B.tech Project External Evaluation ,VIIIth Sem

STROKE PREDICTION

Presented by

AKHIL SIBI, 2018008191

KEVIN SABU, 2018004754

ANIKET SHUKLA, 2018006541

Supervised by

JYOTSNA SETH , Asst. Professor (CSE)

Sharda University, Gr. Noida

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING AND TECHNOLOGY
May 13, 2022

Approval from guide for the evaluation

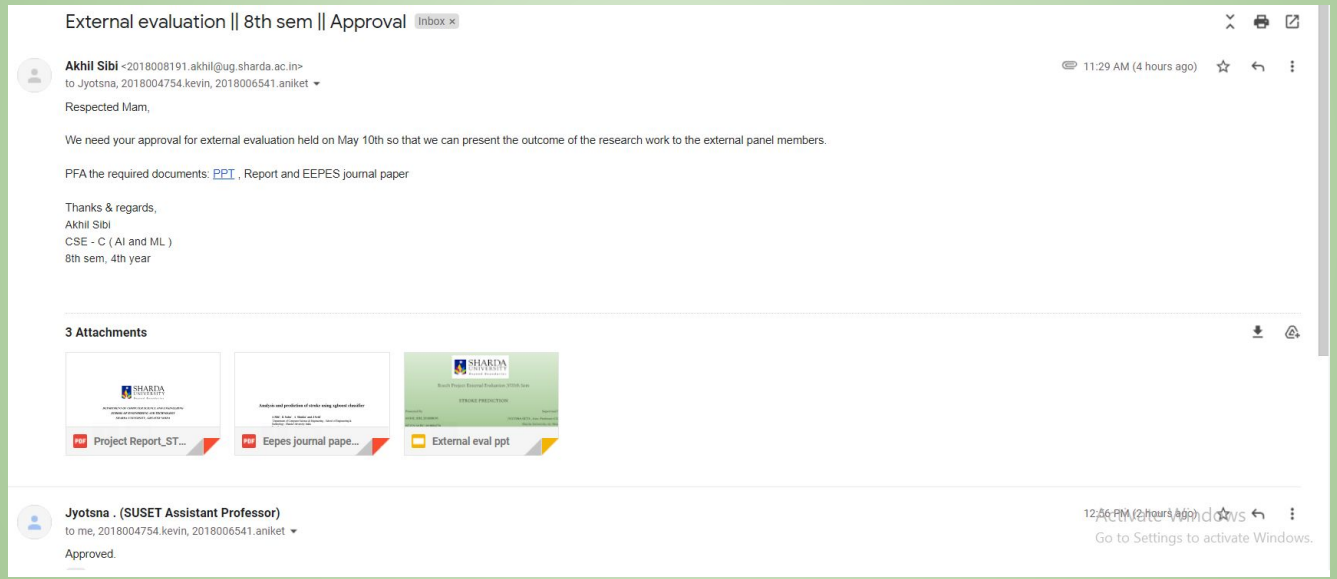


Figure 1. Screenshot of the approval in mail by the guide

Contents of the Presentation

1) Workload distribution
2) Introduction
3) Software Requirements specifications
4) Literature review
5) Design
6) Methodology
7) Results
8) Conclusion
9) Project outcome
10) Project report
11) Github link
12) References

WORKLOAD DISTRIBUTION

1) AKHIL SIBI : 60% Journal paper , 50% implementation , 50% project report
2) ANIKET SHUKLA : 20% Journal paper , 25 % implementation , 25% project report
3) KEVIN SABU : 20% Journal paper , 25 % implementation , 25% project report

INTRODUCTION

Stroke is the second leading cause of death worldwide, accounting for approximately 11% of all deaths, according to the World Health Organization (WHO). It causes difficulty walking and speaking, as well as facial paralysis or numbness, and the patient's life expectancy ranges from 1 to 5 years.

The early prediction of stroke incidence is a critical component of our research work because it allows us to avoid the worst-case scenario and provide proper treatment at the right time. The dataset is open source, taken from Kaggle and attributed to the Author: 'fedesoriano'

Our Proposed approach is the comparative analysis of various machine learning classifiers that are trained, tested, and tuned with hyperparameters and finding out which is the best classifier. Choosing the classifiers for modeling the prediction system will be based on a review of past research works, which will provide us with the essential information and knowledge to move forward with the project.

SOFTWARE REQUIREMENTS SPECIFICATIONS

Software requirements specification (SRS) is a Technical document that specifies the functionality of a piece of software or a product to stakeholders and customers. It is completed prior to the design process, and the document discusses the product's functional and nonfunctional needs, who is targeted for the product launch based on a market survey, and how the product will contribute to the company's growth and profitability. While developing the prediction system, we considered the various types of requirements, such as functional and nonfunctional requirements, as well as how they interact with one another, in order to create a personalized and realistic prediction system that will accurately predict the chances of stroke when given patient data.

FUNCTIONAL REQUIREMENTS

1) for the integrity of our research work, the prediction system must have an accuracy of more than 90% while making predictions on the testing dataset or the inputs provided, because accuracy matters a lot in the field of medicine or any other, and so good accuracy ensures the software's reliability to make the right predictions.
2) Prediction system should be able to provide accurate results in a short period of time, in terms of seconds.

NON FUNCTIONAL REQUIREMENTS

1) RELIABILITY: There should be no manipulation with test results within the prediction system, and the test findings on stroke should always be accurate to the point so that the doctor can make the correct diagnosis.

2) PERFORMANCE: In this fast-paced world and in the medical profession, performance matters, and failure to load the data or lag in the prediction system will only slow the diagnosis of patients who are at risk of having a stroke in the future.

HARDWARE AND SOFTWARE SPECIFICATIONS

Our study hardware is a simple, efficient laptop with an i5 8th gen processor, a minimal graphic card, and 8gb ram. Good internet access ensured the seamless operation of the study for research and training of the models in Google colab notebook, and university access to extra hardware resources such as computer labs, etc. facilitated the growth of our work in the field of stroke.

Instead of Jupyter notebook, we chose Google Colab notebook, which is cloud-based and provides free access to powerful GPUs in the cloud. The notebook also comes preinstalled with the necessary python libraries for machine learning, such as scikit learn, pandas, numpy, matplotlib, and so on, so we don't have to worry about installing them via pip.

LITERATURE SURVEY

We did a thorough analysis of past studies in stroke prediction from a range of well-known sources, including IEEE, ScienceDirect, Research Gate, Scopus, and others, and compared them based on findings, algorithms used with accuracy, benefits, and limitations. The study helped us determine the most common risk factors for stroke occurrence. These past research works inspired us to create our prediction system, and the approaches used in these previous works aided us in investigating alternative powerful machine learning algorithms such as XGBoost, LightGBM, and CatBoost, among others.

AUTHORS	SOURCE	FINDINGS	ALGORITHMS	ADVANTAGES	LIMITATIONS
M. S. Singh and P. Choudhary [1]	IEEE Xplore	A comparison of their proposed AI-based approach with other methods trained on the CHS dataset.	The proposed Classification model is built with a Decision Tree for feature selection, a PCA for dimensionality reduction, and a back propagation neural network for Classification, and it predicts Stroke with an accuracy of 97.70%.	The research work is straightforward, highlighting the steps in developing their proposed approach, which can be used to train other high-level machine learning models.	The research work inspires our own work, but the only limitation is that alternative higher level deep learning techniques are not explored, which could have further enriched their observations with new insights.

Figure 2. survey of research works on stroke prediction

PROPOSED APPROACH

Our proposed method is to do a comparison study of several machine learning classifiers that have been trained, tested, and tweaked with hyperparameters to decide which classifier is the best. The identification of risk factors from datasets utilizing visualizations and training machine learning algorithms on these characteristics is crucial for successful prediction and understanding of stroke incidence. Our proposed approach ensures that the research activity coincides with novelty and that the end result is reached so that we may share the results with the scientific community in the form of a research paper and deploy the prediction system in real circumstances for the doctors to diagnose stroke patients.

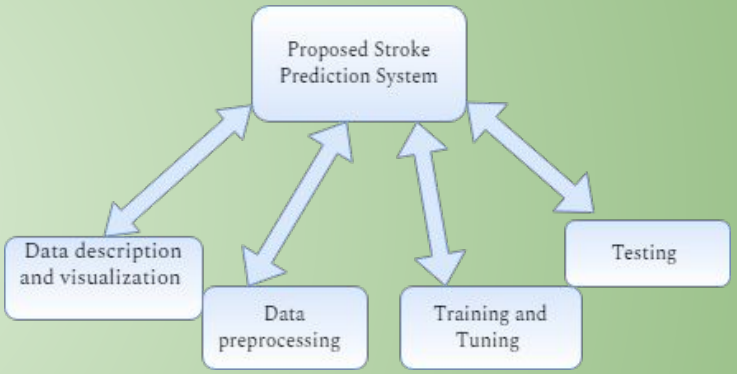


Figure 3. proposed stroke prediction system

FEASIBILITY STUDY

The Feasibility study was completed very early in the project's development in order to assess the practicability of the proposed system and whether it would be successful in execution and delivery or not. We have covered all the aspects in the study ranging from Technical and Financial areas and have come to the conclusion that the research work is feasible to conduct and it will be a great progress in dealing with stroke and prediction of the stroke through early onset symptoms to save people’s lives.

RISK MANAGEMENT

Risks can arise while making our prediction system that can impact the performance of the models. These risks can range from low accuracy and ROC AUC scores etc and so to manage these risks , we have to balance the dataset in preprocessing using smote , hyperparameter tuning is essential during training so that to drastically increase the accuracy of the models on the testing dataset.

DESIGN

Designing the prediction system is a key component of the planning phase prior to implementation and coding since it allows us to envision how the system works, its behavior, and the actions required.

Figure 4 represents the flowchart which depicts the flow of the project from start to finish in a sequential manner AND These steps are clear, well thought out and concise enough that we have built the prediction system from scratch by following the methodology highlighted in the flowchart that ensured the positive success rate of the research work and ensured insightful , rich observations from the study

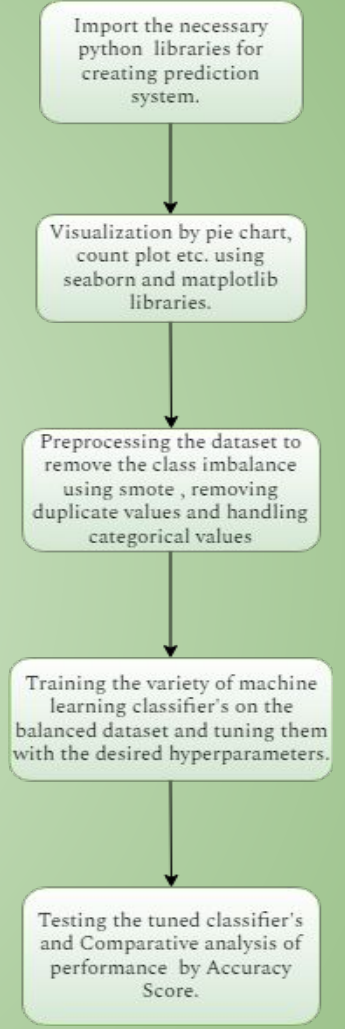


Figure 4. Flowchart of the Prediction system

DFD

Data flow diagrams (DFDs) are graphical representations of system level designs that show the ins and outs of the flow of information or data and how the results will be generated from the inputs provided to the system. DFDs are very useful in the planning phase of SDLC, as well as in the requirement gathering phase, where visualization of the system's operation at various stages of DFD can guide the development of the product or software. Therefore we have shown a simple dfd 0 level diagram for our stroke prediction system that can help in understanding the inner workings of the system.

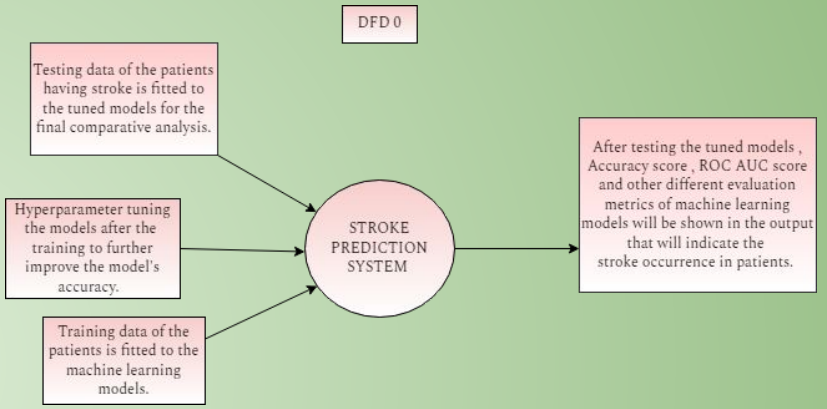


Figure 5. Level 0 DFD of Stroke prediction system

SDLC MODEL

For project management, we used Agile approach in which we divided the creation of the prediction system into several iterations , as shown in the flowchart, resulting in improved collaboration and project execution. Its advantages include efficient project management , promotes greater team collaboration and ensures work life balance and productivity of the employees

METHODOLOGY

Following the design of the prediction system, we proceeded with the implementation by following the research stages outlined in order to reach our study objective of comparison analysis.

1) **Data description**: we import the necessary python libraries for building our prediction system like Pandas , matplotlib etc. for reading and analyzing the dataset and visualizing the key risk factors affecting stroke occurrence.For our modeling, we use the Stroke Prediction Dataset from Kaggle , which contains 5110 Patient Records with feature characteristics related to the patient's lifestyle, such as BMI, Heart disease, hypertension, smoking, gender, and so on.
2) **Data visualization**: To highlight the relationship between the plotted features, we developed amazing visuals using matplotlib, seaborn libraries, and other tools. The major purpose is to raise awareness about stroke and how different clinical variables in the dataset play a significant effect in stroke incidence.

	id	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_sta
0	9046	Male	67.0	0	1	Yes	Private	Urban	228.69	36.6	formerly smol
1	51676	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	NaN	never smol
2	31112	Male	80.0	0	1	Yes	Private	Rural	105.92	32.5	never smol
3	60182	Female	49.0	0	0	Yes	Private	Urban	171.23	34.4	smo
4	1665	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.0	never smol

Figure 6. Dataset description

```
labels =df['stroke'].value_counts(sort = True).index
sizes = df['stroke'].value_counts(sort = True)

colors = ["lightblue","red"]
explode = (0.05,0)

plt.figure(figsize=(7,7))
plt.pie(sizes, explode=explode, labels=labels, colors=colors, autopct='%1.1f%%', shadow=True, startangle=90,)

plt.title('Number of stroke in the dataset')
plt.show()
```

Figure 7. Countplot of stroke number

3) **Data preprocessing**: it is a very important and crucial step in data science and machine learning because the dataset is often imbalanced in nature, has duplicate and null values, and so it's a must to preprocess and clean the dataset very efficiently without losing the original data and so the first important step is to import the necessary libraries for preprocessing, such as LabelEncoder, OneHotEncoder, and ColumnTransformer, which will handle the Categorical values of the dataset and convert them to numerical, and finally we have SMOTE, which will balance the dataset for model training and improved performance.

4) **Training**: we import numerous types of machine learning classifiers from the sklearn library or pip install the relevant ones, which are as follows: XGBoost[15], LightGBM[19], CatBoost[20], Random Forest, AdaBoost, MLP, SVM, Logistic Regression, Decision Tree, KNeighbours, BernoulliNB, and GaussianNB. We will train and test each model, calculating its Accuracy Score, ROC AUC Score, Precision Score, Recall, F1 Score, and so on.)

```
// import the encoders for categorical value's and SMOTE for class imbalance.

from imblearn.over_sampling import SMOTE
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
df['gender'] = le.fit_transform(df['gender'])
df['ever_married'] = le.fit_transform(df['ever_married'])
df['work_type'] = le.fit_transform(df['work_type'])
df['Residence_type'] = le.fit_transform(df['Residence_type'])
df['smoking_status'] = le.fit_transform(df['smoking_status'])
```

```
x = df.iloc[:,1:-1].values
y = df.iloc[:,-1].values

print('X Shape', x.shape)
print('Y Shape',y.shape)
```

```
X Shape (5110, 10)
Y Shape (5110,)
```

```
ct = ColumnTransformer(transformers=[('encoder',OneHotEncoder(),[0,5,9])],remainder='passthrough')
x = np.array(ct.fit_transform(x))
```

Figure 8. Label encoders

```
# import the machine learning models

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.naive_bayes import BernoulliNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from lightgbm import LGBMClassifier
from sklearn.neural_network import MLPClassifier
from catboost import CatBoostClassifier
from sklearn.ensemble import AdaBoostClassifier
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, roc_auc_score, ConfusionMatrixDisplay, precision_score,
from sklearn.model_selection import cross_val_score
```

```
models = []
models.append(['Logistic Regreesion', LogisticRegression(random_state=0)])
models.append(['SVM', SVC(random_state=0)])
models.append(['KNeighbors', KNeighborsClassifier()])
models.append(['GaussianNB', GaussianNB()])
models.append(['BernoulliNB', BernoulliNB()])
models.append(['Decision Tree', DecisionTreeClassifier(random_state=0)])
models.append(['Random Forest', RandomForestClassifier(random_state=0)])
models.append(['XGBoost', XGBClassifier(eval_metric= 'error')])
models.append(['LightGBM',LGBMClassifier(random_state = 42)])
models.append(['Catboost',CatBoostClassifier(n_estimators=150,l2_leaf_reg=0.1,verbose = 0)])
models.append(['Adaboost',AdaBoostClassifier(n_estimators=2000, random_state = 0)])
models.append(['MLP', MLPClassifier()])
```

Figure 9. Importing the classifiers for modeling

5) **Tuning**: To improve the accuracy and performance of the models even further, we will train them again on the dataset with the desired hyperparameters which will take some time to train. After Fine-tuning the models, we Fit the Testing data to see which one is the best. Hyperparameters chosen can range from Learning rate , evaluation metrics etc.

6) **Testing**: After tuning the models , we come to a conclusion that Random forest , CatBoost , XGBoost and LightGBM are having the best performance in terms of training accuracy and so it is obvious that for our final comparative analysis of the models on the testing dataset , we will test these four models mentioned and not take the other ones. Testing will be done on the basis of their accuracy score and ROC AUC score which will clearly reveal to us which is the best classifier for model building in the future where the system will be used in the real time deployment in hospitals.

The research methodology should be followed step by step, with careful consideration given to clean code, visualizations, model development, and tuning, so that we may have a good prediction system with minimal error, and all protocols, ethics, and regulations must be followed.

```
from sklearn.model_selection import GridSearchCV
grid_models = [(LogisticRegression(),[{'C':[0.25,0.5,0.75,1],'random_state':[0]}]),
               (KNeighborsClassifier(),[{'n_neighbors':[5,7,8,10], 'metric': ['euclidean', 'manhattan', 'chebyshev', 'minkowski']}]),
               (SVC(),[{'C':[0.25,0.5,0.75,1],'kernel':['linear', 'rbf'],'random_state':[0]}]),
               (GaussianNB(),[{'var_smoothing': [1e-09]}]),
               (BernoulliNB(), [{'alpha': [0.25, 0.5, 1]}]),
               (DecisionTreeClassifier(),[{'criterion':['gini','entropy'],'random_state':[0]}]),
               (RandomForestClassifier(),[{'n_estimators':[100,150,200],'criterion':['gini','entropy'],'random_state':[0]}]),
              (XGBClassifier(), [{'learning_rate': [0.01, 0.05, 0.1], 'eval_metric': ['error']}]),
              (LGBMClassifier(),[{'learning_rate': [0.01, 1.0],'num_leaves': [24, 80]}]),
              (CatBoostClassifier(),[{ 'learning_rate': [0.03, 0.1]}]),
              (AdaBoostClassifier(),[{'learning_rate':[0.1, 0.3]}])

              ]
```

```
for i,j in grid_models:
    grid = GridSearchCV(estimator=i,param_grid = j, scoring = 'accuracy',cv = 10)
    grid.fit(X_train_res, y_train_res)
    best_accuracy = grid.best_score_
    best_param = grid.best_params_
    print('{}:\nBest Accuracy : {:.2f}%'.format(i,best_accuracy*100))
    print('Best Parameters : ',best_param)
    print('')
    print('----------------')
    print('')
```

Figure 10. Tuning of the models with the desired parameters

```
classifier = XGBClassifier(eval_metric= 'error', learning_rate= 0.1)
classifier.fit(X_train_res, y_train_res)
y_pred = classifier.predict(X_test)
y_prob = classifier.predict_proba(X_test)[:,1]
cm = confusion_matrix(y_test, y_pred)

print(classification_report(y_test, y_pred))
print(f'ROC AUC score: {roc_auc_score(y_test, y_prob)}')
print('Accuracy Score: ',accuracy_score(y_test, y_pred))
```

Figure 11. Testing tuned XGBoost

RESULT

So, first and foremost, we'd want to highlight the visualizations and plottings of the dataset features that were crucial in the connection of the features with stroke occurrence.

In this figure 17 is a pie chart plot of the number of people suffering from stroke in the dataset , we come to know only 4.9% suffer from stroke and are positive while 95.1% are negative with stroke and so it gives us a glimpse on the number of the stroke patients. So just like that , it will be followed by multiple other visualization on the clinical features of patients like work type , hypertension and smoking status etc and so in the figure , we have demonstrated the correlation of the clinical features or the risk factors that contribute to stroke from the dataset and as seen in the previous plottings and now this , we have come to know that with age , chances of stroke increases followed by heart Disease , hypertension and marital status etc and so with age , one should be cautious of their lifestyle habits

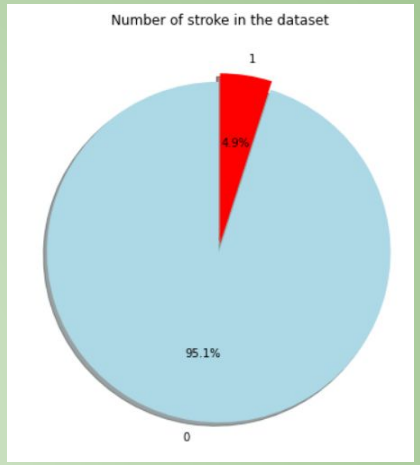


Figure 12. Pie Chart on number of stroke affected people

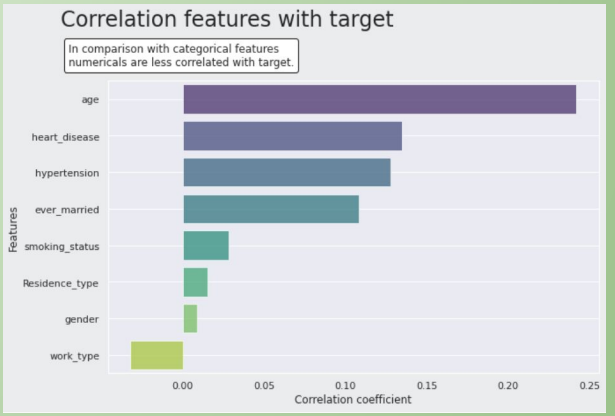


Figure 13. Correlation of features.

After visualization results , we have showcased the results of training and testing of the models before the tuning phase and seen in the figure 25 we compare the models performance with accuracy , ROC- AUC , K - Fold Mean Accuracy, Precision , recall , f1 score and standard deviation in which we have to only mind accuracy score and roc auc score and it is observed that XGBoost is the top performing model in terms of all the performance metrics highlighted having an accuracy score of 94.61% and roc auc score of 0.52 followed by LightGBM , Catboost , Random Forest and many others etc.

After this Tuning comes, which is an integral part of our research work that aims to further enrich the models accuracy and give us new observations on the performance of the models on the testing dataset .Therefore , we tune each one of the Classifiers with their Desired parameters (Learning rate , n_estimators , eval_metrics , number of leaves etc) using gridsearchcv so that we can further improve their performance and so after , we train them once again on the training dataset to note their training accuracy and so after training we observe that these four models which are XGBoost , Random forest , CatBoost and LightGBM stand out from others while achieving higher testing accuracy with their best defined parameter and so it stands to reason that we will only test these four to determine which model is the best for real-time stroke prediction and for comparative analysis purposes and so we fit the tuned and trained model on the testing dataset once again to see the improved accuracy score and roc auc score.

	Model	Accuracy	K-Fold Mean Accuracy	Std. Deviation	ROC AUC	Precision	Recall	F1
7	XGBoost	94.618395	96.764621	7.229892	0.525712	0.428571	0.055556	0.098361
8	LightGBM	94.520548	96.944389	6.986720	0.525195	0.375000	0.055556	0.096774
9	Catboost	94.520548	96.430712	7.157676	0.525195	0.375000	0.055556	0.096774
6	Random Forest	94.422701	97.162683	6.762548	0.515936	0.285714	0.037037	0.065574
10	Adaboost	93.835616	96.443450	7.166005	0.521579	0.200000	0.055556	0.086957
5	Decision Tree	89.726027	94.927664	5.821712	0.578570	0.160000	0.222222	0.186047
4	BernoulliNB	84.442270	87.092872	3.878098	0.646847	0.152318	0.425926	0.224390
2	KNeighbors	81.800391	89.121262	0.992939	0.676615	0.148936	0.518519	0.231405
11	MLP	80.821918	81.955048	2.442578	0.662707	0.137755	0.500000	0.216000
0	Logistic Regreesion	77.690802	78.473787	1.689052	0.751090	0.154762	0.722222	0.254902
1	SVM	72.113503	78.499724	1.881722	0.739134	0.130990	0.759259	0.223433
3	GaussianNB	33.170254	64.500315	1.789182	0.629725	0.070941	0.962963	0.132147

Figure 14. Comparative Analysis of Classifiers after training and testing

	precision	recall	f1-score	support
0	0.95	1.00	0.97	968
1	0.43	0.06	0.10	54
accuracy			0.95	1022
macro avg	0.69	0.53	0.54	1022
weighted avg	0.92	0.95	0.93	1022

ROC AUC score: 0.7911023109886747
Accuracy Score: 0.9461839530332681

Figure 15. Tuned XGBoost performance

	precision	recall	f1-score	support
0	0.95	0.99	0.97	968
1	0.29	0.04	0.07	54
accuracy			0.94	1022
macro avg	0.62	0.52	0.52	1022
weighted avg	0.91	0.94	0.92	1022

ROC AUC score: 0.747876492194674
Accuracy Score: 0.9442270058708415

Figure 16. Tuned Random forest performance

	precision	recall	f1-score	support
0	0.95	0.99	0.97	968
1	0.38	0.06	0.10	54
accuracy			0.95	1022
macro avg	0.66	0.53	0.53	1022
weighted avg	0.92	0.95	0.93	1022

ROC AUC score: 0.7794038873584327
Accuracy Score: 0.9452054794520548

Figure 17. Tuned LightGBM performance

	precision	recall	f1-score	support
0	0.95	0.99	0.97	968
1	0.30	0.06	0.09	54
accuracy			0.94	1022
macro avg	0.62	0.52	0.53	1022
weighted avg	0.92	0.94	0.92	1022

ROC AUC score: 0.7665863177226814
Accuracy Score: 0.9432485322896281

Figure 18. Tuned CatBoost performance

CONCLUSION

From Figure 15 to 18 , we see the performance of these 4 Tuned models on the Testing data once again and so it is clear from the new observations that XGBoost is the best performing model on the stroke dataset during the comparative analysis and it achieves an impressive all rounder accuracy score of 94.61% and roc auc score of 0.79 and it is followed by LightGBM , Random forest and Catboost and their results are also neatly compiled in the below table for more clarity and so in conclusion these are the final results from our research work of comparative analysis on stroke dataset for the prediction system in which we have tried all the machine learning approaches and in the XGBoost emerged as the winner and best classifier for modeling of the prediction system.

XGBoost stroke prediction model should be integrated in real-time stroke prediction applications and hardware systems where it will produce accurate results since it has been vigorously trained , tuned and tested on the stroke dataset and has shown an impressive allrounder results

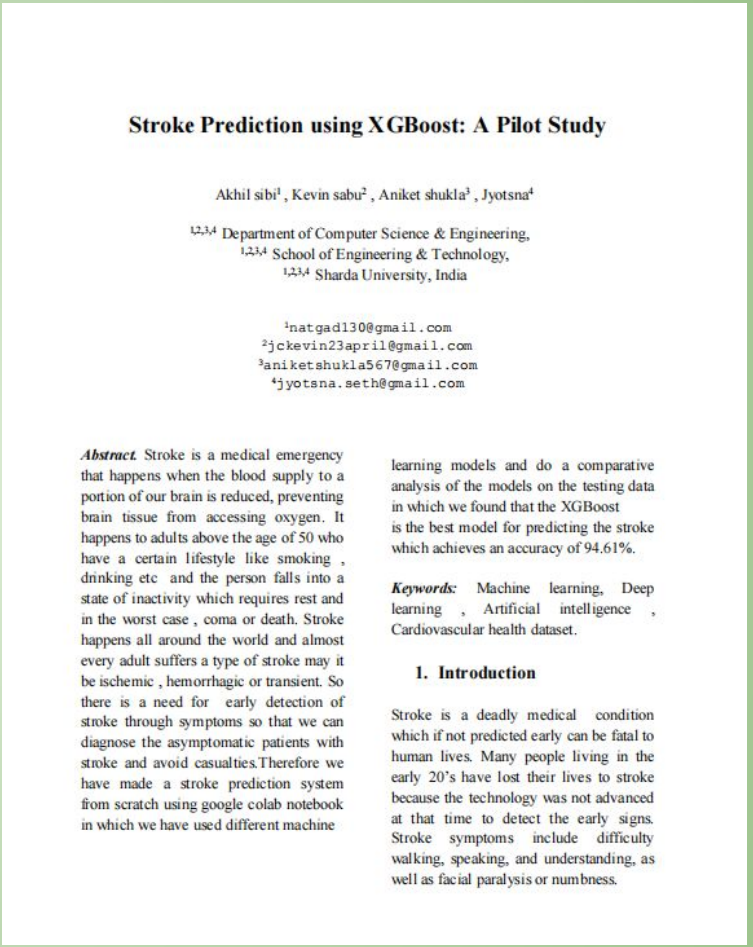
Table 8.1 comparative analysis of tuned classifiers

Model	Accuracy Score
Xgboost	94.61 %
LightGBM	94.52%
Random forest	94.42 %
Catboost	94.32 %

PROJECT OUTCOME

7th sem

Our project work is research-oriented, and the insights and results will be shared with the scientific community. For the 7th semester outcome, we had a simple research paper prepared in IEEE format, in which the results were neatly compiled, and then we submitted the paper to an IEEE conference named International conference of emerging technologies (INCET) in Bengaluru, but the paper was rejected during acceptance because some changes were required.



Stroke Prediction using XGBoost: A Pilot Study

Akhil sibi[1] , Kevin sabu[2] , Aniket shukla[3] , Jyotsna[4]

[1,2,3,4] Department of Computer Science & Engineering,
[1,2,3,4] School of Engineering & Technology,
[1,2,3,4] Sharda University, India

[1]natgad130@gmail.com
[2]jckevin23april@gmail.com
[3]aniketshukla567@gmail.com
[4]jyotsna.seth@gmail.com

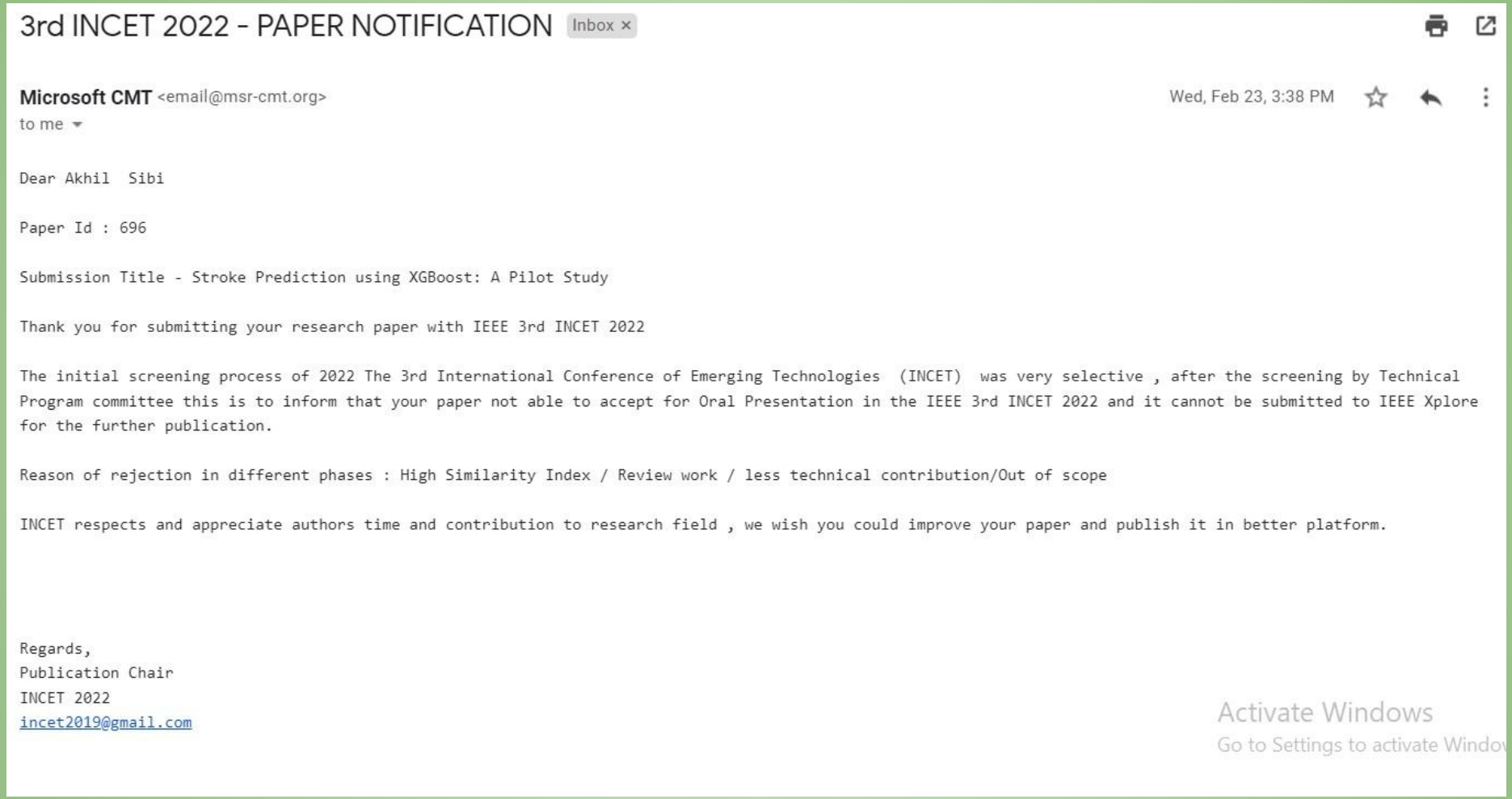
Abstract. Stroke is a medical emergency that happens when the blood supply to a portion of our brain is reduced, preventing brain tissue from accessing oxygen. It happens to adults above the age of 50 who have a certain lifestyle like smoking , drinking etc and the person falls into a state of inactivity which requires rest and in the worst case , coma or death. Stroke happens all around the world and almost every adult suffers a type of stroke may it be ischemic , hemorrhagic or transient. So there is a need for early detection of stroke through symptoms so that we can diagnose the asymptomatic patients with stroke and avoid casualties.Therefore we have made a stroke prediction system from scratch using google colab notebook in which we have used different machine learning models and do a comparative analysis of the models on the testing data in which we found that the XGBoost is the best model for predicting the stroke which achieves an accuracy of 94.61%.

Keywords: Machine learning, Deep learning , Artificial intelligence , Cardiovascular health dataset.

1. Introduction

Stroke is a deadly medical condition which if not predicted early can be fatal to human lives. Many people living in the early 20's have lost their lives to stroke because the technology was not advanced at that time to detect the early signs. Stroke symptoms include difficulty walking, speaking, and understanding, as well as facial paralysis or numbness.

Figure 19. 7th sem research paper



3rd INCET 2022 - PAPER NOTIFICATION Inbox ×

Microsoft CMT <email@msr-cmt.org> Wed, Feb 23, 3:38 PM

to me

Dear Akhil Sibi

Paper Id : 696

Submission Title - Stroke Prediction using XGBoost: A Pilot Study

Thank you for submitting your research paper with IEEE 3rd INCET 2022

The initial screening process of 2022 The 3rd International Conference of Emerging Technologies (INCET) was very selective , after the screening by Technical Program committee this is to inform that your paper not able to accept for Oral Presentation in the IEEE 3rd INCET 2022 and it cannot be submitted to IEEE Xplore for the further publication.

Reason of rejection in different phases : High Similarity Index / Review work / less technical contribution/Out of scope

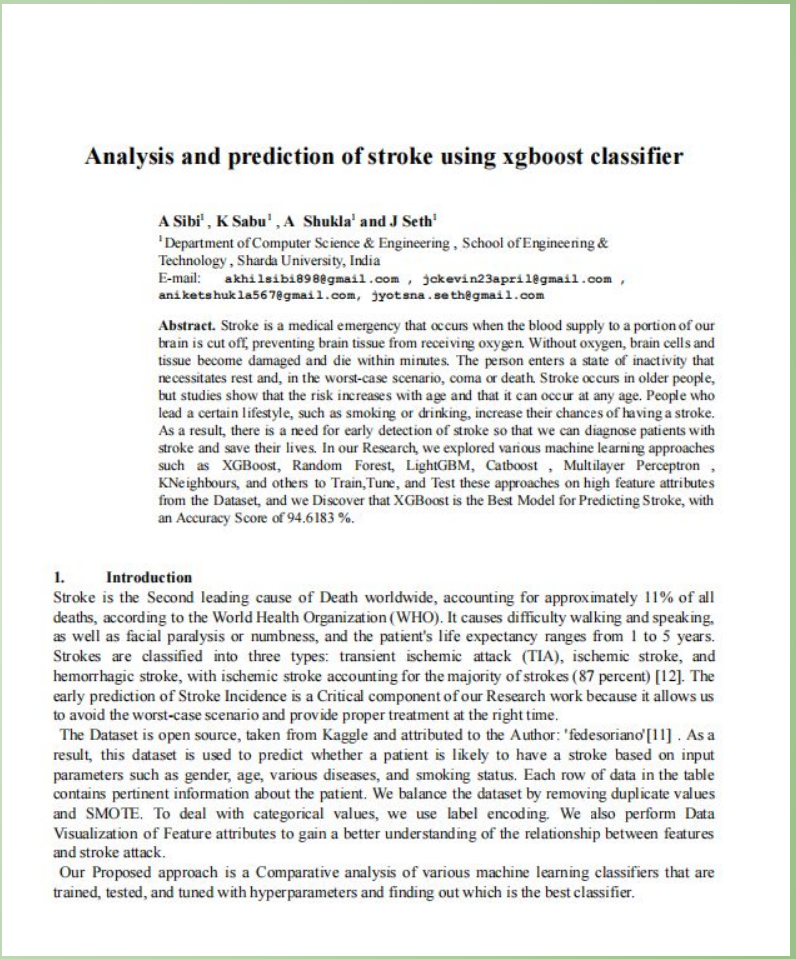
INCET respects and appreciate authors time and contribution to research field , we wish you could improve your paper and publish it in better platform.

Regards,
Publication Chair
INCET 2022
incet2019@gmail.com

Figure 20. INCET Paper notification

8th sem

Upon the positive feedback from INCET , In the eighth semester, we added four more classifiers to the comparative analysis, which were AdaBoost, CatBoost, LightGBM, and MLP neural network, among others, and we intricately formatted the paper from IEEE format to Journal of physics: Conference series, in which formatted and added more content to the literature review, results, and methodology, among others, and we submitted the paper to an International Conference on Electronics , Engineering Physics and Earth Science , 2022 (EEPES) held in hybrid mode
The good news is that our journal paper has been conditionally accepted in the prestigious EEPES conference and so we have to make the following revisions to the paper according to the editors and reviewers comments by May 30 and then we are all good to go for the conference and publication of our paper to the Journal of physics: Conference series.



Analysis and prediction of stroke using xgboost classifier

A Sibi[1] , K Sabu[1] , A Shukla[1] and J Seth[1]
[1]Department of Computer Science & Engineering , School of Engineering & Technology , Sharda University, India
E-mail: akhilsibi898@gmail.com , jckevin23april@gmail.com , aniketshukla567@gmail.com, jyotsna.seth@gmail.com

Abstract. Stroke is a medical emergency that occurs when the blood supply to a portion of our brain is cut off, preventing brain tissue from receiving oxygen. Without oxygen, brain cells and tissue become damaged and die within minutes. The person enters a state of inactivity that necessitates rest and, in the worst-case scenario, coma or death. Stroke occurs in older people, but studies show that the risk increases with age and that it can occur at any age. People who lead a certain lifestyle, such as smoking or drinking, increase their chances of having a stroke. As a result, there is a need for early detection of stroke so that we can diagnose patients with stroke and save their lives. In our Research, we explored various machine learning approaches such as XGBoost, Random Forest, LightGBM, Catboost , Multilayer Perceptron , KNeighbours, and others to Train,Tune, and Test these approaches on high feature attributes from the Dataset, and we Discover that XGBoost is the Best Model for Predicting Stroke, with an Accuracy Score of 94.6183 %.

1. Introduction

Stroke is the Second leading cause of Death worldwide, accounting for approximately 11% of all deaths, according to the World Health Organization (WHO). It causes difficulty walking and speaking, as well as facial paralysis or numbness, and the patient's life expectancy ranges from 1 to 5 years. Strokes are classified into three types: transient ischemic attack (TIA), ischemic stroke, and hemorrhagic stroke, with ischemic stroke accounting for the majority of strokes (87 percent) [12]. The early prediction of Stroke Incidence is a Critical component of our Research work because it allows us to avoid the worst-case scenario and provide proper treatment at the right time.

The Dataset is open source, taken from Kaggle and attributed to the Author: 'fedesoriano'[11] . As a result, this dataset is used to predict whether a patient is likely to have a stroke based on input parameters such as gender, age, various diseases, and smoking status. Each row of data in the table contains pertinent information about the patient. We balance the dataset by removing duplicate values and SMOTE. To deal with categorical values, we use label encoding. We also perform Data Visualization of Feature attributes to gain a better understanding of the relationship between features and stroke attack.

Our Proposed approach is a Comparative analysis of various machine learning classifiers that are trained, tested, and tuned with hyperparameters and finding out which is the best classifier.

Figure 21. 8th sem journal paper

[EEPES 2022] Your submission has been conditionally accepted - needs revision! Inbox ×

EEPES 2022 <office@eepes.eu> 18:37 (21 minutes ago)

to me, jckevin23april, aniketshukla567, jyotsna.seth

Dear Author/s

On behalf of the Organizing Committee, We are very pleased to inform you that your submission:

ID: 8

Title: Analysis and prediction of stroke using xgboost classifier

HAS BEEN CONDITIONALLY ACCEPTED – NEEDS REVISION for the EEPES 2022 conference!

We have included the reviewers and editors' feedback at the end of this message. You have to address their recommendations and comments*. Failur to do this may result in not forwarding the paper to our Publisher!**

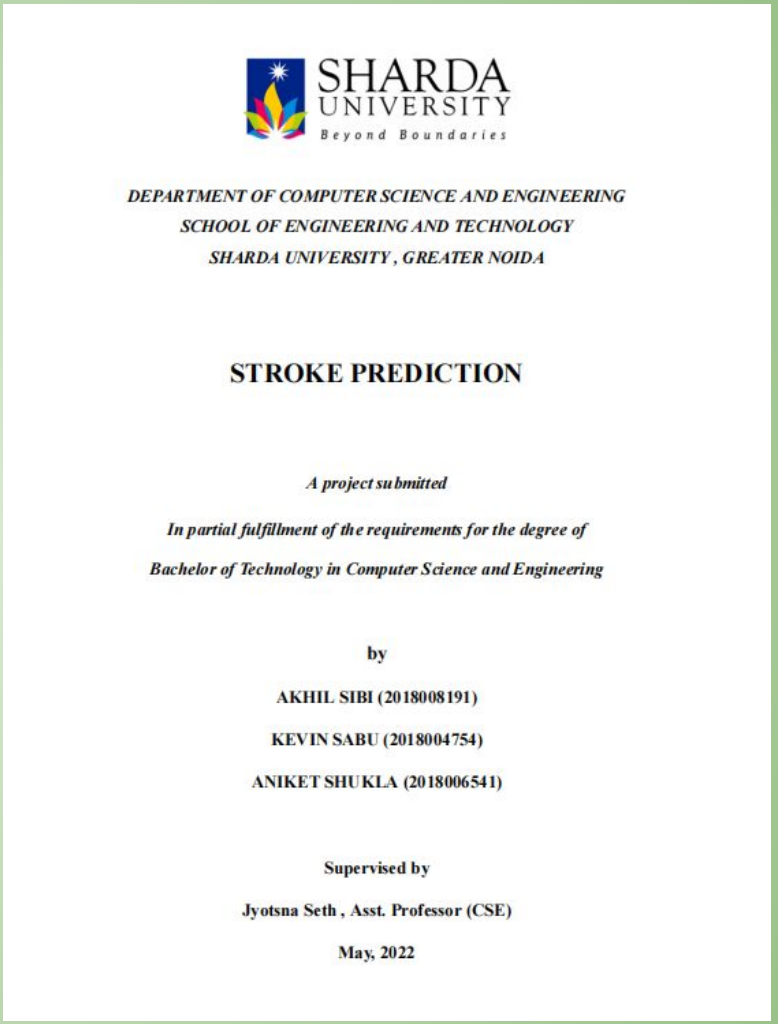
VERY IMPORTANT!

1. Please, fill in the EEPES 2022 Submission & Registration Form: http://eepes.eu/index.php/registration-form and upload your revised manuscript.
2. BOTH, the source file (Word) and the PDF file, must be uploaded.
3. Follow the payment instructions at http://eepes.eu/index.php/fees-and-payment. Paying the registration fee is compulsory in order to present and publish your paper in the conference Proceedings (only papers with paid registration fee will be published).
4. Please, upload the requested files by 30 May 2022.

Figure 22. EEPES acceptance notification

PROJECT REPORT

The project report is well-organized, complete, and well-presented, with sections such as Introduction, Literature review, Design, Methodology, Result, Conclusion, and References. While writing our project report, we adhere to the project report format established by the CSE department.



SHARDA UNIVERSITY
Beyond Boundaries

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING AND TECHNOLOGY
SHARDA UNIVERSITY , GREATER NOIDA

STROKE PREDICTION

A project submitted

In partial fulfillment of the requirements for the degree of

Bachelor of Technology in Computer Science and Engineering

by

AKHIL SIBI (2018008191)

KEVIN SABU (2018004754)

ANIKET SHUKLA (2018006541)

Supervised by

Jyotsna Seth , Asst. Professor (CSE)

May, 2022

Figure 23. Title page of project report

As shown in Figure 22, by running the Turnitin check on the project report, we achieved a similarity index of 4%, which is less than the 15% specified by the CSE department, indicating that the originality of the report is preserved throughout the writing process and the work done is of high quality.


Project_stroke

ORIGINALITY REPORT

4%	3%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	devashree-madhugiri.medium.com Internet Source	<1%
2	ebin.pub Internet Source	<1%
3	www.science.gov Internet Source	<1%
4	Chidozie Shamrock Nwosu, Soumyabrata Dev, Peru Bhardwaj, Bharadwaj Veeravalli, Deepu John. "Predicting Stroke from Electronic Health Records", 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2019 Publication	<1%
5	link.springer.com Internet Source	<1%
6	Tianyu Liu, Wenhui Fan, Cheng Wu. "A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical	<1%

Figure 24. Similarity index of the report

GITHUB LINK

The team also has uploaded the google colab notebook , project report and the ppt in the github repository , where they may be easily accessible and seen for future reference.

PFA the link : StrokePrediction_MajorProject

REFERENCES

1) M. S. Singh and P. Choudhary, “Stroke prediction using artificial intelligence,” 2017 8th Ind. Autom. Electromechanical Eng. Conf. IEMECON 2017, pp. 158–161, Oct. 2017, doi: 10.1109/IEMECON.2017.8079581.
2) C. H. Lin et al., “Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry,” Comput. Methods Programs Biomed., vol. 190, p. 105381, Jul. 2020, doi: 10.1016/J.CMPB.2020.105381.
3) T. Liu, W. Fan, and C. Wu, “A hybrid machine learning approach to cerebral stroke prediction based on imbalanced medical dataset,” Artif. Intell. Med., vol. 101, p. 101723, Nov. 2019, doi: 10.1016/J.ARTMED.2019.101723.
4) S. N. Min, S. J. Park, D. J. Kim, M. Subramaniyam, and K. S. Lee, “Development of an Algorithm for Stroke Prediction: A National Health Insurance Database Study in Korea,” Eur. Neurol., vol. 79, no. 3–4, pp. 214–220, May 2018, doi: 10.1159/000488366.
5) N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/JAIR.953.
6) T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-August-2016, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785.

THANK YOU !!!