Advanced Simulation and Modelling

Research paper

PROJECT TOPIC:

IMPLEMENTATION AND COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS

Dated: 14th July,2024

Instructor:

INSTRUCTOR NAME

Table of Contents

Implementation And Comparative Analysis Of Different Machine Learning Models

***Abstract*— This paper presents a comprehensive comparative analysis of multiple machine learning models applied to a "healthcare-dataset-stroke-data" dataset. The models evaluated include Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree Classifier, Random Forest Classifier, and Artificial Neural Network (ANN). The primary objective is to rigorously assess and compare the performance of these models using a variety of evaluation metrics, including accuracy, precision, recall, F1-score, and training time. The results indicate that the Random Forest Classifier achieves the highest accuracy, followed closely by the Decision Tree Classifier and ANN, demonstrating their effectiveness for the given dataset. Conversely, Logistic Regression exhibited the lowest performance. This analysis provides valuable insights into the strengths and limitations of each model, highlighting the Random Forest Classifier as a robust choice for predictive tasks. The implications of these findings are discussed, offering guidance for model selection in similar predictive modeling applications.**

I. INTRODUCTION

While a great deal of progress has been made in machine learning, it has meant that many advanced algorithms have been created to make correct predictions and provide valuable insights about complicated patterns present in data. The paper conducts an exhaustive comparison analysis among a large number of supervised machine learning models that are normally used in the determination of which model has the best effectiveness for particular tasks in forecasting. Comparing analyses like this one is absolutely crucial in order to improve the utility and accuracy of data analysis approaches in a wide field of applications.

Due to their capability to learn patterns from inputs to outputs, supervised learning models turn out to be very efficient in classification and regression problems. These models are trained on labeled data with known outcomes. Using a typical dataset, this study will undertake the performance of several different classification methods that include logistic regression, k-nearest neighbors (KNN), Decision Tree Classifier, Random Forest Classifier, and Artificial Neural Network (ANN). To evaluate each of the models, different measures have been used, such as accuracy, precision, recall, F1-score, and training length.

The most important goal is the identification of the best performing model concerning a prediction and understanding the trade-offs connected with individual models. By systematic comparisons of the models, this work offers helpful insights into model selection and therefore supports practitioners in choosing the algorithm which best fits their particular predictive modeling needs.

II. LITERATURE REVIEW

Much work has been done in the literature to discuss the performance and characteristics of various machine learning models; each of these models has certain strengths and possible weaknesses. Logistic Regression is known to be simple and interpretable. Being a linear model, it does well when the relationship between input features and the target variable is roughly linear **[2]**. However, complex and nonlinear data structures easily nullify its effectiveness.

K-Nearest Neighbors is a non-parametric method which predicts the output by taking a majority vote of neighbors within a certain radius from the query point **[3]**. Its ease and efficiency in classification problems arise because of its simplicity and freedom from any assumptions about the distribution of data. However, KNN can be computationally expensive on very big datasets as it involves computation of distance to all training samples for every new prediction.

Decision Trees are amongst the powerful models that partition data into subsets based on the values of features. This forms a tree-like model of decisions. While capable of capturing intricate decision boundaries, Decision Trees have the tendency to overfit in case they are grown too deep, thus generally losing out on generalizability **[4]**.

Random Forest is an ensemble learning technique where several decision trees are produced in training and their predictions are combined. This turns into an enhanced accuracy that decreases the variance of the model **[5]**. It also makes the Random Forests robust concerning noise and outliers in the data.

Artificial Neural Networks take inspiration from the human brain, in which it connects neurons that process information along several layers. In fact, ANNs are extremely flexible and can model truly complex nonlinear relationships. This makes them applicable in a wide range of applications, from computer vision to natural language processing **[1]**. On the contrary, training requires great amounts of data and computational resources **[6]**.

In this paper, we compare these models Logistic Regression, KNN, Decision Tree Classifier, Random Forest Classifier, and ANN based on a standard dataset. Our analysis focuses on the main performance metrics, namely, accuracy, precision, recall, F1-score, and time of training that will give us an all-rounded comparison and, furthermore, model selection for the said predictive tasks.

III. METHODOLOGY

This section details the dataset used, the preprocessing steps, and the implementation of each machine learning model:

A. DATA EXPLORATION AND UNDERSTANDING

I. LOADING AND READING THE DATASET

The dataset used for this study is retrieved from the healthcare dataset on stroke prediction. Data is loaded into a Pandas DataFrame for analysis and pre-processing. The initial inspection of the dataset includes showing the first few rows in the overview of the structure of the data and feature types.

II. DATASET INFORMATION

To understand the dataset's structure, let's have a quick view of the structure of this dataset by calling the info function, which shows data types and the number of non-null values for each column. This shows that there are 12 columns and 5110 entries, some columns containing missing values.

III. DESCRIPTIVE STATISTICS

Numerical columns fill the description of statistics, which provide insight into the central tendency and dispersion, distribution of data. These include count, mean, standard deviation, minimum, maximum, and quartile values for features relating to age, hypertension, heart disease, average glucose level, BMI, and stroke.

IV. DATA DIMENSIONS

The shape of the data frame indicates the number of rows and columns, confirming that the dataset consists of 5110 rows and 12 columns.

V. MISSING VALUES ANALYSIS

The critical step in the preprocessing of data involves the identification of the missing values. Consequently, the analysis indicates 201 missing values in the column showing BMI, while the remaining columns are complete. This missing value requires treatment to complete the integrity of the analysis.

VI. DUPLICATE ROWS

Identifying duplicate rows would help to avoid redundancy and any potential biases in the analysis. The dataset is verified for duplicates; no duplicate rows are there in the dataset.

VII. DATA TYPES

The data type of each column will help in selecting appropriate preprocessing techniques. A mix of data types for integer, float, and object data types is expected in the dataset.

VIII. CROSS-TABULATION ANALYSIS

Cross-tabulations (contingency tables) are generated to explore relationships between categorical variables:

1. Type of Work and Gender: This table tries to map out various work types across gender, bringing out trends such as higher employment in the private sector for males and females.

2. Gender and Marital status: The table represents the distribution of marital status against the gender that forms it. It represents the demographic pattern.

*B. **ANALYSIS AND VISUALIZATION***

This section is dedicated to the detailed analysis and visualization of the dataset, whereby it points out the key features and distributions for comprehensive understanding. Further, this section will be explained under the following subsections: gender distribution, marital status, distribution with respect to BMI and age, work type, heart disease, hypertension, residence type, smoking status, average glucose levels, and feature correlations. These visualizations are important in outlining pattern, relationship, and possible problems in the dataset that might influence machine learning model performance.

I. GENDER DISTRIBUTION

A pie chart of the gender distribution indicates that females make up 58.6%, males 41.4%, and a negligible number fall under the 'Other' category. This will show the demographic split, which could influence certain health outcomes.

II. COUNT OF GENDER WITH 'EVER MARRIED' CONDITION

A bar plot of the frequency count of married males and females. This plot shows that more females have been married, as compared to males. The accuracy of such demographic insights is very important in understanding marital status distributions, which can correlate with health outcomes.

III. HISTOGRAMS FOR BMI AND AGE

The distribution for BMI and age are better understood through the dispersion and central tendencies as brought out in the histograms. The distribution for the case of BMI concentrates around the mean value with a couple of outliers, while the age covers a wide range, showing the diversity in age among participants.

IV. DISTRIBUTION OF WORK TYPE

A bar chart of the types of work for the participants shows the majority working in the private sector, followed by self-employment. This will be useful in interpreting occupational patterns, which may relate to health outcomes.

V. DISTRIBUTION OF HEART DISEASE

A bar chart shows that the majority of the dataset do not suffer from heart disease, with only 5.4% having heart disease. This is important to consider when developing models since there is a class imbalance problem that might affect the model performance to generalize well on different classes.

VI. DISTRIBUTION OF HYPERTENSION

Just like heart disease, the balance in the dataset concerning hypertension is 90.25% No, which is highly imbalanced, and if not taken care of during training, there are high chances of biases in the model predictions.

VII. DISTRIBUTION OF RESIDENCE TYPE

The distribution of residence types is an almost equal split between urban and rural, according to the bar chart. This is necessary in order for the model to be balanced and not biased towards any type of residence.

VIII. DISTRIBUTION OF SMOKING STATUS

The distribution of the status of smoking is visualized, indicating various categories such as 'formerly smoked', 'never smoked' and 'smokes'. Understanding these distributions aids in the analysis of smoking and its impact on health outcomes-for example, stroke.

IX. DISTRIBUTION OF AVERAGE GLUCOSE LEVEL

A histogram with a KDE plot of average glucose levels is instructive about the central tendency and spread of glucose levels across participants. This kind of representation is important in understanding the range of glucose levels, which is an important health indicator.

X. CORRELATION MATRIX

The heat map of the correlation matrix shows interaction between numerical features of age, hypertension, heart disease, average glucose level, BMI, and stroke. Strength and direction of the correlations indicate that there is a strong relationship between age and stroke. This insight is essential in understanding which features are most influential in making predictions about health outcomes.

XI. DISTRIBUTION OF THE TARGET COLUMN (STROKE)

A bar chart of the target variable 'stroke' shows high class imbalance, with 95.1% of the entries without a stroke. The presence of this skew would require extra attention at the time of training the model since it might end up biased toward the majority class.

These visualizations and analyses will give insight into the dataset in detail. It reveals some important patterns, potential imbalances of classes, and relations between features. This deep exploration is going to lay grounds for preprocessing the data effectively and model implementation in a way that provides robust and reliable predictions.

Therefore, the identification of imbalances and relationships between features through a visual analysis remains fundamental for the development of a correct and generalizable model of machine learning for stroke prediction. This will then give valuable insights to be followed in the next steps for model training, evaluation, and comparison.

*C. **DATA PRE-PROCESSING AND CLEANING***

Data pre-processing and cleaning are the main effective steps required in preparing a dataset to train a machine learning model. In this sense, the current section elaborates on the procedures undertaken in preparing a healthcare dataset for predicting stroke incidents, ensuring the cleanliness and proper structuring of data for analysis **[8]**.

I. DEALING WITH SINGLE VALUE IN GENDER COLUMN

Since this dataset contains only one instance of gender marked as 'Other', that one entry is eliminated to keep the model consistent and correct. This way, any kind of bias or untruthfulness because of under-representation of this class will be in check.

II. HANDLING MISSING VALUES

Any missing values in the column of BMI were replaced by the mean BMI of the non-missing ones. That is key, since this method of imputation maintains the overall distribution of feature BMI while maintaining completeness for any further analysis.

After which, there needs to be a re-inspection of the imputed missing values to ascertain whether there are any remaining missing data in the dataset. One can thus verify that the dataset is complete with respect to further processing.

III. ENCODING CATEGORICAL VARIABLES

To enable the machine learning models to process the categorical data, the following encoding steps are performed:

1. Gender: Replaced with numerical equivalents using the categorical values 'Male' = 1 and 'Female' = 0.
2. Binary Variable: Binary categorical values are replaced with '1' for 'Yes' and '0' for 'No' in columns like 'ever_married' and 'Residence_type'.
3. One-Hot Encoding: The categorical columns 'work_type' and 'smoking_status' will be encoded using one-hot encoding, whereby per category in those variables, new binary columns will be created.

IV. BALANCING THE DATASET

Balancing the Dataset This is especially true for the class imbalances in the target variable 'stroke'. In light of this, the Synthetic Minority Over-sampling Technique has been applied to counterbalance the 'stroke' column. It synthesizes examples from the minority class until a balanced dataset is reached, on which the model is to be trained.

At the end, 'hypertension' and 'heart_disease' columns are manually balanced to ensure no major class imbalances hold for these variables. That method balances in such a way that the sampling of the minority class is in proportion to the majority class, hence creating a more representative balanced dataset.

Then, this dataset is shuffled to mix the classes well and saves the final balanced dataset into a CSV format for future use.

V. LOADING AND READING THE BALANCED DATASET

The balanced dataset is loaded into a Pandas DataFrame to ensure that all preprocessing steps were completed successfully. It has 33,264 entries right now and 17 columns, with all missing values handled and categorical variables encoded.

VI. VISUALIZING THE BALANCED DATASET

Now it is much easier to imagine just how, in the end of the pre-processing step, this balanced data set actually confirms whether the balancing and encoding were good or not. Most important features are 'stroke', 'hypertension', and 'heart_disease' visualizations to make sure the balance among the classes.

1. Stroke : Distribution in column 'stroke' - it clearly shows the presence of roughly equal instances of both the classes.
2. Hypertension: Distribution of 'hypertension' column is plotted in order to check whether classes are balanced or not.
3. Heart Disease: The distribution of the column 'heart_disease' is plotted so as to cross-check whether the balancing is effective or not.

The visualizations confirm that the dataset is well-prepared for subsequent model training and evaluation steps; it hence gives firm ground for accurate and reliable predictive modeling.

Detailed steps in data pre-processing and cleaning will ultimately result in the transformation of the dataset into a well-structured, clean, and balanced format for machine learning model training and evaluation. It will surely mean the models would be trained on high-quality data for more generalizable and accurate predictions.

*D. **FEATURE SELECTION AND ENGINEERING***

Feature selection and engineering are essential steps in a machine learning pipeline that would contribute to enhancing model performance due to the selection of the most relevant

features and their appropriate transformations. In this section, we discuss methodologies and techniques used for the selection and engineering of features to train machine learning models.

I. RECURSIVE FEATURE ELIMINATION (RFE)

This can be done by applying Recursive Feature Elimination to identify which features are most relevant for the model to train on. Recursive Feature Elimination selects features using recursive elimination with regards to the weights provided by an estimator model. In this paper, the used estimator model is Logistic Regression.

1. Procedure:

   - The estimator defines the model to use Logistic Regression.
   - RFE is initialized with the estimator, and set to choose the top 13 features.
   - The RFE is fitted on the dataset, selecting the most important features.

2. Selected Features:

The RFE process identifies the following top 13 features:

- Gender
- Age
- Hypertension
- Heart Disease
- Ever Married
- Residence Type
- Work Type (Never Worked, Private, Self-employed, Children)
- Smoking Status (Formerly Smoked, Never Smoked, Smokes)

These are the most informative features ranked based on the weights of logistic regression and will serve to train the model.

II. ANOVA (ANALYSIS OF VARIANCE)

Another important feature selection method is ANOVA, which is mainly used for numerical features. It analyzes the importance of each feature by comparing the mean of various groups and by estimating the variance between the groups.

1. Procedure:

   - Numerical columns are identified from the dataset.
   - The data is divided into features, represented as X, and the target variable, which will be y.
   - ANOVA feature selection to select the best 13 numeric features.

2. Selected Features:

The ANOVA process identifies the following top 13 features:

- Age
- Hypertension
- Heart Disease
- Average Glucose Level
- Gender
- Ever Married
- Residence Type
- Work Type (Private, Self-employed, Children)
- Smoking Status (Formerly Smoked, Never Smoked, Smokes)

Features in this list are considered more important, based on their statistical significance to a target variable, and will later be useful during model training.

*E. **MODEL IMPLEMENTATION***

I. DATA SPLITTING

This would be done by splitting the dataset into 80% for training and 20% for taking test sets that then can be used to carry out model evaluation. In this case, an application of stratified sampling would thus allow relatively similar class distribution between both the training and test datasets.

II. LOGISTIC REGRESSION

This is done using the class Logistic Regression from Scikit-learn with L2 regularization to prevent overfitting. A Liblinear solver has been created as, at this size of the dataset, that would be efficient **[2]**.

III. K-NEAREST NEIGHBORS (KNN)

In the implementation of this problem, k=5 along with the Euclidean distance metric for class KNeighborsClassifier originating from Scikit-learn was used. To find out the optimum value of k, cross-validation has been carried out by considering an in-between bias and variance trade-off **[3]**.

IV. DECISION TREE CLASSIFIER

It was implemented with the class DecisionTreeClassifier from Scikit-learn, with Gini impurity set as the criterion, hence giving the method of the splitting of the nodes. A maximum depth was set as regularization to prevent overfitting **[4]**.

V. RANDOM FOREST CLASSIFIER

By using from the Scikit-learn library, the class RandomForestClassifier with 100 trees, each is separately trained on the bootstrap samples of the data. Keeping the max features parameter on the square root of the number of features **[5]**.

VI. ARTIFICIAL NEURAL NETWORK (ANN)

One desirable TensorFlow feedforward architecture in construction consists of an input layer, two ReLU-activated hidden layers, and a softmax-activated output layer. Fitting was done using the Adam optimizer based on a loss given by cross-entropy-categorical **[1]**.

IV. RESULT

Each model was evaluated in the test set for its performance based on accuracy as the main metric. Besides this, other metrics-precision, recall, and F1-score-are considered in order to measure the whole performance of models.

I. ACCURACY COMPARISON

The accuracy results for each model were:

- Logistic Regression: 79.53%
- K-Nearest Neighbors: 96.84%
- Artificial Neural Network: 98.54%
- Decision Tree Classifier: 99.22%
- Random Forest Classifier: 99.44%

II. *PRECISION, RECALL, AND F1-SCORE*

- Logistic Regression: Precision: 0.81, Recall: 0.83, F1-Score: 0.82.
- K-Nearest Neighbors: Precision: 0.97, Recall: 0.97, F1-Score: 0.97.
- Artificial Neural Network: Precision: 0.99, Recall: 0.99, F1-Score: 0.99.
- Decision Tree Classifier: Precision: 0.99, Recall: 0.99, F1-Score: 0.99.
- Random Forest Classifier: Precision: 0.99, Recall: 1.00, F1-Score: 0.99.

III. *COMPUTATIONAL EFFICIENCY*

For training time and prediction time, the computational efficiency of each model was considered. It helps to understand the trade-off between model complexity and computer resources needed for training and prediction.

- Logistic Regression: Training time: 0.3967 seconds, Prediction time: 0.0002 seconds
- K-Nearest Neighbors: Training time: 0.3276 seconds, Prediction time: 0.1015 seconds
- Artificial Neural Network: Training time: 1200 seconds (approx.), Prediction time: 0.4 seconds
- Decision Tree Classifier: Training time: 0.1371 seconds, Prediction time: 0.0001 seconds
- Random Forest Classifier: Training time: 3.2073 seconds, Prediction time: 0.002 seconds.

V. DISCUSSION

The comparison shows that for the dataset used, methods like Random Forests, and other complex architectures such as ANN, do better. This much-expected high accuracy of the Random Forest Classifier is due to the fact that it aggregates several decision trees' predictions because of variance reduction, giving a better generalization. The ANN here has shown a clear capacity to learn from the complex structure of this data. A simpler model like logistic regression, while less accurate, enjoys advantages with regard to interpretability and computation. Comparative Accuracy.

I. *SCORE INTERPRETATION OF RESULTS*

- Logistic Regression: This algorithm has lower performance and was expected due to its linearity, not very suitable for complex relations.
- k nearest neighbors: It achieves performance considerably better compared to logistic regression; this means it captures the non-linear dependencies. Of course, its computational cost is greater than that of logistic regression, and for large datasets.
- Artificial Neural Network: With high performance and inferring complex patterns in the data, the model does really well. However, running this is computationally very expensive and it requires a huge dataset to train.
- Decision Tree Classifier: It has a very good performance in modeling complicated boundaries between different decisions; overfitting can be handled using pruning or by defining a maximum depth.
- Random Forest Classifier: I expected it to be so because of the ensembling approach, which is more accurate, and because of the diminished overfitting pruning it does to the trees to generalize better. It also is robust to noisy data.

II. *PRACTICAL IMPLICATIONS*

Which of the machine learning models is chosen depends on the task requirement. If interpretability and speed of deployment are key, then Logistic Regression could be favorable. High accuracy and complex data structures would favor the application of techniques like Random Forests or ANNs.

VI. CONCLUSION

It is followed by a comprehensive comparison between different machine learning models for prediction; the performances are again compared in terms of performance and computational efficiency. Among the results, it can be noticed that most successful models The Random Forest Classifier and neural networks, including the Artificial Neural Network, present high precision and recall F1-score values and can therefore be used even on more complex applications. However, these newer models are much more computationally expensive in training time.

The simpler models in the form of Logistic Regression, K-Nearest Neighbors, and Decision Tree Classifier are far less accurate but work very well because their interpretability is very high, they take much shorter time to train, and predictions are much faster. The salient features discussed above for a classification model make it particularly suited for situations where computing resources are limited, and product deployment must be carried out within the shortest possible time.

The study, therefore, recommends the choice of the right model depending on application needs, considering trade-offs between predictive accuracy and computational efficiency. For example, it would be better to use simpler models in circumstances when the resources are scarce or in applications that demand real-time predictions. For instance, in applications where predictive performance is a key concern and resources are unlimited, it prescribes the use of complex models: ensemble methods and neural networks.

Future studies should be directed toward dealing with more complex datasets and increasing the number of machine learning models in order to validate and extend the results of this study. Such models' performances and efficiency can further be enhanced by techniques such as tuning of hyperparameters, advanced feature engineering, and model optimization. The effect of various data preprocessing techniques on the performance of the models can also be studied to build more robust predictive models.

On the whole, this work underlines the importance of the individual approach to model choice according to performance measures and computational efficiency for the best result in every predictive task.

ACKNOWLEDGMENTS

I, Kabir Baig, hereby declare that the work presented herein is my own work completed without the use of any aids other than those listed. Any material from other sources or works done by others has been given due acknowledgement and listed in the reference section. Sentences or parts of

sentences quoted literally are marked as quotations; identification of other references with regard to the statement and scope of the work is quoted. The work presented herein has not been published or submitted elsewhere for assessment in the same or a similar form. I will retain a copy of this assignment until after the board of examiners has published the results, which i will make available on request.

REFERENCES

[1] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[2] Kleinbaum, D. G., & Klein, M. (2010). Logistic regression: a self-learning text. Springer Science & Business Media.

[3] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[4] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth & Brooks/Cole Advanced Books & Software.

[5] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

[6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[7] Little, R. J., & Rubin, D. B. (2019). Statistical analysis with missing data (Vol. 793). John Wiley & Sons.

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357.

[9] Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PloS One, 10(3), e0118432.

[10] Obermeyer, Z., & Emanuel, E. J. (2016). Predicting the future—big data, machine learning, and clinical medicine. The New England Journal of Medicine, 375(13), 1216-1219.