"ALEXANDRU IOAN CUZA" UNIVERSITY

Project - FeedbackHHC

Student name: *Bodnar Alina-Florina; Galan Andrei-Iulian; Ignat Gabriel-Andrei; Sorodoc Tudor-Cosmin ; Ungurean Ana-Maria*

Course: *Artificial Intelligence*
Due date: *28.01.2024*

Table of Contents

1. Introduction

Healthcare quality prediction is a critical task in ensuring optimal patient outcomes. In this report, we present a meticulous exploration of our predictive modelling approach applied to home health agencies. The journey unfolds from **data preprocessing**, **feature selection**, to the deployment and evaluation of advanced **machine learning models**.

2. Data Preprocessing

In the realm of data preprocessing, meticulous attention to detail transforms raw datasets into a refined and structured form, setting the stage for effective machine learning model training. This crucial phase involves a series of operations aimed at refining and organizing the data, ensuring that it aligns with the requirements of machine learning algorithms.
Let's delve into some specific preprocessing steps applied to our dataset:

2.1. Geospatial Enhancement : Transforming Addresses into Coordinates.

To bring a geographical perspective into our dataset, we utilized the Google Maps API to convert textual addresses, such as `State`, `ZIP Code`, `City/Town`, and `Address`, into specific latitude and longitude coordinates. Giving each location a unique set of geographical coordinates, it will make it easier for our machine learning model to understand the spatial relationships between different places.
Example: AK ANCHORAGE 27001 4001 DALE STREET, SUITE 101 $\Rightarrow$ 61.1839967 -149.8175083

2.2. Transforming Dates into Numerical Values.

To facilitate the interpretation of dates, we transformed them into numerical values. The `Date Certified` column was transformed into the number of days since the agency was certified. This transformation provides a numerical measure that reflects how long ago the certification was acquired. The rationale behind this approach is to assign larger values to agencies that obtained their certification earlier, as it is assumed that agencies with longer certification histories may have accumulated more experience and expertise.

2.3. Transforming Text into Vectors.

In order to incorporate the textual information from the `Provider Name` column into our machine learning model, a technique called **Doc2Ve**c was applied (Doc2Vec is a neural network-based approach that learns the distributed representation of documents. It is an unsupervised learning technique that maps each document to a fixed-length vector in a high-dimensional space) This method converts text into numerical vectors, allowing the model to process and analyze textual data effectively. You can learn more about Doc2Vec here.
Procedure:

- Each entry in the Provider Name column was treated as a separate document.
- The trained model was used to infer vectors for each document in the "Provider Name" column.

- Result Integration:
  - The inferred vectors were incorporated into the dataset as new columns labeled as `Provider_Name_1`, `Provider_Name_2`, and so forth.
  - The original Provider Name column was dropped.

2.4. Label Encoding.
To optimize model performance and facilitate streamlined data interpretation, we strategically applied label encoding to specific columns associated with *performance categorization*, *offered services*, and *type of ownership*.

Performance Categorization: Achieve a numeric representation of performance categorizations for enhanced model understanding. *Columns Affected:* `DTC Performance Categorization`, `PPR Performance Categorization`, `PPH Performance Categorization`.
Mapping:

- "Worse Than National Rate" $\rightarrow$ 0
- "Same As National Rate" $\rightarrow$ 1
- "Better Than National Rate" $\rightarrow$ 2
- "Not Available" $\rightarrow$ 3
- "-" $\rightarrow$ 4.

Offered Services: Quantify the availability of various services in a numeric manner. *Columns Affected:* `Offers Nursing Care Services`, `Offers Physical Therapy Services`, `Offers Occupational Therapy Services`, `Offers Speech Pathology Services`, `Offers Medical Social Services`, `Offers Home Health Aide Services`.
Mapping:

- "Yes" $\rightarrow$ 0
- "No" $\rightarrow$ 1

Type of Ownership:
Mapping:

- "PROPRIETARY" $\rightarrow$ 0
- "VOLUNTARY NON PROFIT - RELIGIOUS AFFILIATION" $\rightarrow$ 1
- "GOVERNMENT - STATE/COUNTY" $\rightarrow$ 2
- .....

2.5. Removing Irrelevant Features.
Columns Removed:

1. Columns with Footnote in their name: These columns were excluded from the dataset because they consistently contained either **"-"** or the text **"The number of patients is too small to report"**. Since these footnotes provide little meaningful information, retaining them would only introduce noise into the model.
   Impact on Model: Introducing noise in the context of machine learning refers to including irrelevant or uninformative data in the dataset, which can negatively impact the model's ability to learn meaningful patterns. Noise can "distract" the model from recognizing important features, leading to overfitting or decreased predictive performance. In this case, as columns with footnotes contain repetitive and non-informative content, they were retained.

2. Telephone Number and CMS Certification Number : These columns were strategically eliminated as they contained unique identifiers for each record. Including them in the predictive modeling process could lead to overfitting, where the model memorizes the training data instead of learning general patterns.
   Impact on Model: Removing these columns ensures that the model focuses on relevant features, reducing the risk of overfitting and enhancing its ability to generalize to new data.

2.6. Removing Outliers.
Outliers, data points significantly deviating from the general trend, can impact the performance of machine learning models. To systematically detect and remove outliers, we employed the **Interquartile Range (IQR)** method.
Interquartile Range (IQR) method:

- Calculate the first quartile (Q1) and the third quartile (Q3) for the specific column.
- Determine the Interquartile Range (IQR) as the difference between Q3 and Q1.
- $LowerBound = Q1 - 1.5 \times IQR$
- $UpperBound = Q3 + 1.5 \times IQR$
  - The choice of 1.5 is a rule of thumb that has been widely adopted in statistics. It provides a balance between detecting potential outliers and avoiding the exclusion of too many data points that may still be valid.
- Remove all data points outside the lower and upper bounds.

Now, we will illustrate through **a graph** how the numbers of rows reduced**numbers of rows reduced** after **removing outliers** for each column.

1. How often the home health team began their patients' care in a timely manner
2. How often the home health team determined whether patients received a flu shot for the current flu season
3. How often patients got better at walking or moving around
4. How often patients got better at getting in and out of bed
5. How often patients got better at bathing

6. How often patients' breathing improved
7. How often patients got better at taking their drugs correctly by mouth
8. How often home health patients had to be admitted to the hospital
9. How often patients receiving home health care needed urgent, unplanned care in the ER without being admitted
10. Changes in skin integrity post-acute care: pressure ulcer/injury
11. How often physician-recommended actions to address medication issues were completely timely
12. Percent of Residents Experiencing One or More Falls with Major Injury
13. Application of Percent of Long Term Care Hospital Patients with an Admission and Discharge Functional Assessment and a Care Plan that Addresses Function
14. DTC Numerator
15. DTC Denominator
16. DTC Observed Rate
17. DTC Risk-Standardized Rate
18. DTC Risk-Standardized Rate (Lower Limit)
19. DTC Risk-Standardized Rate (Upper Limit)
20. PPR Numerator
21. PPR Denominator
22. PPR Observed Rate
23. PPR Risk-Standardized Rate
24. PPR Risk-Standardized Rate (Lower Limit)
25. PPR Risk-Standardized Rate (Upper Limit)
26. PPH Numerator
27. PPH Denominator
28. PPH Observed Rate
29. PPH Risk-Standardized Rate
30. PPH Risk-Standardized Rate (Lower Limit)
31. PPH Risk-Standardized Rate (Upper Limit)
32. How much Medicare spends on an episode of care at this agency, compared to Medicare spending across all agencies nationally
33. No. of episodes to calculate how much Medicare spends per episode of care at the agency, compared to spending at all agencies (national)

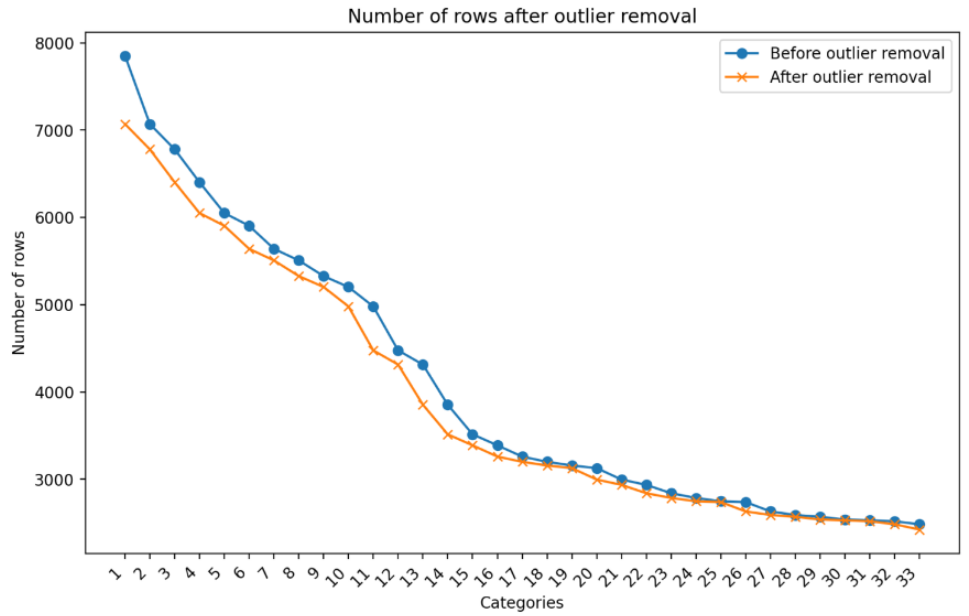

Figure 1: How the number of rows decreased when removing outliers

2.7. Handling Missing Values.
These missing entries can hinder the performance of our models, and various strategies exist for addressing them. One common approach is to replace missing values with the mean of their respective columns.

Why Impute with Column Mean? Imputing missing values with the mean of the column provides a way to maintain the overall statistical characteristics of the data, ensuring that the central tendency of the original distribution is conserved. This strategy is effective when the absence of values is unrelated to any specific pattern or factor, making it a suitable choice for scenarios where data is missing at random.
Now, we will illustrate through 2 graphs the **mean** and the **median** of each column. (The index for each column was above defined)

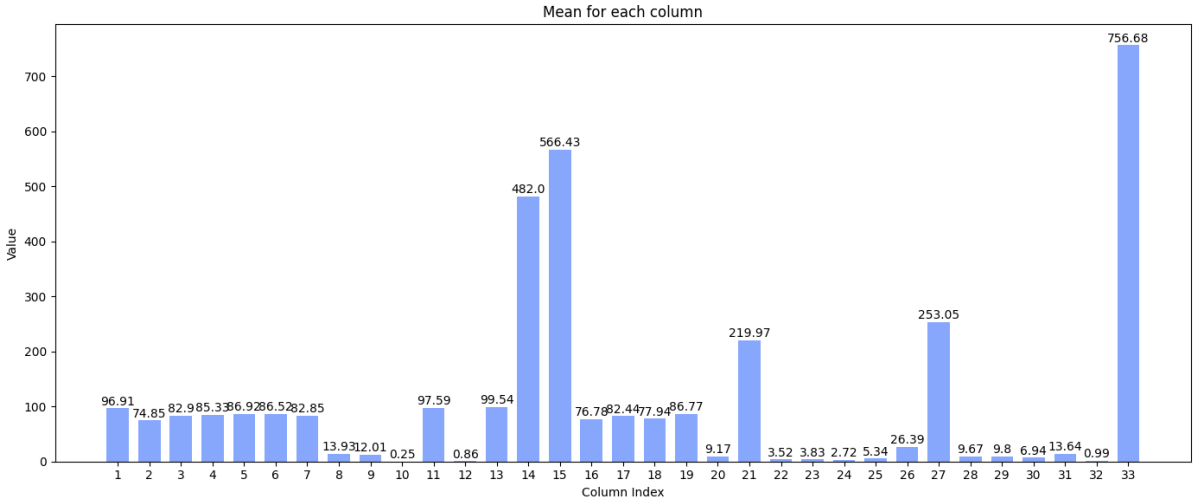

Figure 2: Mean for each column

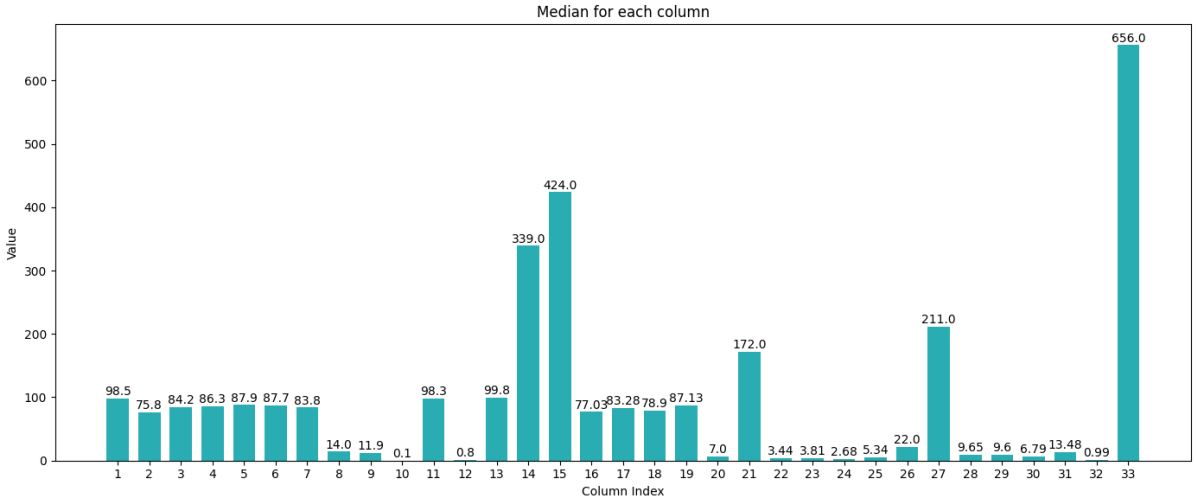

Figure 3: Median for each column

Please note that when using the application, you have the option to select a column, which will generate a graph displaying two bars: one representing the mean value and the other representing the median value.

3. Feature Selection

Feature selection is a crucial step in refining datasets for machine learning models. In this section, we delve into the methodology of feature selection based on correlation matrices, particularly using the **Pearson correlation coefficient**.
The Pearson correlation coefficient, often denoted as r, is a statistical measure that quantifies the linear relationship between two variables. Specifically, it assesses how well a change in one variable predicts a proportional change in another. The coefficient ranges from -1 to 1, signifying the strength and direction of the correlation:

- **Positive Correlation** $(\mathbf{r > 0})$**:** As one variable increases, the other tends to increase proportionally.
- **Negative Correlation** $(\mathbf{r < 0})$**:** As one variable increases, the other tends to decrease proportionally.
- **No Correlation** $(\mathbf{r = 0})$**:** There is no discernible linear relationship between the variables.

You can learn more about the Pearson correlation coefficient here.

3.1. Removing Features Weakly Correlated with the Output.

To ensure that our model focuses on the most relevant features, we removed columns with a weak correlation with the output variable. Specifically, we removed columns with a correlation coefficient less than **0.2**.(the absolute value of the correlation coefficient, of course) (This threshold was chosen based on the correlation matrix heatmap and also by analyzing the performance of the ML models with different thresholds)

3.2. Removing Features Highly Correlated between each other.

Now, we calculate the correlation matrix for the remaining features and remove features that are highly correlated with each other. (with a correlation coefficient greater than **0.8**). To keep things clear and avoid repetition, we decided to keep only one column from each group of similar ones.
Example: if *col*_1, *col*_2 and *col*_3 are highly correlated (in our case, have a coefficient greater than 0.8), we will keep only one of them.
Having unique and independent features makes our dataset cleaner and helps the model make better sense of the data.

Finally, after applying all the above steps **(Preprocessing and Feature Selection)** we started with .this dataset and ended up with this one, which is nice and ready to be used for training the ML models.

4. Machine Learning Models

In our quest to predict and understand patient care quality, we employed several machine learning algorithms including: Ada Boost , Random Forest , Neural Network , and Bayes Naive .

Additionally, for Ada Boost , Random Forest , and Neural Network , we conducted **hyperparameters tuning** to enhance their performance. Now, let's explore the concept of hyperparameter tuning.

4.1. A Dive into Hyperparameter Tuning.

Hyperparameter tuning is the process of selecting the optimal values for a machine learning model's hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network,etc. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

What are Hyperparameters? In the context of machine learning, hyperparameters are configuration variables that are set before the training process of a model begins. They control the learning process itself, rather than being learned from the data. Hyperparameters are often used to tune the performance of a model, and they can have a significant impact on the model's accuracy and generalization.

4.1.1. GridSearch and RandomSearch.

Two prevalent methods for hyperparameter tuning are Grid Search and Random Search. Both techniques systematically explore different combinations of hyperparameter values, allowing us to discover the configuration that yields the best performance on a given task.

1. **Grid Search:** This method entails defining a grid of hyperparameter values to be evaluated exhaustively. The model is trained and validated for each combination in the grid, helping identify the set of hyperparameters that optimize performance.

2. **Random Search:** In contrast, Random Search randomly samples hyperparameter combinations from predefined ranges. While not as exhaustive as Grid Search, Random Search often requires fewer evaluations and can be more efficient in finding good hyperparameter values.

4.2. Random Forest Classifier.

```
possible_parameters_rf = {
# Number of trees in the forest
"n_estimators": [10, 100, 200, 500, 750, 1_000],

# Function to measure the quality of a split
"criterion": ["gini", "entropy"],

# Maximum depth of the tree
"max_depth": [None, 5, 7, 8, 9, 10],

# Whether bootstrap samples are used when building trees
"bootstrap": [True, False]
}
```

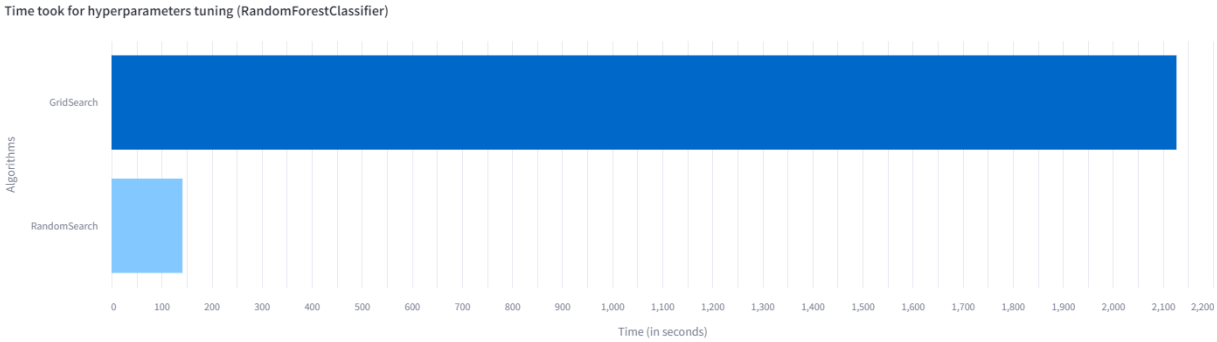

Figure 4: Time took for hyperparameters tuning

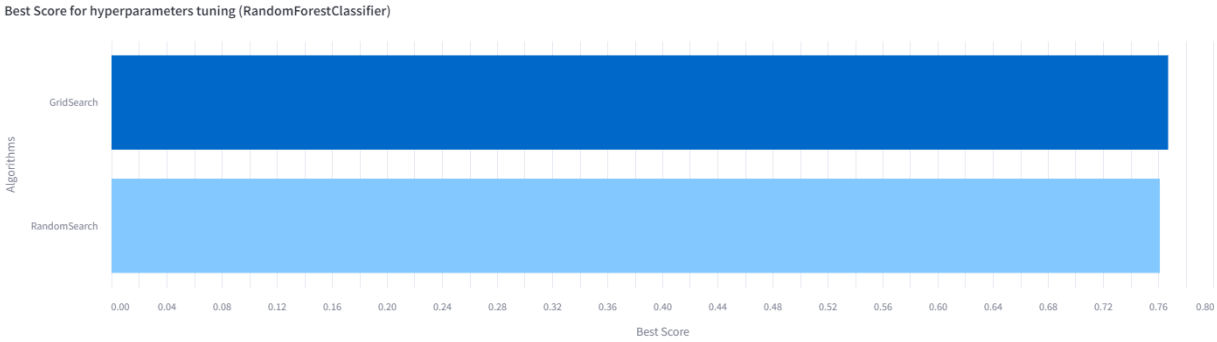

Figure 5: Score for hyperparameters tuning

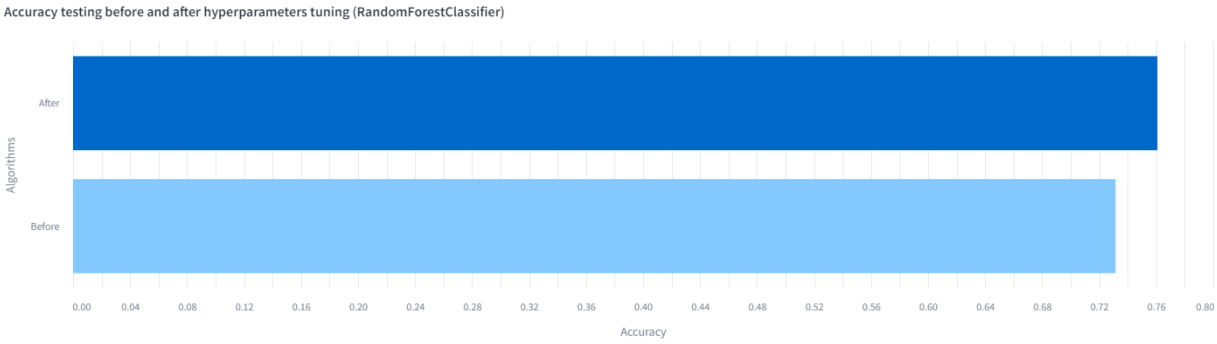

Figure 6: Accuracy testing beore/after hyperparameters tuning

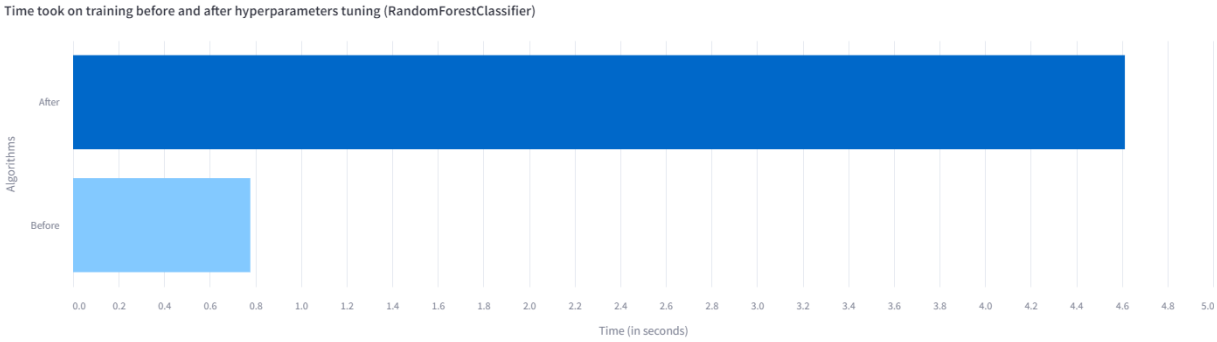

Figure 7: Time took on training before/after hyperparameters tuning

4.3. Ada Boost Classifier.

```
possible_parameters_ab = {
# Number of iterations
"n_estimators": [10, 25, 50, 100, 200, 250, 500],

# Learning rate for each iteration
# Multiplied with the weight of the distribution
"learning_rate": [0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0],

# Base estimator or weak learner used in AdaBoost
"estimator": [
DecisionTreeClassifier(max_depth=1),
DecisionTreeClassifier(max_depth=2, criterion="gini"),
DecisionTreeClassifier(max_depth=2, criterion="entropy")
]
}
```

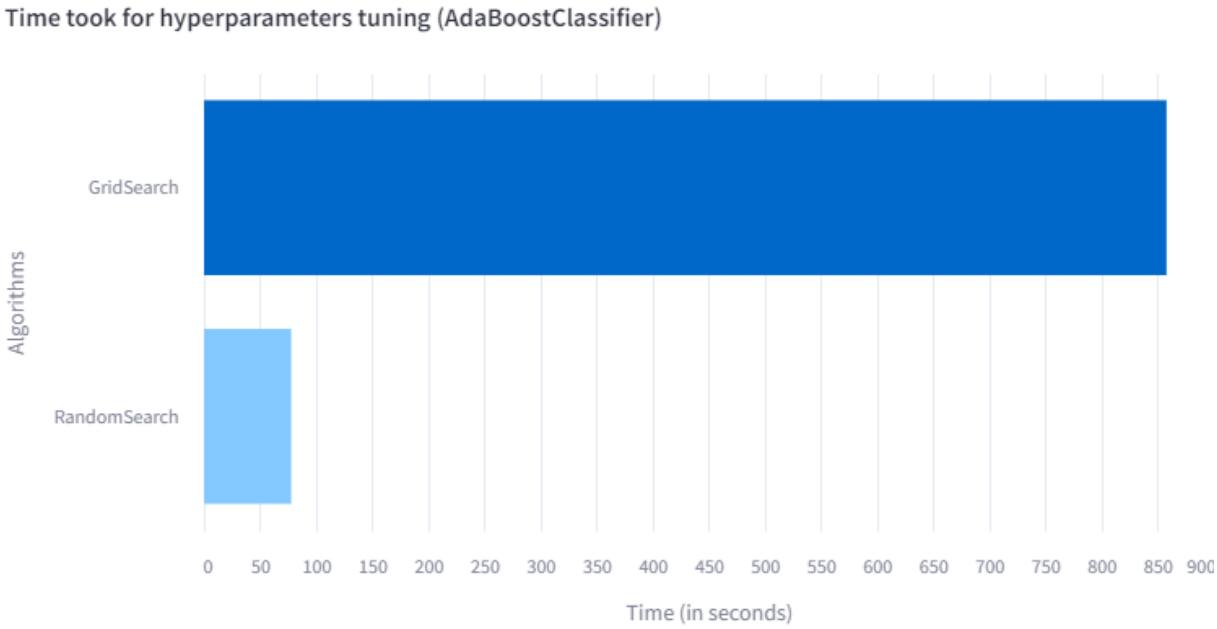

Figure 8: Time took for hyperparameters tuning

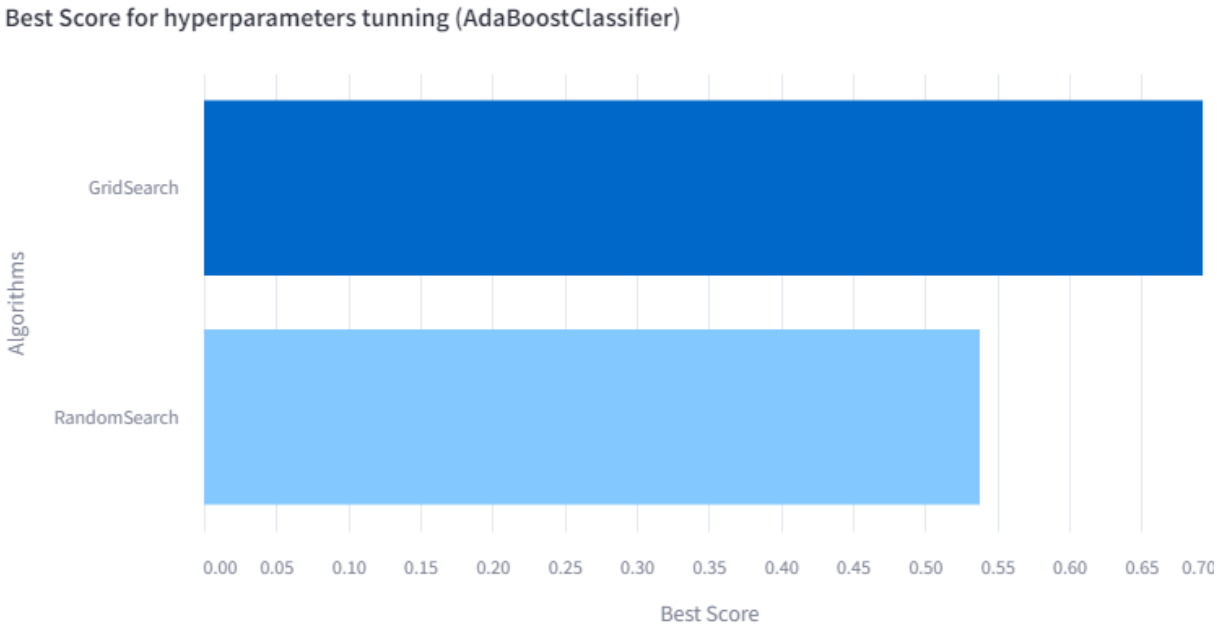

Figure 9: Score for hyperparameters tuning

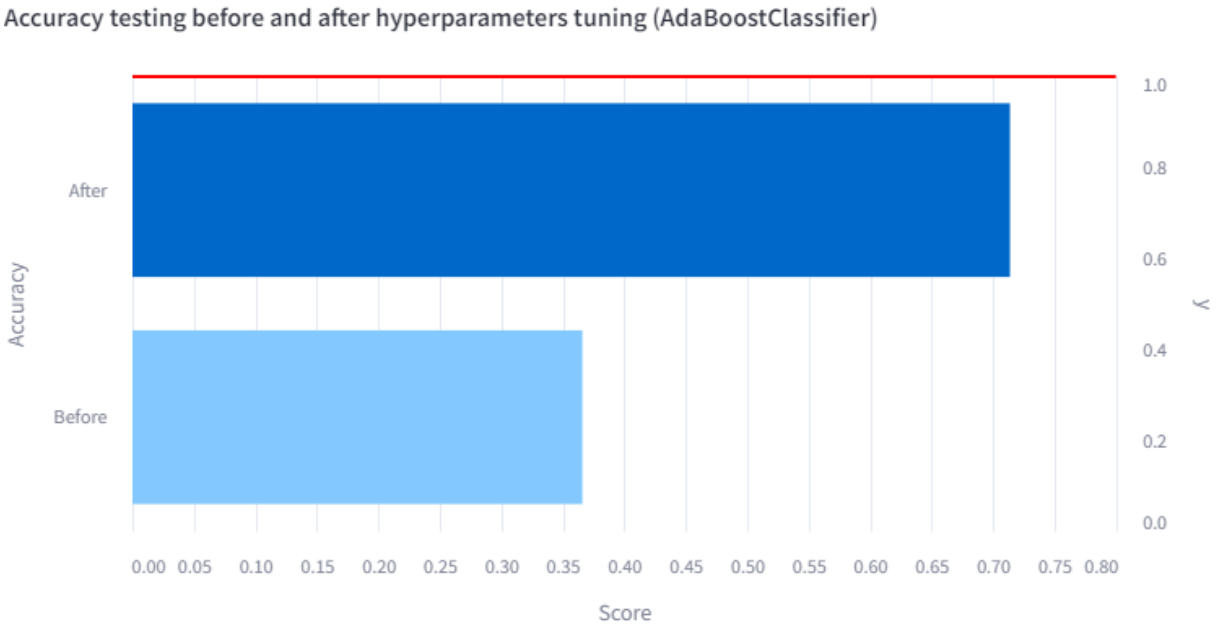

Figure 10: Accuracy testing beore/after hyperparameters tuning

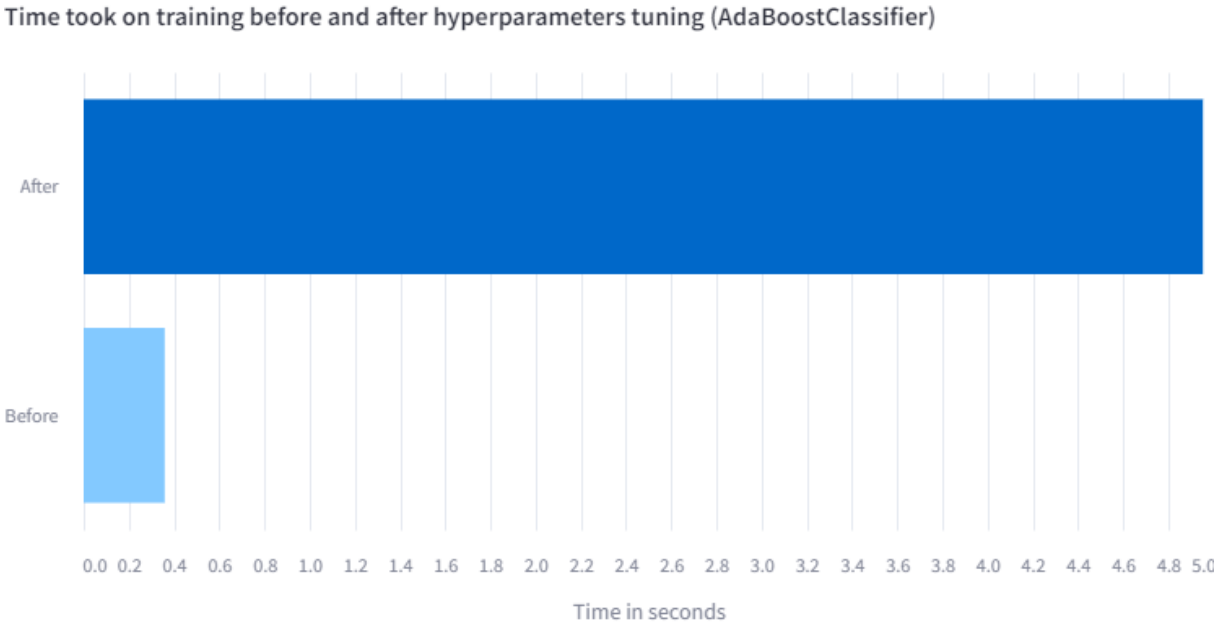

Figure 11: Time took on training before/after hyperparameters tuning

4.4. Neural Network.

```
possible_parameters_nn = {
# Maximum number of epochs
"max_iter" : [1_000, 1_500, 2_000],

# Activation function for the hidden layer
"activation" : ['relu', 'tanh', 'logistic'],
```

```
# Regularization parameter
"alpha" : [0.0001, 0.1, 0.05, 0.001],

# Size of the hidden layers
"hidden_layer_sizes" : [(100, 100),(50,50), (60,75)]
}
```

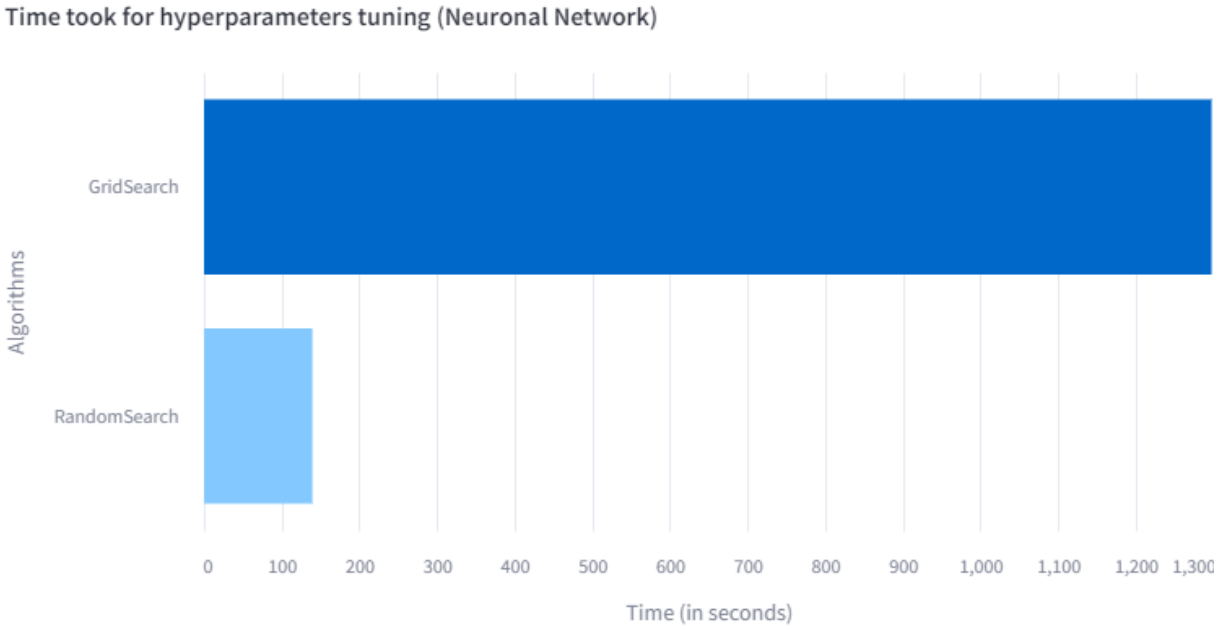

Figure 12: Time took for hyperparameters tuning

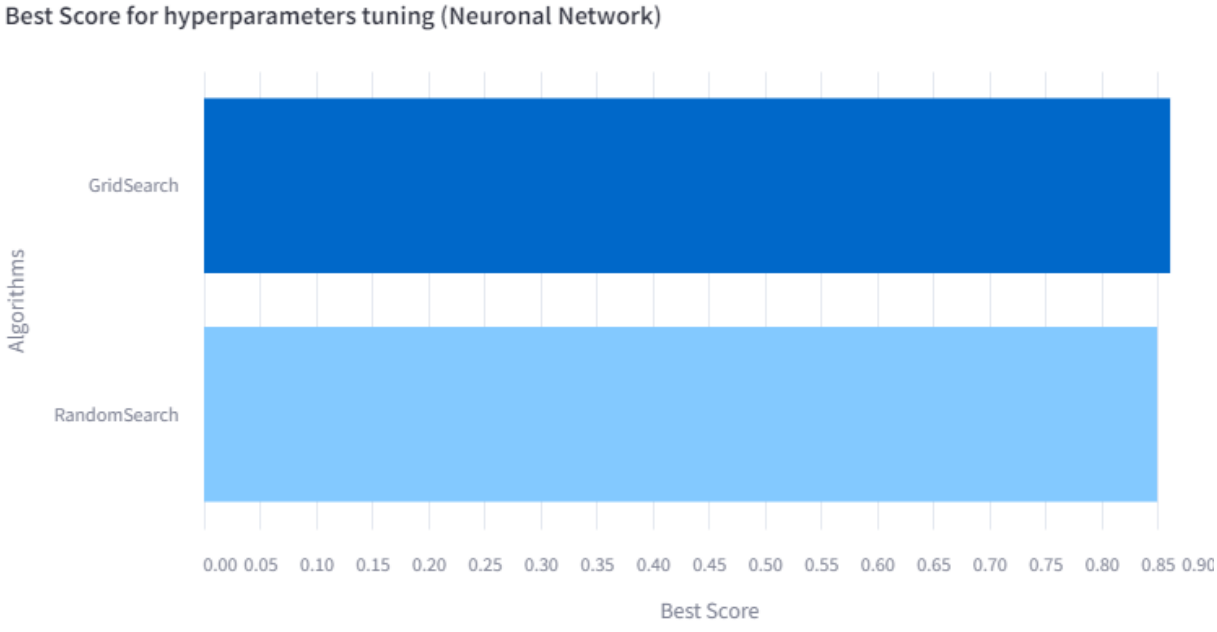

Figure 13: Score for hyperparameters tuning

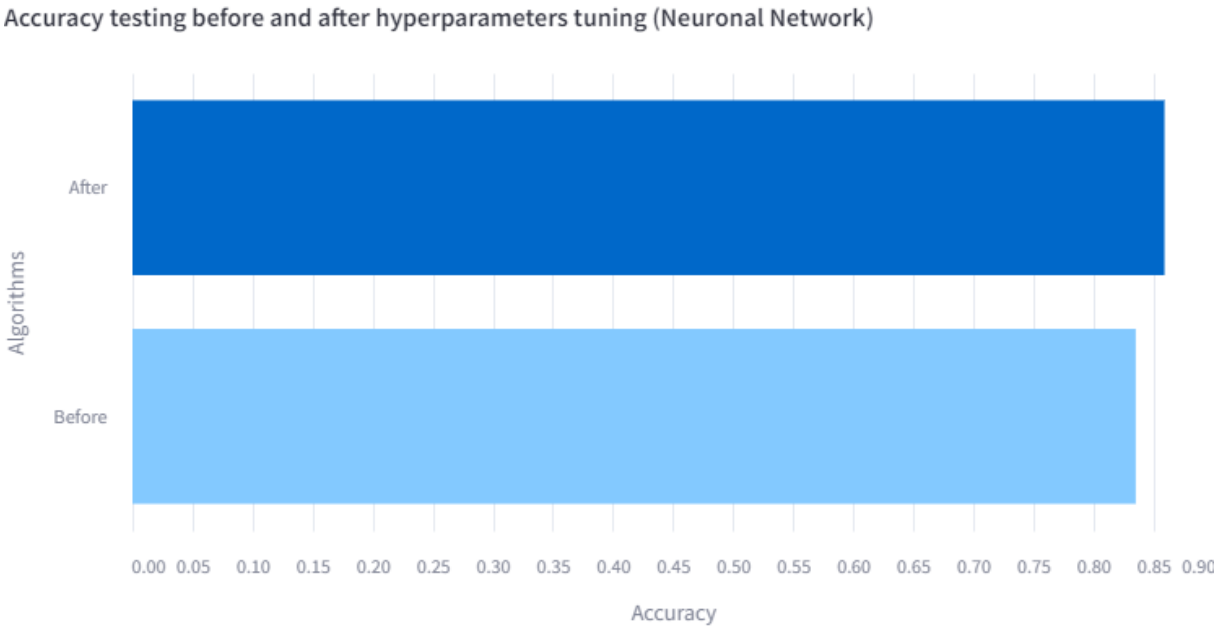

Figure 14: Accuracy testing beore/after hyperparameters tuning

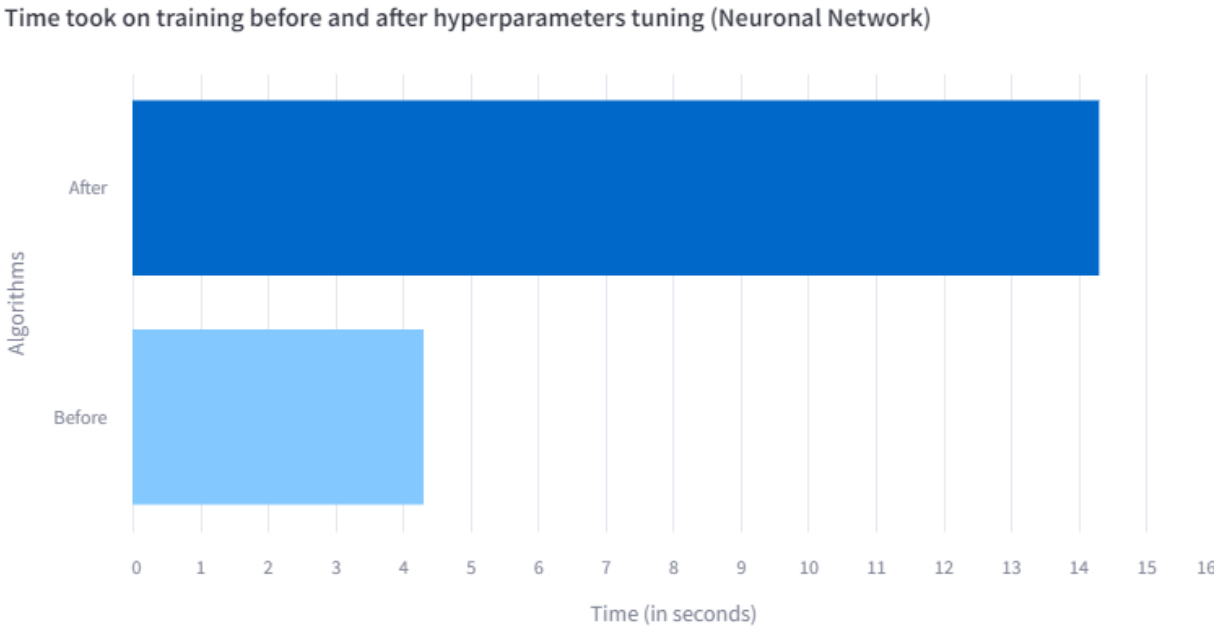

Figure 15: Time took on training before/after hyperparameters tuning

4.5. Comparations Between Models.

Next, we will illustrate the differences between these models using graphs.

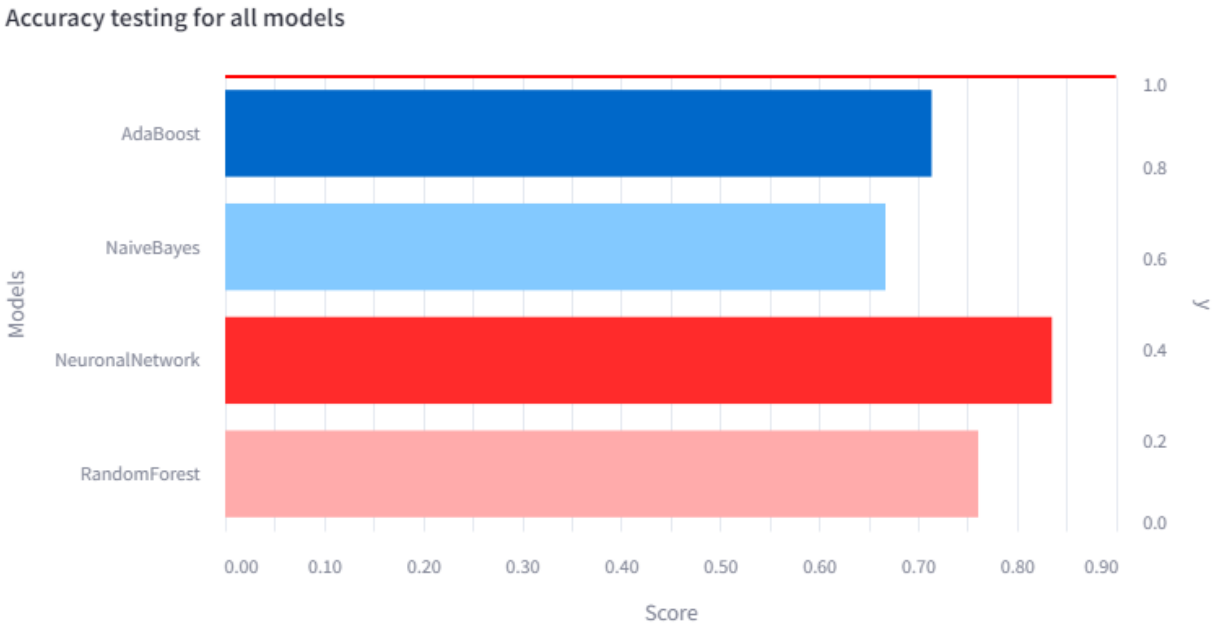

Figure 16: Accuracy on testing

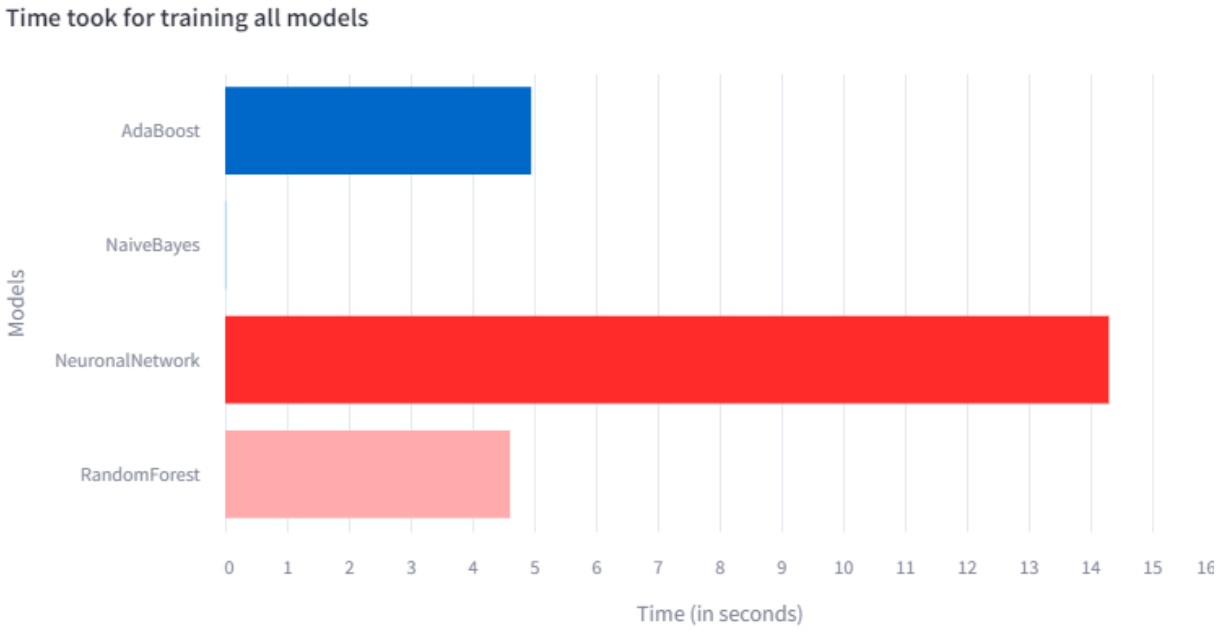

Figure 17: Time took for training

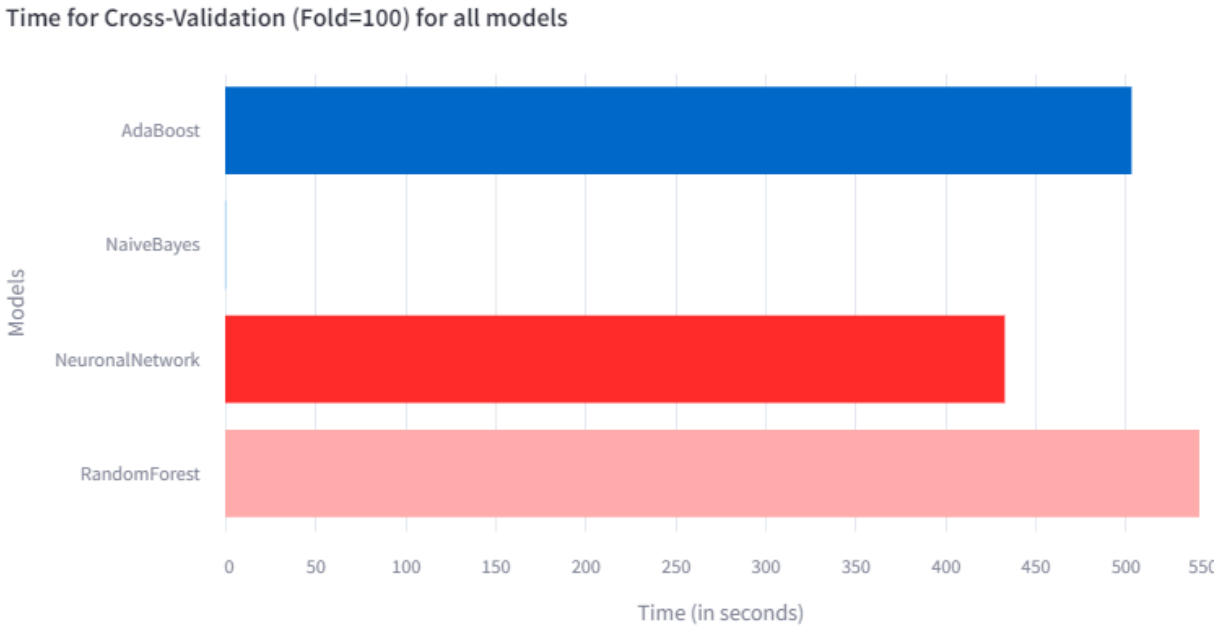

Figure 18: Time took for CV (with fold=100)

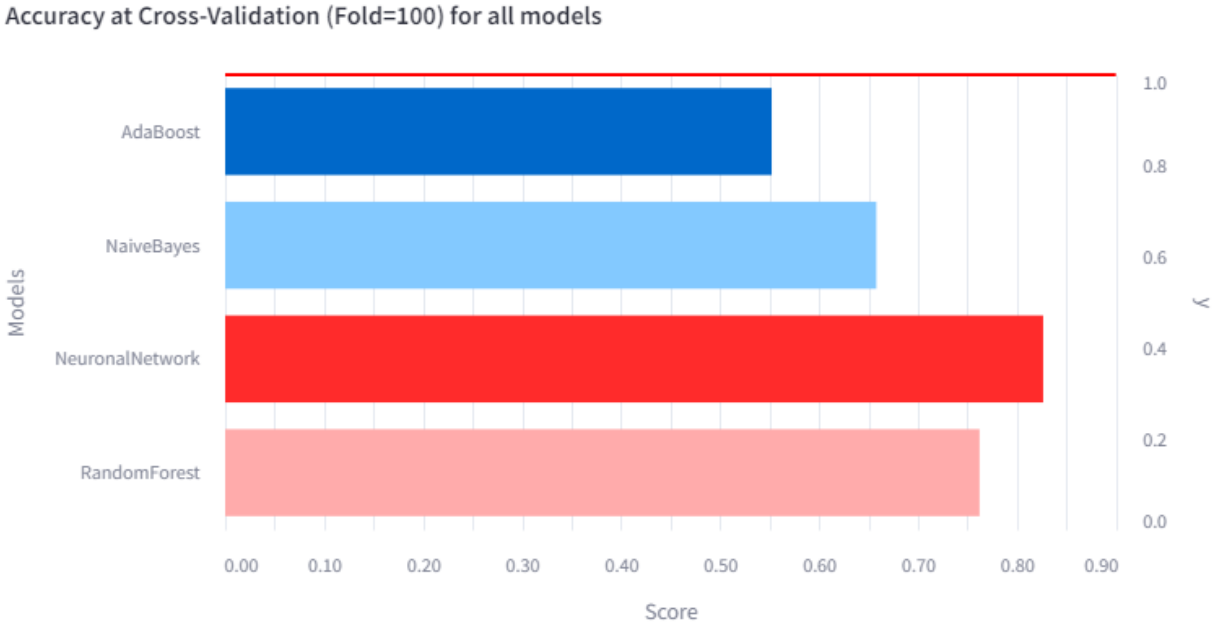

Figure 19: Accuracy at CV (with fold=100)

5. Conclusion

In conclusion, this research has included the critical steps of data preprocessing, feature selection, and machine learning model building, focusing on their significance in real-world problem-solving. We transformed addresses, dates, and text, employing label encoding and feature elimination to refine our dataset. This process paved the way for model training and evaluation.

Feature selection played a pivotal role in enhancing model performance, ensuring that our machine learning models could focus on the most pertinent aspects of the

data. The strategic removal of weakly correlated and highly correlated features not only streamlined the input variables but also contributed to the interpretability and computational efficiency of the models.

During model selection and optimization, we recognized the trade-offs between accuracy and training time. The Neural Network, exhibiting the highest accuracy, emerged as a powerful contender. However, this was coupled with a notable drawback - an extended training time. On the other hand, the Naive Bayes model boasted a swift training process but at the expense of lower accuracy.

The Random Forest Classifier, securing the second-best accuracy, struck a balance with a commendable training time. It showcased the versatility to handle diverse datasets and complex relationships within the data. Meanwhile, the Ada Boost Classifier, with its third-best accuracy, demonstrated a favorable compromise between model performance and training time.

Our comparative analysis highlighted the strengths and weaknesses of each model. The Neural Network excelled in accuracy but required patience during training. The Random Forest Classifier, with its commendable accuracy and reasonable training time, emerged as a well-rounded choice. The Ada Boost Classifier demonstrated competitive accuracy and efficient training times. In contrast, the Naive Bayes model proved to be a quick but less accurate option.

Lastly, we would like to highlight that our application, FeedbackHHC, serves as a valuable tool for finding home care services. Leveraging the insights gained from this research, FeedbackHHC not only aids in identifying suitable home care but also predicts the quality of services based on significant attributes, contributing to informed decision-making in the realm of home care selection.

5.1. Video - Demo. For a demonstration of our application: Click here!