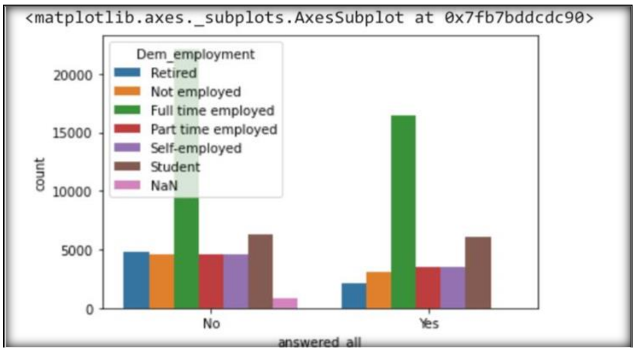

Figure 5 :Employment Graph

The below graph visualizes the effect of isolation state with their mental health.

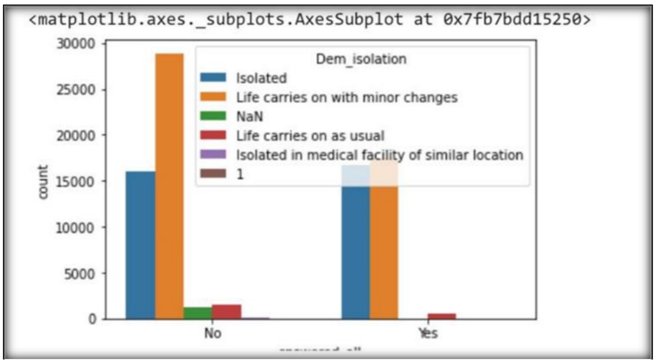

Figure 6: Isolation Graph

The below visualization indicates the effect of their family concern and their situation with their mental health.

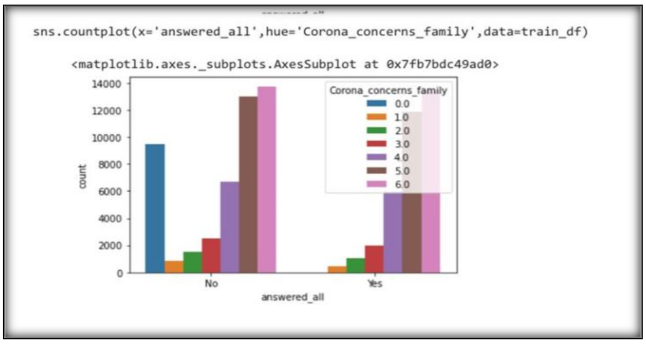

Figure 7:Family Concern Graph

Standardization

Standardization, where the readings are focused around the mean with just a unit standard deviation, is another scaling approach. This feature and the consequence are both 0.

Standardization ¶

```
scaler = MinMaxScaler()
train_df['Dem_age'] = scaler.fit_transform(train_df[['Dem_age']])
train_df.head()
```

	answered_all	Dem_age	Dem_gender	Den_edu	Dem_edu_mom	Dem_employment	Country
0	0	0.558442	0	4	6	4	6
1	1	0.259740	0	6	6	2	6
2	0	0.220779	0	0	0	0	108
3	0	0.311688	0	0	6	0	6
4	1	0.168831	0	0	0	0	6

...

```
y=train_df['answered_all'].values
y
```

```
array([0, 1, 1, ..., 0, 0, 0])
```

Figure 8: Standardization

SplittingTraining and Testing Dataset

- Sklearn is used to split the train and test datasets.
- Randn() method is used for splitting.
- It is a built-in pandas method.

Splitting training and testing dataset

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=0)
```

Figure 9: Splitting the dataset

RESULTS AND ANALYSIS

Naïve Bayes: The Naïve bayes Classifier is a simple and excellent classifying method for developing efficient machine learning techniques that can make quick predictions. It's a classification method, meaning it draws conclusions from the probability of an object.

1.Naive Bayes

```
import warnings
warnings.filterwarnings('ignore')

from sklearn.naive_bayes import GaussianNB
clf1 = GaussianNB()
clf1.fit(X_train, y_train)

      GaussianNB()

y_pred1=clf1.predict(X_test)
from sklearn.metrics import confusion_matrix,accuracy_score
print(X_test.shape)
cn1=confusion_matrix(y_test,y_pred1)
print(cn1)
accuracy_score(y_test,y_pred1)

      (24739, 17)
      [[5517 8753]
       [1585 8884]]
      0.5821173046606573
```

Figgure 10: Naïve Bayes Classifier

Gradientboosting: One sort of machine learning booster is GB. It depends on the assumption that when the best potential next model is coupled with prior models, the estimation error is minimized. To decrease error, the fundamental notion is to specify the desired results for the following model.

2.Gradient Boosting

```
from sklearn.ensemble import GradientBoostingClassifier
SEED = 1
clf2 = GradientBoostingClassifier(n_estimators = 200, max_depth = 1, random_state = SEED)
clf2.fit(X_train, y_train)

     GradientBoostingClassifier(max_depth=1, n_estimators=200, random_state=1)


y_pred2=clf2.predict(X_test)
print(y_pred2)
from sklearn.metrics import confusion_matrix,accuracy_score
cn=confusion_matrix(y_test,y_pred2)
print(cn)
accuracy_score(y_test,y_pred2)

     [0 1 0 ... 0 1 0]
     [[9855 4415]
      [3125 7344]]
     0.6952180767209669
```

Figure 11: Gradient Boosting

XGBoost: EGB, or Extreme gradient-boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning model. It provides parallel tree boosting and is the leading machine learning model for regression, classification, and ranking problems.

3.XG-Boost

```
import xgboost as xgb
from xgboost import XGBClassifier
clf3 = XGBClassifier(learning_rate=0.1, n_estimators=20, max_depth=5,min_child_weight=3, g

clf3.fit(X_train, y_train)

     XGBClassifier(colsample_bytree=1.0, gamma=0.2, max_depth=5, min_child_weight=3,
                   n_estimators=20, nthread=4, seed=27, subsample=0.6)


y_pred3=clf3.predict(X_test)
print(y_pred3)
from sklearn.metrics import confusion_matrix,accuracy_score
cn=confusion_matrix(y_test,y_pred3)
print(cn)
accuracy_score(y_test,y_pred3)


   [1 1 0 ... 0 1 0]
   [[9020 5250]
    [1735 8734]]
   0.7176522899066252
```

Figure 12: XG-Boost

CatBoost: **Yandex** recently released an open-source machine learning model. It can easily integrate with deep learning frameworks like Google's Tensor Flow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today.

```
import catboost as ctb
from catboost import CatBoostClassifier
clf4= CatBoostClassifier(loss_function='Logloss',eval_metric='AUC'
clf4.fit(X_train,y_train,verbose=False)

    <catboost.core.CatBoostClassifier at 0x7f88a656c690>

y_pred4=clf4.predict(X_test)
print(y_pred4)
from sklearn.metrics import confusion_matrix,accuracy_score
cn=confusion_matrix(y_test,y_pred4)
print(cn)
accuracy_score(y_test,y_pred4)

    [1 1 0 ... 0 1 0]
    [[9927 4343]
     [2164 8305]]
    0.7369740086503093
```

Figure 13: CatBoost

LightGbm: It is a decision tree-based gradient boosting model that can be used for rankings, classifications, and a number of other classification techniques.

```
5.LightGbm

import lightgbm as lgb
from lightgbm import LGBMClassifier
clf5 = LGBMClassifier()
clf5.fit(X_train, y_train)

    LGBMClassifier()

y_pred5=clf5.predict(X_test)
print(y_pred5)
from sklearn.metrics import confusion_matrix,accuracy_score
cn=confusion_matrix(y_test,y_pred5)
print(cn)
accuracy_score(y_test,y_pred5)

    [1 1 0 ... 0 1 0]
    [[9840 4430]
     [2042 8427]]
    0.7383887788512066
```

Figure 14: LightGbm

Simulated Annealing: It is a statistical model for obtaining a function global optimum**.** Specifically, it is a optimization technique to approximate global optimization in a large search space for an optimization problem. It is often used when the search space is discrete.

```
from sklearn import svm


svc_params = {'C':np.logspace(-8, 10, 19, base=2),
              'fit_intercept':[True, False]
             }
# Using a linear SVM classifier
clf = svm.LinearSVC()
# Initialize Simulated Annealing and fit
sa = SimulatedAnneal(clf, svc_params, T=10.0, T_min=0.001, alpha=0.75,
                         verbose=True, max_iter=1, n_trans=5, max_runtime=300,
                         cv=3, scoring='f1_macro', refit=True)
sa.fit(X_train, y_train)
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
cn=confusion_matrix(y_test,y_test_pred)
print(cn)
accuracy_score(y_test,y_test_pred)

    [[11486  2784]
     [ 6231  4238]]
    0.6355956182545778
```

Figure 15: Simulated Annealing-2

Equating Models Accuracy

- Naive bayes : 0.58
- Gradient boosting : 0.69
- Xgboost : 0.71
- Catboost: 0.73
- Light gbm: 0.738
- Simulated annealing : 0.63

```
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
```

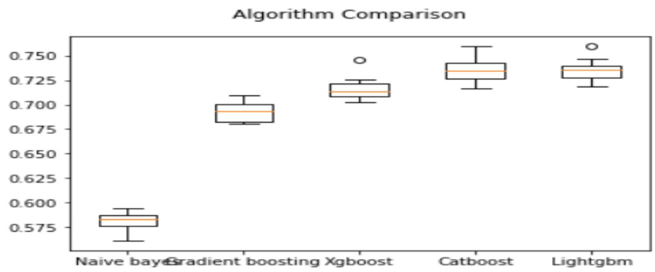

Figure 16: Comparing Accuracies

CONCLUSION

This project is mainly to predict "Covid 19 recovery female patients Mental Health using demographic factors". In this, we used different classification algorithms to get more accurate results. By observing the results, we can say that for this dataset LightGbm has given the more accurate results (0.738 accuracy) and CatBoost also predicted the output as almost the same accuracy (0.736) as the LightGbm predicted. And remaining algorithms like Naive bayes, Gradient Boost, XG-Boost gave 0.58, 0.69, 0.71 accuracy respectively. Simulated Annealing Optimization is also used in the analysis in order to verify most possible accuracy but 0.61 is the output, comparatively less than LightGbm.

REFERENCES

[1]. Aarts EHL, Korst JHM (1989) Simulated annealing and Boltzmann machines. Wiley, ChichesterzbMATHGoogle Scholar

[2]. Aarts EHL, Van Laarhoven PJM (1985) Statistical cooling: a general approach to combinatorial optimisation problems. Philips J Res 40:193–226MathSciNetGoogle Scholar

[3]. Aarts EHL, Korst JHM, Michiels W (2005) Simulated annealing. In: Burke EK, Kendall G (eds) Search methodologies. Springer, New York, pp 187–210Google Scholar

[4]. Abramson D (1991) Constructing school timetables using simulated annealing: sequential and parallel algorithms. Manag Sci 37:98–113Google Scholar

[5]. Alrefaei MH, Andradottir S (1999) A simulated annealing algorithm with constant temperature for discrete stochastic optimisation. Manag Sci 45:748–764zbMATHGoogle Scholar

[6]. Altiparmak F, Karaoglan I (2008) An adaptive tabu-simulated annealing for concave cost transportation problems. J Operational Res Soc 59:331–341zbMATHGoogle Scholar

[7]. Anagnostopoulos A, Michel L, Van Hentenryck P, Vergados YA (2006) Simulated annealing approach to the traveling tournament problem. J Scheduling 9:177–193zbMATHGoogle Scholar.

[8]. Arai K, Sakakibara J (2006) Estimation of sea surface temperature, wind speed and water vapour with microwave radiometer data based on simulated annealing. Adv Space Res 37(12):2202–2207Google Scholar

[9]. Azizi N, Zolfaghari S (2004) Adaptive temperature control for simulated annealing: a comparative study. Comput Operations Res 31(4):2439–2451MathSciNetzbMATHGoogle Scholar.

[10]. Bai R, Burke EK, Kendall G, McCollum B (2006) A simulated annealing hyper-heuristic for university course timetabling. In: Burke EK, Rudova H (eds) In: Proceedings of PATAT 2006, Brno, Czech Republic, August–September 2006. Lecture notes in computer science, vol 3867. Springer, HeidelbergGoogle Scholar.

[11]. Bianci L, Dorigo M, Gambardella LM, Gutjahr WJ (2008) A survey on metaheuristics for stochastic combinatorial optimisation. Nat Comput. DOI 101007, Online September 2008Google Scholar.

.