SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I8P102

Volume 11 Issue 8, 14-24, August 2024
© 2024 Seventh Sense Research Group®

Original Article

Cognitive Learning Approach to Enrich Understanding of Machine Learning on Healthcare Data

Prasanna Palsodkar[1,]*, Prachi Palsodkar[1], Yogita Dubey[2], Roshan Umate[2]

[1]*Department of Electronics Engineering, Yeshwantrao Chavan College of Engineering, Maharashtra, India.*
[2]*Department of Research and Development, Jawaharlal Nehru Medical College, Datta Meghe Institute of Higher Education and Research, Maharashtra, India.*

**Corresponding Author : palsodkar.prasanna@gmail.com*

Received: 02 June 2024 Revised: 03 July 2024 Accepted: 01 August 2024 Published: 31 August 2024

Abstract *- Machine Learning (ML) has a significant impact on applications across various disciplines, with a key requirement of domain knowledge. A fully guided cognitive framework is presented in the case study for medical data analysis using the ML approach with an influence of the feature extraction effect, ensemble methods, and voting classifier using hyperparameter tuning. A case study-driven guided project design technique helps the student to comprehend the subject better. The learner gets familiar with reliable data sources and associated analysis jargon. The student is familiar with different intermediate steps, process flow, and legitimate conclusion dragging. The result shows that confidence in capstone project design in the relevant field and handling medical data is developed in learners. The learning removes hesitation of interdisciplinary work in the cognitive classroom. This strategy can successfully drive lifelong learning for all emerging computer science courses.*

Keywords *- Cognitive learning, Diabetic, Ensemble, Healthcare, Machine learning, Project-based learning, Voting classifier.*

1. Introduction

Immersive technologies in engineering courses need highly skilled structuring and execution. Such courses need to be introduced in the curriculum, considering the skill development of learners to cope with market challenges and industry demand.

In those courses, teachers face two-way challenges, knowledge up-skilling and state-of-the-art delivery. An empirical way of learning with live case studies builds the lifelong learning experience of the learner. Effective project development experience in related areas builds the learner's confidence in learning and aids in effective portfolio design.

Artificial Intelligence (AI) is the demanding technology of this tech era. Learners in this domain are increasing day by day. Theoretically, learners are getting good guided support in the classroom. In experimental learning, a learner needs conceptual help and optimum workflow. Most of the time, learners apply a blind approach to executing a project without a proper understanding of the subject matter or experimentation requirements.

A capstone project in Machine Learning (ML) helps learners in their portfolio design. However, the execution of such projects necessitates a thorough understanding of concepts. During laboratory experiments, if the learner understands the requirements of subject matter, utility, workflow, evaluation steps, optimization methodologies, and conceptual conclusions on the results, it helps him have a better capstone project design experience.

This paper gives a brief framework for designing an ML-based project and a deep understanding of the methodologies that need to be invoked by the learner.Project-Based Learning (PBL) is one of the tried-and-true methods for improving subject understanding (I. Calvo et al., 2018).

Although the current computer science and software engineering curricula ensure that students have studied a variety of programming-related classes, they do not guarantee that students will have the social skills necessary for a project to be successful (D. A. Umphress et al., 2002).

Understanding programming restrictions and improving our ability to characterize programming problem-solving are both necessary for ML (R. P. Medeiros et al., 2002). A suitable research roadmap helps learners solve any given challenge. In the classical approach of PBL, learners get assigned project tasks and have to apply their knowledge discretely. This may come out as mechanical experimentation without the capstone of solid conclusions.

Most machine learning beginners, particularly those with non-computer science backgrounds, struggle with technical programming skills. Guidance at different abstraction levels. A guided laboratory approach at different abstraction levels provides the right direction for beginners and an insightful understanding of experimentation.

In this paper, a guided laboratory approach is provided for a machine learning subject learner. A complete guided process is delivered here for ML aspirants seeking insightful project design. Section 2 explains the need for data understanding in ML project execution. Section 3 details why data analysis is important before ML algorithm application. Section 4 gives learning behaviour synthesis and shows an optimization method for driving sensible conclusions. Finally, it is concluded in Section 5.

2. Material and Methods

In this section, generalized learner centric flow for ML development is discussed in brief. New stack ML learners can easily understand the process design flow with the approach discussed in subsequent sections.

2.1. Data Sources

Before applying ML algorithms, the learner must first look for ground truth as well as relevant and high-quality data. The learner must use ethical ways to collect data from reliable resources. Fabricated data may mislead the study. A few reliable sources of data are listed in Table 1.

Table 1. Sources of datasets

Sr. No	Sources
1	Government Datasets (e.g. Indian Gov. data set)
2	UCI Machine Learning Repository
3	Kaggle Data Set
4	Amazon Dataset
5	IEEE Data Port
6	Google Dataset Search Engine
7	Data Sets Sub-reddit

Table 2. Dataset and ML development environments

Ref. No.	Inference	Implications
Gebru, T et al., 2021	Datasheets for Dataset Perception	New Directives for data set generation to maintenance
Margaret Mitchell et al., 2019	Model Cards	Model cards contain details about the model's context and its performance metrics
Boyd, K. L. 2021	Understand Ethical Issues in Training Data	It focuses on the ethical considerations of machine learning engineers (recognition, understanding, and decision making on real-world datasets).
Chmielinski, K. S et al., 2022	Leveraging Framework to Lessen Harms in AI	Concept, design, depth, and utility of the dataset development process
Yavanoglu, O et al., 2017	Cyber Security Datasets	Datasets for AI and ML to find network traffic and abnormalities
Lemaître, G et al., 2017	Imbalanced Dataset Learn	Imbalanced-learn tool box insight

2.2. Data Analysis

Downloading the data set and thoroughly understanding it from reliable sources is the first step in ML enthusiasm. Exploratory Data Analysis (EDA) is the essential step before applying ML. Data preparation is one of the essential steps of the ML project lifecycle because the quality of the data influences the quality of the ML model (Patel, H. et al., 2022). EDA, with effective data visualization prepares data and ensures quality for building ML models. Data preparation consists of preliminary stage data cleaning, which involves detecting duplicates, violations of integrity constraints, missing values, etc.

Human expertise is required in data cleaning, specifically in error detection, repair, validation, and specification (Rezig, E.K. et al., 2019). In data analysis, effective visualization plays a significant role. Most of the time, data is multidimensional and needs to be converted into 2D or 3D for effective visualization. Principal Component Analysis (PCA), t-distributed Stochastic Neighbour Embedding (t-SNE), Linear Discriminant Analysis (LDA), and Normal Discriminant Analysis (NDA) help to convert data from multidimensional to 2D or 3D. Python users use Matplotlib and the Seaborn library for visualization purposes. Univariate, bivariate, and multivariate analyses need to be carried out based on dimensionality. Scatter plots, pair plots, box plots, violin plots, distribution plots, joint plots, bar charts, and line plots are used for data interpretation and analysis.

2.3. ML Algorithms, Optimization and Evaluation

ML algorithms are classified as supervised, unsupervised, semi-supervised, and reinforcement learning. Linear Regression (LR), logistic regression, decision trees, SVM, Naive Bayes, KNN, K-means, random forests, dimensionality reduction, the gradient boosting algorithm, and Ada-Boosting are some of the commonly used algorithms by ML learners. Ensemble algorithms are used when massive loads of data have to be handled to make predictions with high accuracy. Some of the ensemble techniques are bagging or bootstrapping aggregates, boosting stacking classifiers, and voting classifiers. These techniques combine the predictive power of several base estimators to improve robustness. Also, it boosts the performance of the model, but at the same time, we have to compromise with the interpretability of the model.

In ML experimentation for finite sample data with ground truth available, an overfitting problem arises that degrades the generalized performance of the model (Cawley, G. C et al., 2010). Over fitting can be controlled by using regularization, early stopping, or hyper-parameter averaging. Resampling strategies help in model selection, accuracy assessment, and tuning of hyper-parameters (Merghadi A. et al., 2020). The ensemble method, after hyperparameters tuning and cross-validation, improves model performance (Kotthoff, L et al., 2019). The appropriate way of performing model evaluation, model selection, and algorithm selection techniques is an important task in ML-based project execution. On a given data set, data splitting is the first step. Splitting may be 60/40, 70/30, or 80/20. For a large dataset, it may be 90/10. The application of an appropriate ML algorithm is required. The next step that needs to be taken is the validation of unseen data. The following stage is general accuracy estimation and improvement. The bootstrapping technique is used to estimate the uncertainty of performance. Leave-one-out cross-validation and k-fold cross-validation were used to improve the performance of the model.

In model evaluation, sometimes it needs to estimate the generalization of performance and, for more precision, increase the predictive performance by tweaking the learning algorithm and selecting the best-performing model from a given hypothesis space or identifying the ML algorithm that is best suited for the problem. Holdout and bootstrapping techniques are used to estimate a model’s generalization performance. Hyper-parameters help to control the behaviour of ML algorithms when optimizing for performance, finding the right balance between bias and variance.

Holdout shows better results in model evaluation for large data sets. For hyper-parameters optimization, leave-one-out cross-validation is a good option when working with small sample sizes (Raschka, S. 2018). As an ML beginner, first, check whether the data is large or small (here, large means more than 1000 data sets). In the small data set, to avoid overfitting and to generalize performance, k-fold cross-validation with a larger value of k is suitable, or one may use leave-one-out cross-validation. Confusion matrix, accuracy, precision, F1 score, recall, ROC, AUC, Jaccard index, Mathew’s correlation coefficient, Kappa index, and log loss are the evaluation metrics used for model evaluation (Wang J et al., 2020). A few important metrics generally used to compare the performance of the model to other models are as follows:

2.3.1. Accuracy

It is determined as the proportion of samples that were correctly categorized in relation to all samples, as shown in Equation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

2.3.2. Precision

Precision is the proportion of a class of samples that a model properly predicts to the total number of samples in that class, as shown in Equation (2).

$$\Pr ecision = \frac{TP}{TP + FP} \tag{2}$$

2.3.3. Recall

Precision is the proportion of a class of samples that a model properly predicts to the total number of samples in that class, as shown in Equation (3).

$$\operatorname{Re} call = \frac{TP}{TP + FN} \tag{3}$$

Process flow of ML,

1. Understanding of data and pre-processing
2. Split into training and testing data set
3. Evaluate the model
4. Use hyper-parameter tuning
5. Find the best hyper-parameters values
6. Check the performance of the model
7. Check for the generalized performance of the model with unseen data
8. Use model for real-world problem solving

Process flow describes the complete process flow for application design in Ml. In Section 3, the case study of diabetic detection is explained based on an algorithm. Section 3 gives complete experimentation details of the healthcare used case. Different analytics, evaluation criteria, and verification strategies help learners understand in a better way.

3. Methodology with Diabetes Data Set Application Case

3.1. Data Understanding

A classroom-assisted case study was carried out on a diabetic database originating from the National Institute of Diabetes and Digestive and Kidney Diseases, available on Kaggle. In an ML paradigm, the first learner needs to determine the authentic source of data. The next immediate action the learner should take is to understand the data, followed by Exploratory Data Analysis (EDA).

Table 3. Attribute description

Attribute	Details	Mean	Standard Deviation
Pregnancies	Number of Times Pregnant	4.4	2.98
Glucose	Oral Glucose Tolerance Test (Glucose Concentration at 2 Hours)	121.68	30.43
Blood Pressure	Blood Pressure (mmHg)	72.25	12.11
Skin Thickness	Triceps (mm)	26.6	9.63
Insulin	2-Hour Serum Insulin (µu/ml)	93.08	14
BMI	Body Mass Index (Weight in Kg / (Height in inches))^2	6.87	18.2
Diabetic Pedigree function	Diabetic Pedigree Function	0.33	0.078
Age	In Years	0.4769	0

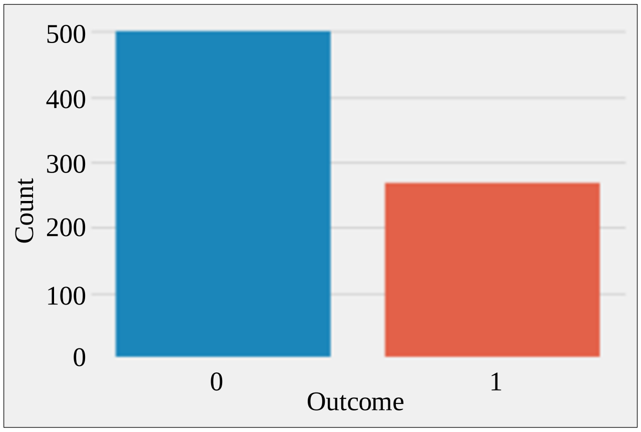


Fig. 1 Frequency count of diabetic and non-diabetic patients

The diabetic database consists of several medical predictor variables and one target variable, outcome. Predictor variables include the number of pregnancies the patient has had, their BMI, insulin level, age, glucose, blood pressure, skin thickness, 2-hour serum insulin, diabetes pedigree function, age, etc. The output label is an outcome that is 0 or 1. Figure 1 shows the frequency count of diabetic and non-diabetic patients determined from the output label. Table 3 describes the attributes available in the data set.

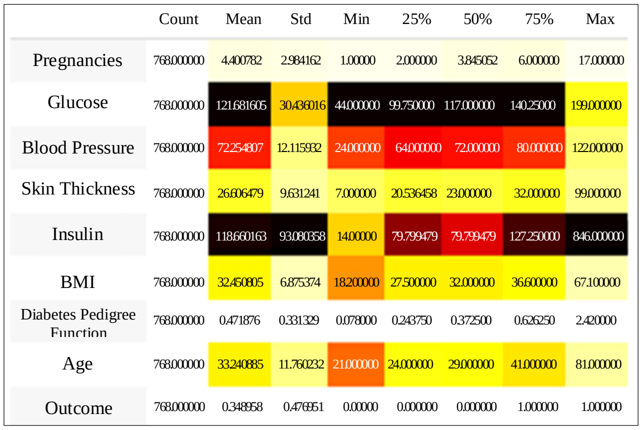
3.2. Data Pre-Processing

Outlier rejection (P), missing value treatment (Q), standardization (R), and feature selection (F) are all parts of the data pre-processing procedure (Hasan M. K et al., 2020). In this case, the output has only two outputs: yes or no; hence, it is a binary classification problem. The learner understands that the data belongs to the supervised classification category and works on classification algorithms.

The learner is ready to do EDA after gaining preliminary data understanding. The first step of data cleaning is critical in EDA. In the data cleaning step, first, remove all the missing values in the given data frame. Then, look for data types that can be used to convert any non-numeric value to a numeric value.

The missing values, null values and values equal to zero for the predictor variables need to be identified in the dataset. It is determined as the proportion of samples that were correctly categorized in relation to all samples. This complete process is called data pre-processing. It is observed that the mean value of a few parameters is zero, which needs to be corrected. It would be good to change the zero value of each feature to another value.

The proportion of zero values in each feature is as follows: in pregnancy cases 111, the percent is 14.45 %; in blood pressure cases 35, the percent is 4.56 %; in skin cases 227, the percent is 29.56 %; in insulin cases 374, the percent is 48.70 %; in BMI cases 11, the percent is 1.43 %. These ratios of the value of zero in the skin thickness and insulin features seem high and need to be changed.



	Count	Mean	Std	Min	25%	50%	75%	Max
Pregnancies	768.000000	4.400782	2.984162	1.00000	2.000000	3.845052	6.000000	17.000000
Glucose	768.000000	121.681605	30.436016	44.000000	99.750000	117.000000	140.25000	199.000000
Blood Pressure	768.000000	72.254807	12.115932	24.000000	64.000000	72.000000	80.000000	122.000000
Skin Thickness	768.000000	26.606479	9.631241	7.000000	20.536458	23.000000	32.000000	99.000000
Insulin	768.000000	118.660163	93.080358	14.00000	79.799479	79.799479	127.250000	846.000000
BMI	768.000000	32.450805	6.875374	18.200000	27.500000	32.000000	36.600000	67.100000
Diabetes Pedigree Function	768.000000	0.471876	0.331329	0.078000	0.243750	0.372500	0.626250	2.420000
Age	768.000000	33.240885	11.760232	21.000000	24.000000	29.000000	41.000000	81.000000
Outcome	768.000000	0.348958	0.476951	0.00000	0.000000	0.000000	1.000000	1.000000

Fig. 2 Statistical information obtained after linear scaling and standard scaling

However, a zero value may be meaningful to the corresponding feature and need an expert with expertise in diabetes. After removing the zero value of each feature, the distribution is similar to the normal distribution. Therefore, perform linear scaling and standard scaling. Figure 2 shows the zero values of each feature converted to mean values; some features have a one-sided shape. Therefore, we decided to perform nonlinear scaling and decided to use the quantile transformer, which changes the distribution closest to the normal distribution.

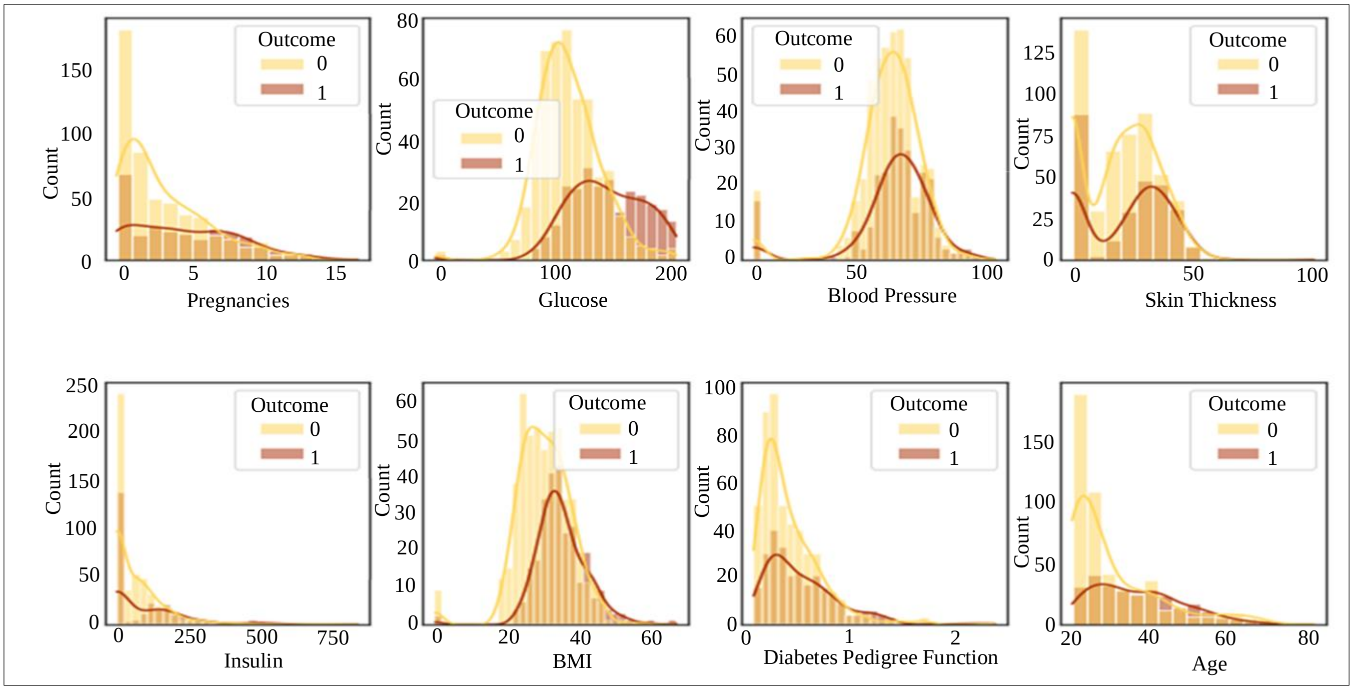


Fig. 3 Feature frequencies

After ensuring that the data is clean and appropriate for use (using the df.info () command), one must proceed with EDA. For data visualization and analysis, various types of graphs, such as bar graphs, density plots, violin plots, box plots, etc., are frequently employed. A bar graph of several features is displayed in Figure 3. It displays the number of observation frequencies present during particular intervals.

Model building and evaluation are done prior to feature extraction. Extreme parameter values that overfit the training data will be penalized by regularization (Hasan M. K et al., 2020). A high value of C instructs the algorithm to prioritize the training set of data. A lower value of C will mean that the model prioritizes complexity over data fitting. Equation (4) can be used to estimate the performance matrix for any model.

$$M = \frac{1}{k} \times P_n \pm \sqrt{\frac{\sum_{n-1}^{k}(P_n - \tilde{P})^2}{k-1}}) \qquad (4)$$

Where,
M : Performance metric,
P_n : Performance metric of each fold,
K : Number of folds

In order to overcome overfitting and to make the model more generalized, model design was carried out with the best hyper-parameters, as shown in Table 4. Table 4 shows that the accuracy of LR, SVM, and DT decreases, but all ensemble methods show improvement in performance after hyper-parameter tuning.

After data preparation and hyper-parameter tuning, feature selection is carried out. Due to dimensionality, sometimes the feature becomes sparser and causes an overfitting problem by losing generalization capability. A correlation analysis is analyzed to reduce dimensionality, principle component analysis and independent component analysis can be the alternative ways.

3.3. Feature Extraction

Most of the datasets related to healthcare contain noisy data instead of irrelevant or redundant data. Feature selection is used in many application areas as a tool to remove irrelevant and/or redundant features. There is no single feature selection method that can be applied to all applications (Khalid S et al., 2014).

The feature selection method affects the accuracy and overall performance of the algorithm. Feature Extraction means selecting only the important features in order to improve the accuracy of the algorithm. It reduces training time and reduces over-fitting. Here, for analysis purposes, two methods are used Correlation Matrix to identify uncorrelated features and then Random Forest Classifier to obtain important features. The random forest classifier selects the best set of features (Hasegawa, K et al., 2017).

Table 4. Analysis before feature extraction with best hyper parameters

Classifier	Accuracy of Validation Set	Best Parameters	Accuracy with the Best Parameter	Recall with Best Parameter	AUC with the Best Parameter	F1 Score with the Best Parameter	Jaccard Index
LR	0.772	Hyper parameter C=0.1	0.765	0.338	0.653	0.73	0.32
SVM	0.769	Hyper parameter C=10 Gamma = 1 the kernel is poly	0.765	0.467	0.687	0.75	0.39
DT	0.736	Maximum depth=4	0.682	0.677	0.681	0.69	0.41
RF	0.763	No of Estimators=8, Maximum features=4, Number of jobs=3	0.786	0.629	0.745	0.78	0.49
ABOOST	0.746	No of Estimators=8, Learning rate =1	0.786	0.580	0.717	0.76	0.44
GBOOST	0.74	No of Estimators=14, Learning rate =0.1	0.786	0.661	0.753	0.79	0.5

Table 5. Feature importance

Feature	Importance
Glucose	24.2%
BMI	17.25%
Age	13.5%
Diabetes Pedigree Function	12.8%
Blood Pressure	9.2%
Pregnancies	8.6%
Skin Thickness	7.3%
Insulin	6.8%

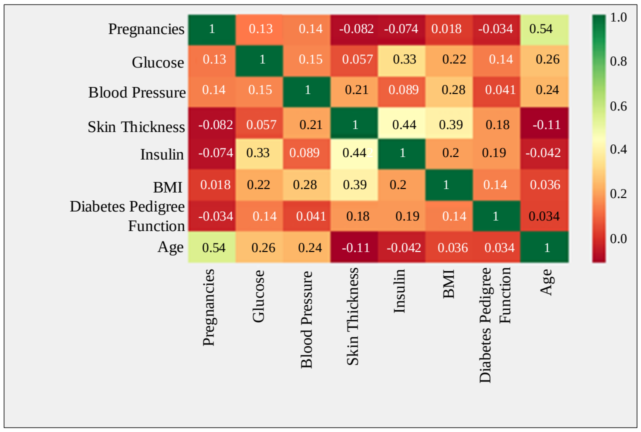


Fig. 4 Correlation matrix to identify uncorrelated features

The traits appear to be uncorrelated in Figure 4. As a result, any features cannot ruled out based solely on the correlation matrix. The following step requires the Random Forest Classifier, which provides the importance of the features as stated in Table 5 and is necessary for important feature selection. Table 5 demonstrates that, as compared to other variables, glucose, BMI, age, and diabetes pedigree function are the most significant.

3.4. Standardization

There can be significant deviations in the data set at times, like in this dataset's BMI, which has 248 distinct values. It is necessary to normalize this significant variance. In order to convert attributes with a Gaussian distribution and varying means and standard deviations to a standard Gaussian distribution with a mean of 0 and a standard deviation of 1, standardization is a useful technique.

3.5. Cross Validation

The data in cross-validation is frequently unbalanced, with numerous instances of class 1 and few instances of other classes. Thus, it becomes necessary to train and test algorithms on each and every instance of the dataset. After that, average out all the accuracy issues throughout the dataset. The dataset is initially divided into k-subsets before the K-Fold Cross Validation is performed. Let us imagine the dataset is divided into (k=5) components. Over the 4 parts, one portion is set aside for the algorithm's testing and training. By altering the testing portion every iteration while training the algorithm over the remaining portions, continue the process. The average of the accuracies and errors gives the algorithm's average accuracy. The term for this is K-Fold Cross-Validation. Occasionally, an algorithm will overfit the data for one training set while underfitting the data for another. Consequently, it may create a generalized model with cross-validation. Table 6 displays how cross-validation, standardization, and feature extraction affect various classifiers.

Table 6. Effect of feature extraction, standardization and cross-validation on classifiers

Classifier	New Accuracy	Accuracy	Effect
Linear SVM	0.78125	0.770633	0.010417
Radial SVM	0.770833	0.765625	0.005208
Logistic Regression	0.776042	0.7725	-0.0035
KNN	0.729167	0.729167	0.000000

3.6. Ensemble Method

To increase the accuracy of the prediction, ensemble approaches mix numerous models once they have been created. Typically, ensemble approaches yield more precise results than a single model would. Base models are the models that are utilized to build these ensemble models (Nilashi M et al., 2022, and Ardabili S. et al., 2020).

Simple yet effective ensemble learning approaches include max voting, averaging, and weighted averaging. Equation (5) represents the ensembling mathematical model.

$$P_i^{en} = \frac{\sum_{j=1}^{m=6}(W_j \times P_{ij})}{\sum_{i=1}^{C=2}\sum_{j=1}^{m=6}(W_j \times P_{ij})} \tag{5}$$

Where,
C=2 indicates whether the patient is diabetic or not,
W_j is the weight corresponding AUC of that jth classifier and $P \in [0,1]$ is the confidence value.

Utilized in the advanced ensemble method are boosting, blending, bagging, and stacking. By stacking, a new model is created by combining several existing ones. In blending, the train set and validation set are split, and the validation set makes the prediction. Bagging mixes the output of various models to provide more universal outcomes. A sequential process known as “boosting” involves trying to fix mistakes made by earlier models. The bagging algorithms are the bagging meta-estimator and Random forests. The boosting algorithms used in machine learning are AdaBoost, GBM, XGBM, Light GBM, and CatBoost. Table 7 demonstrates that ensemble approaches outperform basic classifiers in terms of accuracy.

One of the easiest ways to combine predictions from many machine learning algorithms is by voting. From your training dataset, it first builds two or more standalone models. When requested to produce predictions for new data, models can then be wrapped by a voting classifier, which will average the sub-model predictions.

Table 7. Model evaluation after feature extraction

		Before feature selection					After feature Selection				
Classifier	**Best Parameters**	**Accuracy**	**Recall**	**AUC**	**F1 Score**	**Jaccard Index**	**Accuracy**	**Recall**	**AUC**	**F1 Score**	**Jaccard Index**
LR	Hyperparameter c=0.1	0.765	0.338	0.653	0.73	0.32	0.73	0.27	0.618	0.69	0.25
SVM	Hyperparameter c=10 Gamma is: 1 the kernel is: poly	0.765	0.467	0.687	0.75	0.39	0.765	0.51	0.70	0.76	0.42
DT	Maximum depth=4	0.682	0.677	0.681	0.69	0.41	0.75	0.612	0.714	0.75	0.44
RF	No of Estimators=8, Maximum features=4, number of jobs=3	0.786	0.629	0.745	0.78	0.49	0.77	0.58	0.72	0.77	0.46
ADBOOST	No of Estimators=8, Learning rate =1	0.786	0.580	0.717	0.76	0.44	0.77	0.58	0.72	0.77	0.46
GBOOST	No of Estimators=14, Learning rate =0.1	0.786	0.661	0.753	0.79	0.5	0.77	0.45	0.675	0.74	0.37

A weighted voting classifier will be used. The classifiers will be distributed based on how accurate they are. The classifier with the highest accuracy will, therefore, be given the most weight, and so on. Voting Classifier (VC) is a sort of cooperative learning that involves combining the predictions of many classifiers in order to achieve higher performance than a single classifier (Y. Zhang et al., 2014, Yousaf A. et al., 2020, and Trivedi S et al., 2021). In our analysis, Random Forest, gradient boost, and extra tree classifier are used for voting using Algorithm 1 given below.

Algorithm 1 Voting Classifier:

```
Input $data(x, y)_{i=1}^{N}$
    $T_{RF} = Trained\_RF$
    $T_{GB} = Trained\_GB$
    $T_{ET} = Trained\_ET$
    for $i=1$ To M  do
    If
      $T_{RF} \neq 0 \& T_{GB} \neq 0 \& T_{ET} \neq 0 \& training \quad set \neq 0 \quad then$
      $\Pr obRF\_Pos = T_{RF}.probability(pos-class)$
      $\Pr obRF\_Neg = T_{RF}.probability(Neg-class)$
      $\Pr obGB\_Pos = T_{GB}.probability(pos-class)$
      $\Pr obGB\_Neg = T_{GB}.probability(Neg-class)$
      $\Pr obET\_Pos = T_{ET}.probability(pos-class)$
      $\Pr obET\_Neg = T_{ET}.probability(Neg-class)$
      $Decision = \max(\frac{1}{N_{Classifier}} \sum_{Classiifer} (Avg(\Pr obRF - \Pr obGB - \Pr obET - Pos),$
                  $Avg(\Pr obRF - \Pr obGB - \Pr obET - Pos)))$
      endif
    Return final label
End for
```

3.7. Voting Classifier

Table 8 shows voting classifier gives the best accuracy of 79.16%. Figure 5 shows performance of the voting classifier is best among other classifiers used. An empirical study demonstrates the importance of data interpretation, pre-processing, and domain expertise for creating a strong prediction model. Making important judgments about the design and evaluation of models is aided by statistical analysis of data.

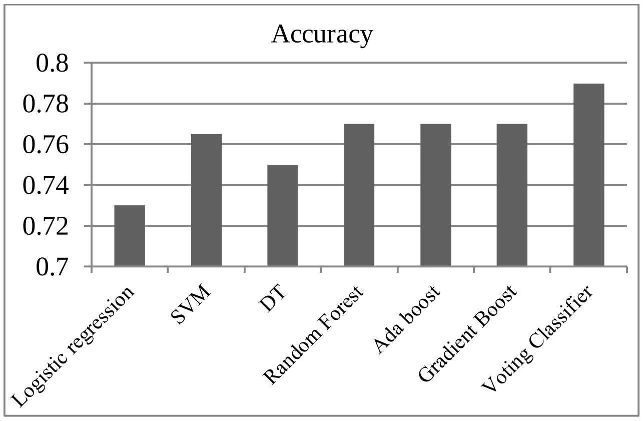


Fig. 5 Performance comparison of different classifiers

Cross-validation, standardization, and hyper-parameters tuning all aid in model generalization and enhance prediction precision. A voting classification ensemble method ensures that the model is more accurate. Following this case, the study will ensure lifetime learning and give the learner the ability to create and apply cases. In section IV, learner opinions are presented following the conclusion of this case study.

Table 8. Voting classifier

Classifier	Accuracy
Voting classifier (Random forest, gradient boosting, Extra tree Classifier)	79.167%

4. Learners Behavior Synthesis

As seen in Figure 6, for capstone project development, lots of data processing and ML algorithm steps are required. The learner must understand all the steps first, and then he can enter into a loop of other use case development. In the initial stage of this activity, some challenging interfaces were observed. First, it was observed that many students from non-programming fields suffer from programming fear. The student also had concerns about how to get started, what are the typical procedures, how to use which algorithms, how to evaluate, and how to determine the success rate.

In a guided case study, each question was answered individually using the knowledge of the resources at hand. After the final study, a new use case was handed over to the user, and the results were analyzed. After the case study, an unknown data set was given to the user and collected feedback with a questioner set in Table 9 to check the overall learning impact. It is observed that 34% of students were able to handle complicated engineering problems with a higher impact, according to feedback from the students gathered using the questioner in Table 9.

Figure 6 shows a cognitive approach learning methodology flow graph. In the activity submission process, it was observed that 100% of students understood project development. 28% of Students showed outstanding performance in implementation, considering all evaluation parameters. Students understood the subject utility for society and all bloom's levels of learning. 34% of students showed the capability to solve complex engineering problems with a higher impact level, 27 % were capable of executing with a moderate impact level, 31% were capable of performing with a normal impact level and 7% with a lower impact level provided in the subject area.

Table 10 shows the research problems of this study. After the successful implementation of the new use case as a part of the final assessment, the learner's submitted reports were analyzed. This shows that except for 14% of the learners, the remaining were confident in solving any unknown medical data using ML.

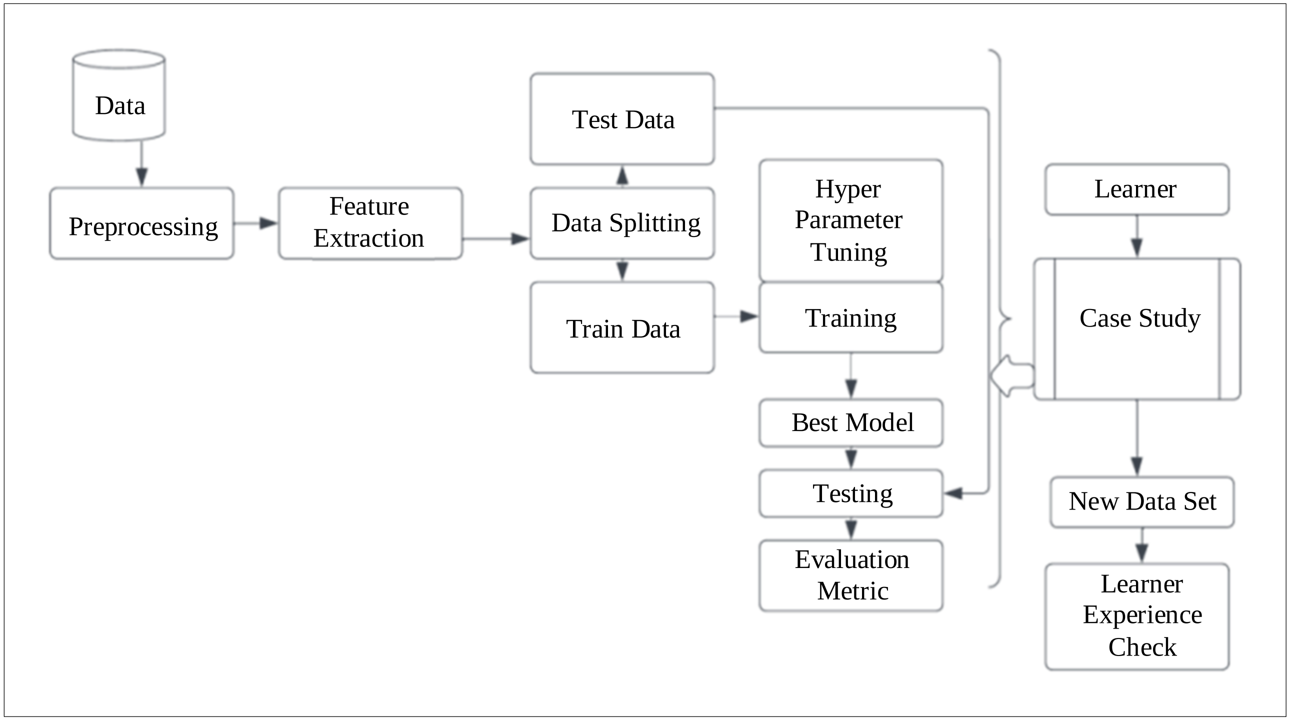


Fig. 6 Process flow

Table 9. Questioner of feedback

Sr. No.	Questions	Impact
1	How confident are you in understanding and applying different algorithms to different use cases to evaluate the models?	Understanding the type of use case and ML algorithm in depth as per utility concern.
2	How confident are you in designing, implementing, analyzing and demonstrating different use cases to evaluate the performance of the models?	Understanding the way of data interpretation and its analysis.
3	How confident are you in designing the development of application models using supervised and unsupervised learning algorithms?	Learner's ability to handle unknown use cases. Thus learning impact of the activity is evaluated.
4	How confident are you in Comparing different machine learning techniques and demonstrating the comprehension of the trade-offs involved in design choices?	Learner's ability to draw design conclusions. The facilitator knows the preparedness level of the learner.

Table 10. Research problem analysis

Sr. No.	Research Problem	Method Adopted to Solve
1	How to create a learning interest in ML?	Use case studies using standard datasets available freely on different community platforms.
2	What are the standard learning methods?	The procedure adopted in Section III.
3	How to prove Learning compliance?	The learning experience is accounted for in Section IV.
4	How to remove the programming phobia of non-programmable learners?	New use case Analysis completed by Learner.

5. Conclusion

Project case studies and implementation play a crucial role in improving learning in tech-oriented subjects. A case study-driven guided project design technique in ML helps the student comprehend the subject better. The learners understand the basics of ML model evaluation, optimization, and ensemble approaches. The learner was familiar with reliable data sources and associated analysis jargon. The student is familiar with different intermediate steps, process flow, and legitimate conclusion dragging. Voting classifiers take the best prediction and average it. Confidence in design and development is developed through a fully guided approach. The learner is aware of how to initiate and complete capstone projects in the relevant field. This strategy can successfully drive lifelong learning for any course. In the classroom, a medical data set is taught using an empirical method based on case studies. Following this study, student's behaviour was examined, and it revealed that 34% of learners felt comfortable building any untested application, and 28% started choosing new challenges with minimum assistance. 31% of students completed new assignments after some review and assistance. 7% of students were able to comprehend the material in its entirety, albeit with some hesitancy because of technological discomfort.

References

[1] Isidro Calvo et al., "A Multidisciplinary PBL Approach for Teaching Industrial Informatics and Robotics in Engineering," *IEEE Transactions on Education*, vol. 61, no. 1, pp. 21-28, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] D.A. Umphress, T.D. Hendrix, and J.H. Cross, "Software Process in the Classroom: the Capstone Project Experience," *IEEE Software*, vol. 19, no. 5, pp. 78-81, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[3] Rodrigo Pessoa Medeiros, Geber Lisboa Ramalho, and Taciana Pontual Falcao, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," *IEEE Transactions on Education*, vol. 62, no. 2, pp. 77-90, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Jack W. Smith et al., "Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus," *Proceedings of the Symposium on Computer Applications and Medical Care*, pp. 261-265, 1988. [Google Scholar] [Publisher Link]

[5] Timnit Gebru et al., "Datasheets for Datasets," *Communications of the ACM*, vol. 64, no. 12, pp. 86-92, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Margaret Mitchell et al., "Model Cards for Model Reporting," *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 220-229, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Karen L. Boyd, "Datasheets for Datasets Help ML Engineers Notice and Understand Ethical Issues in Training Data," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, pp. 1-27, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Kasia S. Chmielinski et al., "The Dataset Nutrition Label (2nd Gen): Leveraging Context to Mitigate Harms in Artificial Intelligence," *arXiv preprint arXiv:2201.03954*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ozlem Yavanoglu, and Murat Aydos, "A Review of Cyber Security Datasets for Machine Learning Algorithms," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, pp. 2186-2193, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] Guillaume Lemaitre, Fernando Nogueira, and Christos K. Aridas, "Imbalanced-Learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 1-5, 2017. [Google Scholar] [Publisher Link]

[11] Hima Patel et al., "Advances in Exploratory Data Analysis, Visualization and Quality for Data-Centric AI Systems," *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4814-4815, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] El Kindi Rezig et al., "Towards an End-to-End Human-Centric Data Cleaning Framework," *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pp. 1-7, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Gavin C. Cawley, and Nicola L.C. Talbot, "On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079-2107, 2010. [Google Scholar] [Publisher Link]

[14] Abdelaziz Merghadi et al., "Machine Learning Methods for Landslide Susceptibility Studies: A Comparative Overview of Algorithm Performance," *Earth-Science Reviews*, vol. 207, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Lars Kotthoff et al., "Auto-WEKA 2.0: Automatic Model Selection and Hyperparameter Optimization in WEKA," *Journal of Machine Learning Research*, vol. 18, pp. 1-5, 2017. [Google Scholar] [Publisher Link]

[16] Sebastian Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," *arXiv preprint arXiv:1811.12808*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Jingwen Wang et al., "A Survey on Trust Evaluation Based on Machine Learning," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1-36, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Md. Kamrul Hasan et al., "Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers," *IEEE Access*, vol. 8, pp. 76516-76531, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Samina Khalid, Tehmina Khalil, and Shamila Nasreen, “A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning,” *2014 Science and Information Conference*, London, UK, pp. 372-378, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[20] Kento Hasegawa, Masao Yanagisawa, and Nozomu Togawa, “Trojan-Feature Extraction at Gate-Level Netlists and Its Application to Hardware-Trojan Detection Using Random Forest Classifier,” *2017 IEEE International Symposium on Circuits and Systems*, Baltimore, MD, USA, pp. 1-4, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Mehrbakhsh Nilashi et al., “Predicting Parkinson’s Disease Progression: Evaluation of Ensemble Methods in Machine Learning,” *Journal of Healthcare Engineering*, vol. 2022, pp. 1-17, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Sina Ardabili, Amir Mosavi, and Annamária R. Várkonyi-Kóczy, “Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods,” *18th International Conference on Global Research and Education*, vol. 101, pp. 215-227, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[23] Yong Zhang et al., ‘‘A Weighted Voting Classifier Based on Differential Evolution,” *Abstract and Applied Analysis*, vol. 2014, pp. 1-6, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[24] Anam Yousaf et al., “Emotion Recognition by Textual Tweets Classification Using the Voting Classifier (LR-SGD),” *IEEE Access*, vol. 9, pp. 6286-6295, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Sandeep Trivedi, and Nikhil Patel, “The Determinants of AI Adoption in Healthcare: Evidence from Voting and Stacking Classifiers,” *ResearchBerg Review of Science and Technology*, vol. 1, no. 1, pp. 69-83, 2021. [Google Scholar] [Publisher Link]