Moorti, Payal

12/20/24

Data Mining I

Stroke Prediction Analysis

Abstract

The dataset selected for the final project was a stroke prediction dataset. The goal was to implement various classification models to determine whether specific health factors and demographics could effectively predict the likelihood of someone having experienced a stroke. Given the dataset's highly imbalanced nature, with significantly fewer stroke cases compared to non-stroke cases, special approaches were taken to appropriately handle the imbalance. The dataset included a binary class attribute: individuals who had experienced a stroke were labeled as "1," while those who had not were labeled as "0." Since the class attribute was already provided, this project involved supervised learning. To evaluate model performance, the Decision Tree and Random Forest algorithms were implemented using both standard splitting techniques and stratified splitting techniques. The primary metric of interest was the recall score, as it measures the model's ability to correctly classify stroke cases.

Related work

The integration of machine learning techniques in the healthcare industry has significantly improved and streamlined the process for the prediction and diagnosis of diseases. In “Identification and Prediction of Chronic Diseases Using Machine Learning Approach” Alanazi highlights the use of algorithms such as Decision Trees, K-Nearest Neighbors

(KNN), and Convolutional Neural Networks (CNN) to predict chronic diseases. By combining structured data such as demographics and laboratory results with unstructured data such as patient symptoms and doctor consultations, machine learning models can enhance prediction accuracy (Alanazi, 2022).

Class imbalance is a prevalent challenge in healthcare-related datasets, as disease occurrences tend to be rare compared to non-disease cases. Ensemble algorithms such as Random Forest have proven to be effective to combat class imbalance. By combining Random Forest with random sub-sampling to balance classes, Random Forest has outperformed other classification methods such as Support Vector Machines (Khalilia et al., 2011). Additionally, Random Forest's ability to compute feature importance enables the identification of critical variables, such as age and the presence of other health issues, that are highly associated with health diagnosis. This can help improve the recall score of the model which is especially valuable for decreasing the number of false negatives (Khalilia et al., 2011).

Data and Preprocessing

The stroke prediction dataset originally contained twelve features and 5110 patient records. Of these records, 249 cases were positive for stroke occurrence, while 4861 were stroke-free. The dataset included three numeric attributes: patient age, average glucose level, and BMI. Binary attributes consisted of hypertension, heart disease, and stroke status, with stroke serving as the target variable for prediction. Categorical features included gender, marital status (yes/no), work type (private, self-employed, government job, children, never worked), residence type (urban/rural), and smoking status (formerly smoked, never smoked, currently smokes, unknown).

The initial preprocessing step focused on handling missing values, specifically the BMI attribute. Given the class imbalance in the dataset, missing values were handled differently for stroke and non-stroke cases. For non-stroke cases, which composed the majority of the dataset, records with missing BMI values were removed. Missing BMI values for the stroke cases were imputed using age-based averages, where the missing values were replaced with the mean BMI calculated from patients within the same age range. The next preprocessing step involved handling categorical attributes. These were converted to binary features using one-hot encoding (pd.get_dummies()), which created separate columns for each category (GeeksforGeeks, 2023). For example, the "Residence Type" attribute transformed into "Residence_Type_Rural" and "Residence_Type_Urban". The third preprocessing step involved handling the class imbalance. To create a more balanced dataset while preserving all the stroke cases, the non-stroke cases were randomly down sampled to 1500 cases. 1500 cases were randomly selected using data.sample(n=1500, random_state=42) where n specifies the desired sample size, and the random_state helps ensure reproducibility (pandas development team, 2024). Additionally, a new feature was created using the BMI values, categorizing them into standard health ranges: underweight (BMI < 18.5), normal (18.5 ≤ BMI < 25), overweight (25 ≤ BMI < 30), and obese (BMI ≥ 30) (Pace Hospital, n.d.).

Methodology

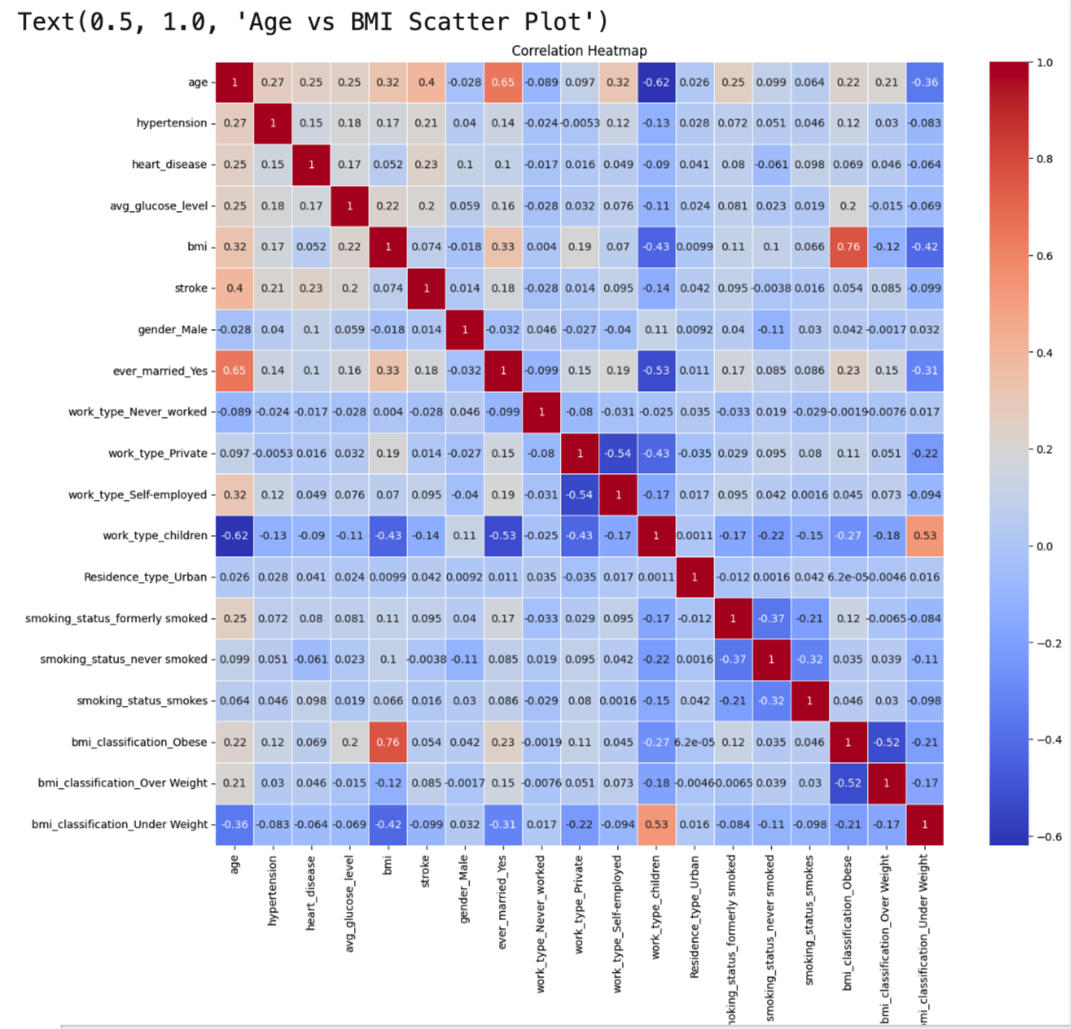
Multiple classification approaches were implemented to evaluate model performance, with special attention paid to recall scores for stroke prediction success. Decision Trees and Random Forest algorithms were chosen for their ability to handle both numerical and categorical features without requiring feature scaling or normalization. Random Forest builds multiple trees using random subsets of data and features, then aggregates their predictions, typically providing better results than a single Decision Tree (Dhillon, 2020). The analysis compared Decision Trees to Random Forest algorithms under different splitting and class weighing strategies. Both the initial Decision Tree implementation and Random Forest implementation used standard splitting with the train_test_split function from scklearn.model_selection (scikit-learn developers, 2024). For the standard splitting method, class imbalance is not taken into account, which led to the models being biased towards the non-stroke cases. The accuracy for the DT model was calculated to be 86.6%, however, the recall was only 13%, indicating a failure in detecting stroke cases. The confusion matrix indicated that only six stroke cases were correctly identified while forty stroke cases were incorrectly classified as non-stroke cases. For the RF model, the accuracy was calculated to be 87.14% while recall was calculated to be only 4%, even lower than the DT model. The confusion matrix indicated that only two stroke cases were correctly detected. Even though the accuracy of both models were relatively high, the detection of stroke cases was extremely low indicating that the models were biased towards the majority class.

When the class-weight parameter was introduced, the recall rate increased significantly. By assigning weights to each class, the model is allowed to pay more attention to the minority class during training (GeeksforGeeks, 2023). The class-weight that was introduced was a one to six ratio, providing more importance to the stroke class. The accuracy for the DT model that utilized the class weight ended up being only 72.57% however, recall increased tremendously to 70% indicating a significant increase in success detecting stroke cases. The confusion matrix for the DT model indicated that 32 stroke cases were correctly classified and only 14 were misclassified as non-stroke. For the RF model, the accuracy was calculated to be 80%, much higher than the recall score of the model that did not incorporate the class weight. 37 stroke cases were correctly classified as stroke cases while only nine cases were missed. The accuracy for this model was 71.71%. The addition of the class weight parameter improved stroke detection but decreased overall model accuracy, demonstrating the tradeoff between accuracy and recall.

Finally, the last approach utilized stratified k-fold cross-validation. This method trains and tests the model on different data subsets while preserving stroke to non-stroke ratio in each fold (Olamendy, 2023). For both RF and DT models, five splits were made, and the classification report and confusion matrix were printed out for each fold. The overall recall, accuracy, recall, and precision over all the folds were then printed by taking the average over each fold. For the DT model, the overall recall was calculated to be 75.57% and the overall accuracy was calculated to be 72.21%. The recall for this method was slightly higher than the regular splitting implementation with the class weight attribute. For the RF model, the overall recall was 79.07%, not too far off from the recall of the RF model that used the class weight attribute with standard splitting methods. The accuracy for this model ended up being 75.24%.

Data Visualizations

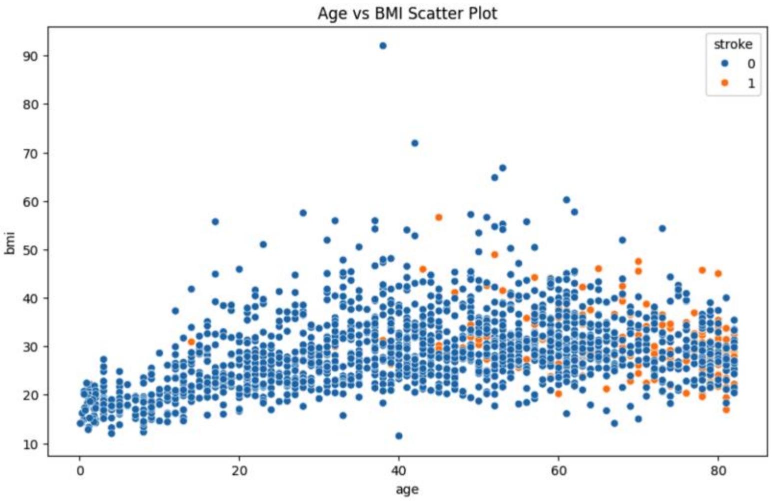
Feature correlation heatmap:



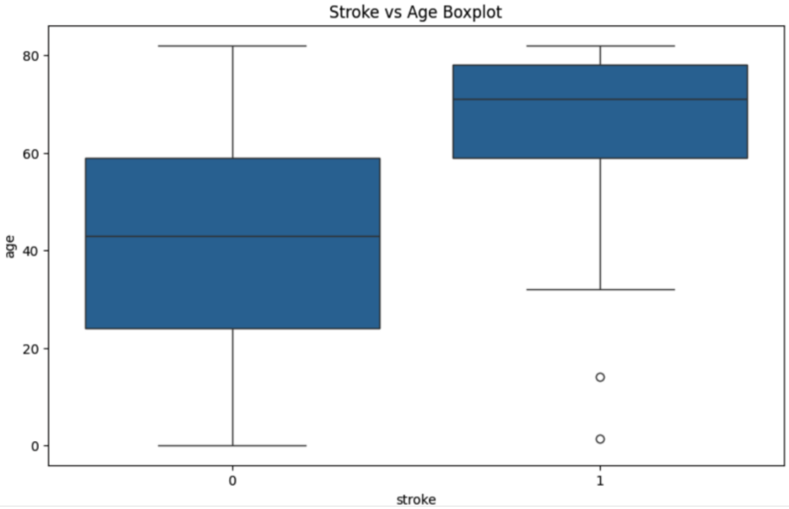
```
Text(0.5, 1.0, 'Age vs BMI Scatter Plot')
```

	age	hypertension	heart_disease	avg_glucose_level	bmi	stroke	gender_Male	ever_married_Yes	work_type_Never_worked	work_type_Private	work_type_Self-employed	work_type_children	Residence_type_Urban	oking_status_formerly smoked	smoking_status_never smoked	smoking_status_smokes	bmi_classification_Obese	bmi_classification_Over Weight	mi_classification_Under Weight
age	1	0.27	0.25	0.25	0.32	0.4	-0.028	0.65	-0.089	0.097	0.32	-0.62	0.026	0.25	0.099	0.064	0.22	0.21	-0.36
hypertension	0.27	1	0.15	0.18	0.17	0.21	0.04	0.14	-0.024	-0.0053	0.12	-0.13	0.028	0.072	0.051	0.046	0.12	0.03	-0.083
heart_disease	0.25	0.15	1	0.17	0.052	0.23	0.1	0.1	-0.017	0.016	0.049	-0.09	0.041	0.08	-0.061	0.098	0.069	0.046	-0.064
avg_glucose_level	0.25	0.18	0.17	1	0.22	0.2	0.059	0.16	-0.028	0.032	0.076	-0.11	0.024	0.081	0.023	0.019	0.2	-0.015	-0.069
bmi	0.32	0.17	0.052	0.22	1	0.074	-0.018	0.33	0.004	0.19	0.07	-0.43	0.0099	0.11	0.1	0.066	0.76	-0.12	-0.42
stroke	0.4	0.21	0.23	0.2	0.074	1	0.014	0.18	-0.028	0.014	0.095	-0.14	0.042	0.095	-0.0038	0.016	0.054	0.085	-0.099
gender_Male	-0.028	0.04	0.1	0.059	-0.018	0.014	1	-0.032	0.046	-0.027	-0.04	0.11	0.0092	0.04	-0.11	0.03	0.042	-0.0017	0.032
ever_married_Yes	0.65	0.14	0.1	0.16	0.33	0.18	-0.032	1	-0.099	0.15	0.19	-0.53	0.011	0.17	0.085	0.086	0.23	0.15	-0.31
work_type_Never_worked	-0.089	-0.024	-0.017	-0.028	0.004	-0.028	0.046	-0.099	1	-0.08	-0.031	-0.025	0.035	-0.033	0.019	-0.029	-0.0019	-0.0076	0.017
work_type_Private	0.097	-0.0053	0.016	0.032	0.19	0.014	-0.027	0.15	-0.08	1	-0.54	-0.43	-0.035	0.029	0.095	0.08	0.11	0.051	-0.22
work_type_Self-employed	0.32	0.12	0.049	0.076	0.07	0.095	-0.04	0.19	-0.031	-0.54	1	-0.17	0.017	0.095	0.042	0.0016	0.045	0.073	-0.094
work_type_children	-0.62	-0.13	-0.09	-0.11	-0.43	-0.14	0.11	-0.53	-0.025	-0.43	-0.17	1	0.0011	-0.17	-0.22	-0.15	-0.27	-0.18	0.53
Residence_type_Urban	0.026	0.028	0.041	0.024	0.0099	0.042	0.0092	0.011	0.035	-0.035	0.017	0.0011	1	-0.012	0.0016	0.042	6.2e-05	0.0046	0.016
smoking_status_formerly smoked	0.25	0.072	0.08	0.081	0.11	0.095	0.04	0.17	-0.033	0.029	0.095	-0.17	-0.012	1	-0.37	-0.21	0.12	-0.0065	-0.084
smoking_status_never smoked	0.099	0.051	-0.061	0.023	0.1	-0.0038	-0.11	0.085	0.019	0.095	0.042	-0.22	0.0016	-0.37	1	-0.32	0.035	0.039	-0.11
smoking_status_smokes	0.064	0.046	0.098	0.019	0.066	0.016	0.03	0.086	-0.029	0.08	0.0016	-0.15	0.042	-0.21	-0.32	1	0.046	0.03	-0.098
bmi_classification_Obese	0.22	0.12	0.069	0.2	0.76	0.054	0.042	0.23	-0.0019	0.11	0.045	-0.27	6.2e-05	0.12	0.035	0.046	1	-0.52	-0.21
bmi_classification_Over Weight	0.21	0.03	0.046	-0.015	-0.12	0.085	-0.0017	0.15	-0.0076	0.051	0.073	-0.18	-0.0046	-0.0065	0.039	0.03	-0.52	1	-0.17
bmi_classification_Under Weight	-0.36	-0.083	-0.064	-0.069	-0.42	-0.099	0.032	-0.31	0.017	-0.22	-0.094	0.53	0.016	-0.084	-0.11	-0.098	-0.21	-0.17	1

Age vs BMI Scatterplot:



Stroke Vs Age Boxplot:



Conclusion

The analysis of the stroke detection dataset demonstrated that addressing class imbalance was crucial for developing an effective stroke prediction model. Inital implementations involved high accuracy but performed poorly at actually detecting stroke cases which was the primary goal of the model. After examining the results of the standard implementations of Random Forest and Decision Tree, a more effective approach needed to be taken to address the class imbalance and remove bias towards the majority class. The models that incorporated class weighting and stratified k-fold cross-validation showed substantially higher recall values, succeeding in detecting stroke cases due to the higher importance placed on the stroke class. The Random Forest model with stratified k-fold cross-validation was the most effective approach as it maintained consistent class distribution across training and testing tests and sustained high recall scores across all five folds. The results highlight the importance of choosing appropriate splitting techniques to accommodate an unbalanced dataset. In medical prediction tasks, it is especially important to address class imbalance as identifying positive diagnosis, in this case a stroke, is more crucial than overall accuracy scores.

References

Link to dataset: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset?resource=download

Alanazi R. Identification and Prediction of Chronic Diseases Using Machine Learning Approach. J Healthc Eng. 2022 Feb 25;2022:2826127. doi: 10.1155/2022/2826127. PMID: 35251563; PMCID: PMC8896926.

Dhillon, A. (2020). Decision trees and random forests. Towards Data Science. https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991

GeeksforGeeks. (2023). How does the class_weight parameter in scikit-learn work? GeeksforGeeks. https://www.geeksforgeeks.org/how-does-the-classweight-parameter-in-scikit-learn-work/

GeeksforGeeks. (2023). ML | One hot encoding. Retrieved December 20, 2024, from https://www.geeksforgeeks.org/ml-one-hot-encoding/

Khalilia, M., Chakraborty, S., & Popescu, M. (2011). Predicting disease risks from highly imbalanced data using random forest. BMC Medical Informatics and Decision Making, 11(51). https://doi.org/10.1186/1472-6947-11-51

Olamendy, J. C. (2023). A comprehensive guide to stratified k-fold cross validation for unbalanced data. Medium. https://medium.com/@juanc.olamendy/a-comprehensive-guide-to-stratified-k-fold-cross-validation-for-unbalanced-data-014691060f17

Pace Hospital. (n.d.). BMI calculator ranges & importance. Retrieved December 20, 2024, from https://www.pacehospital.com/bmi-calculator-ranges-importance

pandas development team. (2024). pandas.DataFrame.sample. pandas: powerful Python data analysis toolkit. Retrieved December 20, 2024, from https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sample.html

scikit-learn developers. (2024). sklearn.model_selection.train_test_split. scikit-learn documentation. https://scikit-learn.org/1.5/modules/generated/sklearn.model_selection.train_test_split.html