Machine Learning Q1 Proposal:
Stroke Predictive Model

Team Members:
Chetan Maviti, Kanishk Sivanandam

10/23/2024

Table of Contents

Part 1 – Statement of Data Mining Goal

Strokes are medical emergencies that require rapid and accurate diagnosis to ensure effective treatment. The sooner a stroke is recognized and cared for, the better the outcome is likely to be. Currently, the process of diagnosis of a stroke in clinical settings can be enhanced by predictive modeling, which can help in identifying potential stroke incidentes before they become critical.

The goal of our project is to develop a machine learning model capable of detecting whether or not a patient is at risk of experiencing a stroke by analyzing a host of patient health and lifestyle features from a stroke dataset. This dataset includes detailed patient information such as patient age, gender, cholesterol levels, blood pressure, etc. Utilizing this data, after necessary preprocessing to deal with missing values, inconsistent data types, etc., a predictive model is created to best identify whether or not a given individual is at risk for stroke or not.

Part 2 – Description of Dataset

The stroke dataset consists of 12,760 instances, each characterized by 27 features that detail various health and lifestyle information of the patients. Data for each patient were collected on different days over a span from January 1, 2020, to December 31, 2024 (4 year span). The dataset contains a mix of binomial, qualitative, and quantitative attributes.

The attributes in the dataset are relatively uniformly distributed. The class variable, “diagnosis”, categorizes patients into “Stroke” and “No Stroke” which indicates whether or not they are at risk of stroke. This class variable is almost perfectly balanced/uniformly distributed with 50.27% labeled “Stroke” and 49.73% labeled “No Stroke”. It is also important to note that the dataset does in fact have a significant number of missing values, totaling 6,663. Attributes such as patient age, BMI index, and symptoms have at least 10% of their data missing. Still, there are no attributes with more than 17% missing values.

Here are our dataset’s attributes laid out, before any preprocessing:

- Patient_Name: Name of the patient
- Patient_ID: Unique ID of the patient
- Patient_Age: Age of the patient
- Patient_Gender: Gender of the patient
- Record_Date: Date of recording of attributes
- Dietary_Habits: Vegetarian, Vegan, etc.
- LDL_Cholesterol: LDL cholesterol level (bad cholesterol)
- Work_Type_of_patient: Profession-type of patient

- Metabolic_Equivalent_of_Task_Score: Measurement of physical activity
- Marital_Status: married, single, etc.
- Physical_Activity: Self-reported physical activity level
- Cholesterol_Levels: Categorized activity level
- Stress_Levels: Self-reported stress levels
- Average_Glucose_Level: Average glucose levels
- Heart_Disease: Indicator for hypertension (1 = Yes, 0 = No)
- Body_Mass_Index: BMI of patient
- Alcohol_Intake: Frequency of alcohol consumption
- HDL_Cholesterol: HDL cholesterol level (good cholesterol)
- Hypertension: Indicator for hypertension (1 = Yes, 0 = No)
- Family_History_of_Stroke: Family history of stroke (Yes/No)
- Diagnosis: Whether the patient is at risk for stroke or not
- Residence_Type: Urban or rural residence
- Systolic_BP: Systolic blood pressure
- Smoking_Status: Current smoking status
- Diastolic_BP: Diastolic blood pressure
- Stroke_History: Previous stroke incidents
- Symptoms: Reported symptoms (confusion, seizures, etc.)
- Blood_Pressure_Levels: Combined BP levels (Systolic / Diastolic)

Note that “Diagnosis” will be used as the class variable.

Part 3 – Data Preprocessing Procedure

3.1 Symptoms Attribute Clean Up

Before we could perform necessary preprocessing steps such as missing value replacement, we first had to deal with the unique “Symptoms” attribute. Unlike the other attributes, “Symptoms” contained lists of values rather than a single value for each instance. This list of values contained any combination of 10 symptoms listed for the patient: “Blurred Vision,” “Seizures,” “Difficulty Speaking,” “Weakness,” “Confusion,” “Headache,” “Dizziness,” “Severe Fatigue,” “Loss of Balance,” and “Numbness.” Due to this unique feature of various nominal values, we decided to split the “Symptoms” column into 10 different binary columns—one column for each symptom. This way each symptom could easily be taken into account by our classifier models without confusion. While at this step, we also removed all instances with missing values for the “Symptom” attribute, as filling in symptoms could lead to misdiagnosis. This step was accomplished on google colab using the pandas library.

```python
import pandas as pd
from google.colab import files
uploaded = files.upload()

df = pd.read_csv('stroke_dataset.csv')

df = df[df['Symptoms'].notna()]

symptom_set = set()
df['symptoms_list'] = df['Symptoms'].apply(lambda x: [s.strip() for s in x.split(',')])
for symptoms in df['symptoms_list']:
    symptom_set.update(symptoms)

for symptom in symptom_set:
    df[symptom] = df['symptoms_list'].apply(lambda symptoms: 1 if symptom in symptoms else 0)

df = df.drop(columns=['Symptoms', 'symptoms_list'])

print(df.head())

df.to_csv('symptoms_fixed.csv', index=False)
files.download('symptoms_fixed.csv')
```

3.2 Clean Up Unnecessary Columns

Attributes such as “Patient_name”, “Patient_id”, and “Record_date” only exist to help identify patients; they are irrelevant when trying to predict future strokes and were consequently removed. Additionally, features like “cholesterol_levels” and “blood_pressure_levels” were removed as they were simply compositions of two other attributes already present within the dataset (cholesterol_levels is the same as the ldl_cholesterol and hdl_cholesterol attributes combined).

3.3 Clean Up Missing Values

Missing values are detrimental when building a model. To handle missing values, we first determined that any instance with an empty blood pressure (systolic and diastolic) attribute could be removed, as blood pressure data is extremely significant in stroke detection, and filling in this data could bias results. For the remaining attributes with missing values, we used Weka’s ReplaceMissingValues filter to fill in missing data, which replaces data with mean for numeric data and mode for nominal data. This was possible as no attribute had any more than 17% missing values, and every feature had a uniform distribution of values. Finally, we used the floor function with the MathExpression Weka filter to round down all numeric values to integers, which only affected the newly filled in values.

3.4 Label Encoding

After cleaning up the dataset, the next step was to convert the nominal attributes with more than 2 possible values (not binomial) to numeric so that they could be analyzed.

This was achieved using the method of label encoding, where each possible value for an attribute is mapped to a numeric value. For example, the “Dietary_Habits” feature, which originally contained 7 possible string values such as “Pescatarian” was converted to a numeric attribute with the numbers 0 through 6, each mapped to one of the original strings. Label encoding was carried out using sklearn’s LabelEncoder method on Google Colab.

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from google.colab import files
uploaded = files.upload()

df = pd.read_csv('filled_values.csv')

nominal_columns = [
    'Dietary_Habits', 'Work_Type_of_patient',
    'Marital_Status', 'Physical_Activity', 'Alcohol_Intake',
    'Smoking_Status'
]

label_encoder = LabelEncoder()

for column in nominal_columns:
    df[column] = label_encoder.fit_transform(df[column])

print(df.head())

df.to_csv('label_encoded.csv', index=False)
files.download('label_encoded.csv')
```

3.5 Data Normalization

After the previous preprocessing steps, the different features had many different ranges of values. Some were binary, while others like the blood pressure attributes were in the hundreds. This discrepancy could overstate the importance of certain attributes when compared to others, causing issues for our classification model. Due to this, we normalized our data on the 0-1 scale using the Normalize filter in Weka. The normalization resulted in float values with a large number of decimal places, which we trimmed to just 3 places using a script in google colab.

```
import pandas as pd
from google.colab import files
uploaded = files.upload()

df = pd.read_csv("normalized.csv")

df = df.round(3)

print(df.head())

df.to_csv('final.csv', index=False)
files.download('final.csv')
```

Part 4 – Attribute Selection and Model Classifiers Used

Our final dataset contained a total of 32 attributes. Having a large number of features when performing classification can lead to overfit models, therefore attribute selection must be carried out to reduce dimensionality. We used 4 attribute selectors from Weka, along with our personal choice for a fifth attribute selector.

4.1 Attribute Selection

OneR:

The OneR attribute selection algorithm creates one ‘set of rules’ to determine what attributes are worth keeping. Attributes are ranked based on their ability to classify instances using a simple set of rules derived from the dataset. The OneR algorithm is detailed below in pseudocode (Sayad):

For each feature,
    For each value of that feature, make a rule as follows;
        Count how often each value of target (class) appears
        Find the most frequent class
        Make the rule assign that class to this value of the feature
    Calculate the total error of the rules of each feature
Choose the feature with the smallest total error.

After conducting OneR in Weka, the results are shown to the right:

We chose a cut-off score of **50.55** with the minimum bucket size being 6, so the only selected attributes were 25, 22, 21, 29, 1, 16, and 4.

```
Ranked attributes:
51.10639    4 LDL_Cholesterol
50.89381   16 Family_History_of_Stroke
50.84549    1 Patient_Age
50.71988   29 Severe Fatigue
50.61359   21 Stroke_History
50.5846    22 Blurred Vision
50.55561   25 Weakness
50.54595   17 Residence_Type
50.51696   15 Hypertension
50.401     27 Headache
50.39134   23 Seizures
50.35269   12 Body_Mass_Index
50.33337    7 Marital_Status
50.27539   30 Loss of Balance
50.27539    2 Patient_Gender
50.18842   13 Alcohol_Intake
50.10146    5 Work_Type_of_patient
50.00483   19 Smoking_Status
49.76326    6 Metabolic_Equivalent_of_Task_Score
49.61832    3 Dietary_Habits
49.54102    9 Stress_Levels
49.50237    8 Physical_Activity
49.46372   20 Diastolic_BP
49.40574   11 Heart_Disease
49.31877   14 HDL_Cholesterol
49.22215   26 Confusion
49.09653   18 Systolic_BP
48.9999    10 Average_Glucose_Level
48.91294   28 Dizziness
48.71002   31 Numbness
48.61339   24 Difficulty Speaking
```

CorrelationAttributeEval:

The CorrelationAttributeEval function evaluates attributes by measuring the Pearson's correlation between each attribute and the class. For nominal attributes, it treats each category as a separate indicator and calculates an overall correlation using a weighted average of these indicators. This correlation approach identifies attributes with strong correlation to the class. The results of the CorrelationAttributeEval for the stroke dataset are shown below:

The results of CorrelationAttributeEval are shown here:

```
Attribute Evaluator (supervised, Class (nominal): 32 Diagnosis):
        Correlation Ranking Filter
Ranked attributes:
 0.017893    16 Family_History_of_Stroke
 0.0161245   14 HDL_Cholesterol
 0.0161027   29 Severe Fatigue
 0.0136457   22 Blurred Vision
 0.012273    21 Stroke_History
 0.0122426   25 Weakness
 0.0117823    9 Stress_Levels
 0.0115932   15 Hypertension
 0.0111576   18 Systolic_BP
 0.010922    17 Residence_Type
 0.0104333   19 Smoking_Status
 0.010073    10 Average_Glucose_Level
 0.0099422    3 Dietary_Habits
 0.0098338   20 Diastolic_BP
 0.0097591    7 Marital_Status
 0.0094259   27 Headache
 0.0092256   23 Seizures
 0.0058731   30 Loss of Balance
 0.005496     2 Patient_Gender
 0.0053756    4 LDL_Cholesterol
 0.0047457    5 Work_Type_of_patient
 0.004386     6 Metabolic_Equivalent_of_Task_Score
 0.0041787   28 Dizziness
 0.004041    12 Body_Mass_Index
 0.0030377   26 Confusion
 0.0026639   31 Numbness
 0.001538     8 Physical_Activity
 0.0004985   13 Alcohol_Intake
 0.0004512    1 Patient_Age
 0.0000894   11 Heart_Disease
 0.0000324   24 Difficulty Speaking
```

By implementing a cut-off value of **0.01**, we are able to select only the attributes 10, 19, 17, 18, 15, 9, 25, 21, 22, 29, 14, and 16.

PrincipalComponents (PCA):

In a statistical sense, principal component analysis allows users to “summarize the information content in large data tables by means of a smaller set of ‘summary indices’ that can be more easily … analyzed” (Sartorius). In machine learning application, these “summary indices” are just “linear combinations or mixtures of the initial variables” (Jaadi) that will be used as attributes. These component vectors (linear combinations of initial variables) are ranked based on how much of the variance from the dataset they capture (Jaadi). The PCA of the stroke dataset is shown below:

```
Ranked attributes:
 0.9394    1 -0.705Metabolic_Equivalent_of_Task_Score+0.329Stress_Levels+0.299Systolic_BP+0.294Hypert
 0.9035    2 0.399Seizures-0.383Difficulty Speaking-0.365Family_History_of_Stroke=No-0.325Patient_Gen
 0.868     3 0.486Numbness-0.317Average_Glucose_Level+0.283Stroke_History-0.244Difficulty Speaking+0.
 0.833     4 0.461Weakness-0.362Work_Type_of_patient+0.341Physical_Activity-0.301Seizures-0.256Hypert
 0.7981    5 0.558Headache+0.324Systolic_BP-0.291Seizures-0.245Blurred Vision-0.2Body_Mass_Index...
 0.7635    6 0.517Severe Fatigue-0.444Loss of Balance-0.388Dizziness+0.279Body_Mass_Index+0.219Blurre
 0.7291    7 0.509Marital_Status-0.405Heart_Disease+0.342Diastolic_BP+0.327Loss of Balance-0.295Avera
 0.695     8 0.401Confusion-0.341Blurred Vision-0.304LDL_Cholesterol+0.259Residence_Type=Urban-0.247N
 0.6613    9 0.465Confusion+0.313Physical_Activity+0.286Dietary_Habits+0.266Stroke_History+0.232Diast
 0.6278   10 0.457Weakness+0.339Difficulty Speaking-0.328Dizziness-0.296Family_History_of_Stroke=No-0
 0.5945   11 0.393Residence_Type=Urban-0.317Blurred Vision-0.305Severe Fatigue-0.299Patient_Gender=Fe
 0.5614   12 -0.395Work_Type_of_patient-0.372Patient_Age-0.334Residence_Type=Urban-0.331HDL_Cholester
 0.5284   13 -0.537Alcohol_Intake+0.393Blurred Vision+0.313Systolic_BP+0.287Residence_Type=Urban+0.25
 0.4956   14 -0.412Smoking_Status-0.334Stroke_History-0.297Weakness+0.295Alcohol_Intake-0.289Average_
 0.4631   15 -0.407Dietary_Habits+0.353Diastolic_BP-0.314Body_Mass_Index-0.278Heart_Disease+0.275Dizz
 0.4307   16 0.384Confusion+0.346Physical_Activity+0.292Loss of Balance-0.256Dizziness+0.247Numbness.
 0.3986   17 -0.424Headache+0.386Numbness-0.378Family_History_of_Stroke=No-0.312Confusion-0.304Patien
 0.3666   18 -0.432Systolic_BP-0.346Family_History_of_Stroke=No-0.328Work_Type_of_patient+0.274Loss o
 0.3348   19 0.444LDL_Cholesterol+0.305Dizziness+0.301Stroke_History-0.291Weakness+0.267Body_Mass_Ind
 0.3032   20 -0.344Stroke_History-0.314Severe Fatigue+0.301Patient_Gender=Female+0.299Numbness-0.278S
 0.2717   21 -0.426Patient_Gender=Female+0.291Hypertension-0.289Marital_Status+0.274HDL_Cholesterol+0
 0.2404   22 -0.436Hypertension-0.379Average_Glucose_Level+0.279LDL_Cholesterol+0.275Severe Fatigue-0
 0.2092   23 0.382Body_Mass_Index+0.376Patient_Age-0.347Family_History_of_Stroke=No-0.296LDL_Choleste
 0.1783   24 -0.598Stress_Levels+0.325Diastolic_BP-0.256Marital_Status+0.246Smoking_Status+0.225Sever
 0.1477   25 0.465Dietary_Habits-0.337Physical_Activity-0.32Headache-0.274Body_Mass_Index-0.255Stroke
 0.1173   26 -0.45Physical_Activity+0.336Alcohol_Intake+0.326Residence_Type=Urban-0.317Average_Glucos
 0.0871   27 -0.545Heart_Disease-0.359Diastolic_BP+0.288Average_Glucose_Level-0.269Stress_Levels-0.25
 0.0572   28 0.383Systolic_BP-0.364Work_Type_of_patient+0.359HDL_Cholesterol+0.358Patient_Gender=Fema
 0.0275   29 0.395Smoking_Status-0.362Family_History_of_Stroke=No+0.324Alcohol_Intake-0.284Work_Type_
```

A cut-off value of **0.80** was chosen, meaning any component vectors that accounted for less than 80% of the variance in the dataset were not used as attributes. This meant only the top 4 component vectors were utilized as attributes with PCA.

ReliefF:

ReliefF is an attribute selection algorithm that evaluates the importance of features by analyzing how effectively each attribute differentiates between neighboring instances from different classes. The algorithm assigns scores to attributes based on their ability to distinguish between classes among otherwise similar instances. The results of reliefF on the stroke dataset are shown below:

The results of ReliefF are shown here:

```
Ranked attributes:
 0.00155019   10 Average_Glucose_Level
 0.00135279   30 Loss of Balance
 0.00134046    5 Work_Type_of_patient
 0.00130641   12 Body_Mass_Index
 0.00107257   17 Residence_Type
 0.00092763   21 Stroke_History
 0.000831     25 Weakness
 0.00078498    9 Stress_Levels
 0.0007292     1 Patient_Age
 0.00069572    2 Patient_Gender
 0.00066834   18 Systolic_BP
 0.00066154   13 Alcohol_Intake
 0.00052662    7 Marital_Status
 0.00040584   29 Severe Fatigue
 0.00035752   26 Confusion
 0.00029085   20 Diastolic_BP
 0.00018359   27 Headache
 0.00000271   14 HDL_Cholesterol
-0.00052179   24 Difficulty Speaking
-0.00054584    6 Metabolic_Equivalent_of_Task_Score
-0.00068161    4 LDL_Cholesterol
-0.00087931   19 Smoking_Status
-0.00092763   23 Seizures
-0.00101459   15 Hypertension
-0.00143975    8 Physical_Activity
-0.0014637     3 Dietary_Habits
-0.00155571   16 Family_History_of_Stroke
-0.00155571   22 Blurred Vision
-0.00231906   28 Dizziness
-0.00300512   31 Numbness
-0.00342062   11 Heart_Disease
```

After applying a cut-off value of **0.001**, the only selected attributes are 17, 12, 5, 30, and 10.

Personal Selection:

Attributes for Removal:

- **patient_gender**: We don't believe gender has a significant impact on the risk of stroke.
- **dietary_habits**: While important for overall health, dietary habits do not directly influence stroke risk.
- **work_type_of_patient**: A patient's job type is too indirectly related to stroke risk; there are better, more direct features in the dataset such as cholesterol and blood pressure.
- **metabolic_equivalent_of_task_score**: Indicates physical activity level via metabolic expenditure rate; again, too indirectly correlated with stroke risk.
- **marital_status**: Demographic factor, not indicative of stroke risk.
- **alcohol_intake**: We believe that only extremely excessive consumption of alcohol would have an effect on stroke risk ("excessive drinker" was not a possible value, only "frequent drinker")

- **residence_type**: Location impacts lifestyle/healthcare, but not directly correlated to stroke risk, again.
- **All Symptoms** (“Blurred Vision,” “Seizures,” “Difficulty Speaking,” “Weakness,” “Confusion,” “Headache,” “Dizziness,” “Severe Fatigue,” “Loss of Balance,” and “Numbness”): All symptoms are self-reported, but we wanted to maintain objectivity when predicting stroke risk.

Train/Test/Validation Split

After selecting 5 different pools of attributes, we created 5 copies of the final.csv post preprocessing, from which we selected the different attribute pools. After this, we performed a 70%/15%/15% train/validation/test split on each new dataset using google colab and scikit-learn. 70% of the data will be used to train the model, 15% will be used to validate the model and potentially readjusting the hyperparameters or dataset, and 15% will be used to test the model. This split resulted in 7244 patients for training, 1552 patients for validation, and 1553 patients for testing.

```
import sklearn
from google.colab import files
uploaded = files.upload()

import pandas as pd
from sklearn.model_selection import train_test_split

attribute = "ReliefF"

df = pd.read_csv(f'{attribute}.csv')

train_set, temp_set = train_test_split(df, test_size=0.3, random_state=42)
val_set, test_set = train_test_split(temp_set, test_size=0.5, random_state=42)

print("Training set size:", len(train_set))
print("Validation set size:", len(val_set))
print("Test set size:", len(test_set))

train_set.to_csv(f'{attribute}train.csv', index=False)
val_set.to_csv(f'{attribute}val.csv', index=False)
test_set.to_csv(f'{attribute}test.csv', index=False)

files.download(f'{attribute}train.csv')
files.download(f'{attribute}val.csv')
files.download(f'{attribute}test.csv')
```

4.2 Classifier Models

RandomForest:

The Random Forest classifier builds a collection of decision trees (hence, called a "forest"), where each tree independently predicts the class for a given input. The final classification is determined by a majority vote from all trees, meaning the most common prediction among all trees is chosen. This approach helps reduce the risk of overfitting by taking the average of multiple predictions.

J48:

J48 is a machine learning algorithm that creates decision trees to classify data. It builds the trees by analyzing training data to find the most informative features, using techniques to handle missing data and to prune the tree to prevent overfitting. This ensures that the model remains generalizable to new input data.

NaiveBayes:

Naive Bayes is a probabilistic classifier that applies Bayes' Theorem. It calculates the class probability using the formula:

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

In the context of the Naive Bayes classifier, X represents the set of features of the data ($X = x_1, x_2, \ldots x_n$), where each x_i is an individual feature. Y represents the class variable the model is trying to predict (stroke vs. no stroke in this case). The above equation can be rewritten as:

$$\prod_{\alpha=1}^{d} P(x_\alpha|y)P(y)$$

Naive Bayes uses these features X to estimate the probability of each possible class Y based on observed attributes.

Decision Table:

The DecisionTable classifier works by first creating a decision tree where each node tests an attribute and branches out based on the outcomes. It then translates these paths into a table of rules, each representing a potential decision path from the tree's root to its leaves. The final classification for each data point is determined by the most frequently occurring outcome among these paths. This simplifies the classification process into an easily interpretable tabular format, making it a more straightforward classification algorithm.

Part 5 – Results and Evaluation

5.1 Results

ReliefF with J48

```
=== Summary ===

Correctly Classified Instances         741               47.7141 %
Incorrectly Classified Instances       812               52.2859 %
Kappa statistic                         -0.0231
Mean absolute error                      0.501
Root mean squared error                  0.5023
Relative absolute error                100.1435 %
Root relative squared error            100.3975 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.751    0.775    0.471      0.751    0.579      -0.028   0.489     0.473     Stroke
                 0.225    0.249    0.496      0.225    0.310      -0.028   0.489     0.516     No Stroke
Weighted Avg.    0.477    0.501    0.484      0.477    0.439      -0.028   0.489     0.495

=== Confusion Matrix ===

   a   b   <-- classified as
 559 185 |   a = Stroke
 627 182 |   b = No Stroke
```

ReliefF with RandomForest:

```
=== Summary ===

Correctly Classified Instances         767               49.3883 %
Incorrectly Classified Instances       786               50.6117 %
Kappa statistic                         -0.0112
Mean absolute error                      0.5032
Root mean squared error                  0.5568
Relative absolute error                100.5852 %
Root relative squared error            111.2912 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.507    0.518    0.474      0.507    0.490      -0.011   0.493     0.470     Stroke
                 0.482    0.493    0.515      0.482    0.498      -0.011   0.493     0.520     No Stroke
Weighted Avg.    0.494    0.505    0.495      0.494    0.494      -0.011   0.493     0.496

=== Confusion Matrix ===

   a   b   <-- classified as
 377 367 |   a = Stroke
 419 390 |   b = No Stroke
```

ReliefF with NaiveBayes:

```
=== Summary ===

Correctly Classified Instances         749               48.2292 %
Incorrectly Classified Instances       804               51.7708 %
Kappa statistic                         -0.0153
Mean absolute error                      0.5005
Root mean squared error                  0.5007
Relative absolute error                100.0463 %
Root relative squared error            100.0793 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.728    0.744    0.474      0.728    0.574      -0.018   0.493     0.477     Stroke
                 0.256    0.272    0.506      0.256    0.340      -0.018   0.493     0.512     No Stroke
Weighted Avg.    0.482    0.498    0.491      0.482    0.452      -0.018   0.493     0.495

=== Confusion Matrix ===

   a   b   <-- classified as
 542 202 |   a = Stroke
 602 207 |   b = No Stroke
```

ReliefF with Decision Table:

```
=== Summary ===

Correctly Classified Instances         744               47.9073 %
Incorrectly Classified Instances       809               52.0927 %
Kappa statistic                          0
Mean absolute error                      0.5002
Root mean squared error                  0.5003
Relative absolute error                 99.9994 %
Root relative squared error             99.9993 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.479      1.000    0.648      ?        0.500     0.479     Stroke
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.521     No Stroke
Weighted Avg.    0.479    0.479    ?          0.479    ?          ?        0.500     0.501

=== Confusion Matrix ===

   a   b   <-- classified as
 744   0 |   a = Stroke
 809   0 |   b = No Stroke
```

OneR with J48:

```
=== Summary ===

Correctly Classified Instances         770               49.5815 %
Incorrectly Classified Instances       783               50.4185 %
Kappa statistic                          -0.0069
Mean absolute error                       0.4988
Root mean squared error                   0.5034
Relative absolute error                  99.7074 %
Root relative squared error             100.6315 %
Total Number of Instances               1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.513    0.520    0.476      0.513    0.494      -0.007   0.513     0.496     Stroke
                 0.480    0.487    0.517      0.480    0.498      -0.007   0.513     0.531     No Stroke
Weighted Avg.    0.496    0.503    0.497      0.496    0.496      -0.007   0.513     0.515

=== Confusion Matrix ===

   a   b   <-- classified as
 382 362 |   a = Stroke
 421 388 |   b = No Stroke
```

OneR with RandomForest:

```
=== Summary ===

Correctly Classified Instances         761               49.0019 %
Incorrectly Classified Instances       792               50.9981 %
Kappa statistic                          -0.0172
Mean absolute error                       0.507
Root mean squared error                   0.5503
Relative absolute error                 101.3572 %
Root relative squared error             110.0037 %
Total Number of Instances               1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.523    0.540    0.471      0.523    0.496      -0.017   0.484     0.474     Stroke
                 0.460    0.477    0.512      0.460    0.484      -0.017   0.484     0.502     No Stroke
Weighted Avg.    0.490    0.507    0.492      0.490    0.490      -0.017   0.484     0.489

=== Confusion Matrix ===

   a   b   <-- classified as
 389 355 |   a = Stroke
 437 372 |   b = No Stroke
```

OneR with NaiveBayes:

```
=== Summary ===

Correctly Classified Instances         766               49.3239 %
Incorrectly Classified Instances       787               50.6761 %
Kappa statistic                          0.0054
Mean absolute error                      0.5001
Root mean squared error                  0.5002
Relative absolute error                 99.9644 %
Root relative squared error             99.9929 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.730    0.724    0.481      0.730    0.580      0.006    0.511     0.494     Stroke
                 0.276    0.270    0.526      0.276    0.362      0.006    0.511     0.528     No Stroke
Weighted Avg.    0.493    0.488    0.504      0.493    0.466      0.006    0.511     0.512

=== Confusion Matrix ===

   a   b   <-- classified as
 543 201 |   a = Stroke
 586 223 |   b = No Stroke
```

OneR with Decision Table:

```
=== Summary ===

Correctly Classified Instances         782               50.3542 %
Incorrectly Classified Instances       771               49.6458 %
Kappa statistic                          0.0071
Mean absolute error                      0.5002
Root mean squared error                  0.5003
Relative absolute error                 99.9899 %
Root relative squared error            100.0033 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.504    0.497    0.483      0.504    0.493      0.007    0.504     0.481     Stroke
                 0.503    0.496    0.524      0.503    0.514      0.007    0.504     0.523     No Stroke
Weighted Avg.    0.504    0.496    0.504      0.504    0.504      0.007    0.504     0.503

=== Confusion Matrix ===

   a   b   <-- classified as
 375 369 |   a = Stroke
 402 407 |   b = No Stroke
```

CorrelationAttributeEval with J48:

```
=== Summary ===

Correctly Classified Instances         799               51.4488 %
Incorrectly Classified Instances       754               48.5512 %
Kappa statistic                          0.0341
Mean absolute error                      0.4957
Root mean squared error                  0.5599
Relative absolute error                 99.0916 %
Root relative squared error            111.919  %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.581    0.546    0.494      0.581    0.534      0.035    0.515     0.485     Stroke
                 0.454    0.419    0.541      0.454    0.493      0.035    0.515     0.536     No Stroke
Weighted Avg.    0.514    0.480    0.518      0.514    0.513      0.035    0.515     0.512

=== Confusion Matrix ===

   a   b   <-- classified as
 432 312 |   a = Stroke
 442 367 |   b = No Stroke
```

CorrelationAttributeEval with RandomForest:

```
=== Summary ===

Correctly Classified Instances         831               53.5093 %
Incorrectly Classified Instances       722               46.4907 %
Kappa statistic                          0.074
Mean absolute error                      0.4974
Root mean squared error                  0.5076
Relative absolute error                 99.4262 %
Root relative squared error            101.4609 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.587    0.513    0.513      0.587    0.548      0.075    0.522     0.489     Stroke
                 0.487    0.413    0.562      0.487    0.522      0.075    0.522     0.539     No Stroke
Weighted Avg.    0.535    0.461    0.539      0.535    0.534      0.075    0.522     0.515

=== Confusion Matrix ===

   a   b   <-- classified as
 437 307 |   a = Stroke
 415 394 |   b = No Stroke
```

CorrelationAttributeEval with NaiveBayes:

```
=== Summary ===

Correctly Classified Instances         796               51.2556 %
Incorrectly Classified Instances       757               48.7444 %
Kappa statistic                          0.0384
Mean absolute error                      0.4994
Root mean squared error                  0.4999
Relative absolute error                 99.8422 %
Root relative squared error             99.9201 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.685    0.646    0.494      0.685    0.574      0.041    0.530     0.500     Stroke
                 0.354    0.315    0.550      0.354    0.430      0.041    0.530     0.543     No Stroke
Weighted Avg.    0.513    0.474    0.523      0.513    0.499      0.041    0.530     0.522

=== Confusion Matrix ===

   a   b   <-- classified as
 510 234 |   a = Stroke
 523 286 |   b = No Stroke
```

CorrelationAttributeEval with Decision Table:

```
=== Summary ===

Correctly Classified Instances         780               50.2254 %
Incorrectly Classified Instances       773               49.7746 %
Kappa statistic                          0.0245
Mean absolute error                      0.5002
Root mean squared error                  0.5003
Relative absolute error                 99.9901 %
Root relative squared error            100.0058 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.758    0.733    0.487      0.758    0.593      0.029    0.506     0.480     Stroke
                 0.267    0.242    0.545      0.267    0.359      0.029    0.506     0.528     No Stroke
Weighted Avg.    0.502    0.477    0.518      0.502    0.471      0.029    0.506     0.505

=== Confusion Matrix ===

   a   b   <-- classified as
 564 180 |   a = Stroke
 593 216 |   b = No Stroke
```

PCA with J48:

```
=== Summary ===

Correctly Classified Instances         744               47.9073 %
Incorrectly Classified Instances       809               52.0927 %
Kappa statistic                          0
Mean absolute error                      0.5002
Root mean squared error                  0.5003
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.479      1.000    0.648      ?        0.500     0.479     Stroke
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.521     No Stroke
Weighted Avg.    0.479    0.479    ?          0.479    ?          ?        0.500     0.501

=== Confusion Matrix ===

   a   b   <-- classified as
 744   0 |   a = Stroke
 809   0 |   b = No Stroke
```

PCA with RandomForest:

```
=== Summary ===

Correctly Classified Instances         785               50.5473 %
Incorrectly Classified Instances       768               49.4527 %
Kappa statistic                          0.0117
Mean absolute error                      0.4991
Root mean squared error                  0.5153
Relative absolute error                 99.7776 %
Root relative squared error            103.0091 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.515    0.503    0.485      0.515    0.499      0.012    0.508     0.488     Stroke
                 0.497    0.485    0.527      0.497    0.511      0.012    0.508     0.519     No Stroke
Weighted Avg.    0.505    0.494    0.507      0.505    0.506      0.012    0.508     0.504

=== Confusion Matrix ===

   a   b   <-- classified as
 383 361 |   a = Stroke
 407 402 |   b = No Stroke
```

PCA with NaiveBayes:

```
=== Summary ===

Correctly Classified Instances         778               50.0966 %
Incorrectly Classified Instances       775               49.9034 %
Kappa statistic                          0.0295
Mean absolute error                      0.5001
Root mean squared error                  0.5002
Relative absolute error                 99.9748 %
Root relative squared error             99.985  %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.855    0.824    0.488      0.855    0.621      0.041    0.527     0.490     Stroke
                 0.176    0.145    0.568      0.176    0.268      0.041    0.527     0.541     No Stroke
Weighted Avg.    0.501    0.471    0.530      0.501    0.437      0.041    0.527     0.517

=== Confusion Matrix ===

   a   b   <-- classified as
 636 108 |   a = Stroke
 667 142 |   b = No Stroke
```

PCA with Decision Table:

```
=== Summary ===

Correctly Classified Instances         744               47.9073 %
Incorrectly Classified Instances       809               52.0927 %
Kappa statistic                          0
Mean absolute error                      0.5002
Root mean squared error                  0.5003
Relative absolute error                 99.9994 %
Root relative squared error             99.9993 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.479      1.000    0.648      ?        0.500     0.479     Stroke
                 0.000    0.000    ?          0.000    ?          ?        0.500     0.521     No Stroke
Weighted Avg.    0.479    0.479    ?          0.479    ?          ?        0.500     0.501

=== Confusion Matrix ===

   a   b   <-- classified as
 744   0 |   a = Stroke
 809   0 |   b = No Stroke
```

Personal Selection with J48:

```
=== Summary ===

Correctly Classified Instances         727               46.8126 %
Incorrectly Classified Instances       826               53.1874 %
Kappa statistic                         -0.0646
Mean absolute error                      0.5255
Root mean squared error                  0.6847
Relative absolute error                105.0537 %
Root relative squared error            136.8746 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.457    0.522    0.446      0.457    0.452      -0.065   0.473     0.463     Stroke
                 0.478    0.543    0.489      0.478    0.484      -0.065   0.473     0.503     No Stroke
Weighted Avg.    0.468    0.533    0.469      0.468    0.468      -0.065   0.473     0.484

=== Confusion Matrix ===

   a   b   <-- classified as
 340 404 |   a = Stroke
 422 387 |   b = No Stroke
```

Personal Selection with RandomForest:

```
=== Summary ===

Correctly Classified Instances         790               50.8693 %
Incorrectly Classified Instances       763               49.1307 %
Kappa statistic                          0.0208
Mean absolute error                      0.4996
Root mean squared error                  0.5037
Relative absolute error                 99.8767 %
Root relative squared error            100.6939 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.552    0.532    0.489      0.552    0.519      0.021    0.506     0.476     Stroke
                 0.468    0.448    0.532      0.468    0.498      0.021    0.506     0.533     No Stroke
Weighted Avg.    0.509    0.488    0.511      0.509    0.508      0.021    0.506     0.506

=== Confusion Matrix ===

   a   b   <-- classified as
 411 333 |   a = Stroke
 430 379 |   b = No Stroke
```

Personal Selection with NaiveBayes:

```
=== Summary ===

Correctly Classified Instances         793               51.0625 %
Incorrectly Classified Instances       760               48.9375 %
Kappa statistic                          0.032
Mean absolute error                      0.5
Root mean squared error                  0.5007
Relative absolute error                 99.954  %
Root relative squared error            100.0859 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.649    0.617    0.492      0.649    0.560      0.034    0.508     0.481     Stroke
                 0.383    0.351    0.543      0.383    0.449      0.034    0.508     0.530     No Stroke
Weighted Avg.    0.511    0.478    0.518      0.511    0.502      0.034    0.508     0.506

=== Confusion Matrix ===

   a   b   <-- classified as
 483 261 |   a = Stroke
 499 310 |   b = No Stroke
```

Personal Selection with Decision Table:

```
=== Summary ===

Correctly Classified Instances         782               50.3542 %
Incorrectly Classified Instances       771               49.6458 %
Kappa statistic                          0.0071
Mean absolute error                      0.5002
Root mean squared error                  0.5003
Relative absolute error                 99.9899 %
Root relative squared error            100.0033 %
Total Number of Instances             1553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.504    0.497    0.483      0.504    0.493      0.007    0.504     0.481     Stroke
                 0.503    0.496    0.524      0.503    0.514      0.007    0.504     0.523     No Stroke
Weighted Avg.    0.504    0.496    0.504      0.504    0.504      0.007    0.504     0.503

=== Confusion Matrix ===

   a   b   <-- classified as
 375 369 |   a = Stroke
 402 407 |   b = No Stroke
```

5.2 Evaluation

Through 4 different classifiers and 5 attribute selection methods, we created 20 different classification models to predict whether or not a patient is at risk of having a stroke based on a variety of factors. The 5 classifier models with the highest accuracies are listed below along with TP rate, FP Rate, and ROC Area, respectively:

1. CorrelationAttributeEval with RandomForest – 53.51%, 0.535, 0.461, 0.522
2. CorrelationAttributeEval with J48 – 51.45%, 0.514, 0.480, 0.515
3. CorrelationAttributeEval with NaiveBayes – 51.26%, 0.513, 0.474, 0.530
4. Personal Selection with NaiveBayes – 51.10%, 0.511, 0.478, 0.508
5. Personal Selection with RandomForest – 50.87%, 0.509, 0.488, 0.506

Since **CorrelationAttributeEval with RandomForest** scored the highest accuracy as well the highest TP rate and lowest FP rate of the top 5, we believe that this classifier model is the best model out of the 20 for predicting the possibility of a stroke for a patient. However, it is important that we acknowledge the somewhat poor results of our classifier models, with the highest accuracy only being slightly above 50%. We believe that these negative results were ultimately caused by the complexity of the problem at hand. First, the dataset contains many subjective attributes such as “Stress_Level” and “Physical_Activity.” As these are self-reported by the patients, they may contain inaccuracies, affecting the training process. Another potential reason is the nature of the many “Symptoms” attributes, as these symptoms may appear in various conditions, and these overlapping patterns may confuse the model. The final potential reason is the possibility of multicollinearity in our problem. Since strokes are a major health issue with various different causes, there is a high chance that many of the attributes are correlated amongst each other in large combinations. This complicates the problem, making it difficult to reduce attributes without losing valuable data, which in turn can lead to overfitting due to high dimensionality. We believe that our poor results were caused by a combination of these issues.

Part 6 – Conclusion and Reproducing Our Model

In summary, the combination of CorrelationAttributeEval with the RandomForest classifier emerged as the most accurate model for predicting stroke risk in our study. However, the challenges of subjective data via self-reporting and potential multicollinearity in stroke-related attributes indicates that there are certainly ways to better the model. To increase accuracy in the future, datasets void of largely subjective data can be prioritized. Additionally, attribute selection algorithms capable of handling complex attribute selection (like multicollinearity) must be explored further to best improve accuracy. By improving on these aspects, we strive to develop a more precise machine learning tool with higher accuracy.

Steps to Reproduce Our Model: CorrelationAttributeEval with RandomForest →
1. Open Weka and load the **final.csv** achieved after following the preprocessing steps
2. Navigate to the **Select attributes** tab. Under Attribute Evaluator click Choose > attributeSelection > **CorrelationAttributeEval**.
3. A popup will appear informing you that you must use the **Ranker** search method in order to perform CorrelationAttributeEval. Click yes. If the popup does not appear, then the correct search method is already selected.
4. Click on the dropdown menu labeled “No class,” and select “**(Nom) Diagnosis**”
5. Click start, then **take note** of the ranked attributes with a value **> 0.01**
6. Return to the Preprocess tab, and check the box to the left of all attributes that you did not take note of in **Step 6** (Excluding Diagnosis). Press the remove button.
7. Hit the **Save…** button, and export as **CorrelationAttributeEval.csv**.
8. Change the “attribute” variable to “CorrelationAttributeEval” in the Train/Test/Validation split code located in **Step 4.1** in Google Colab. Hit the run on the code block, then select the newly exported csv file from the file selector.
9. Return to Weka and load the **CorrelationAttributeEvaltrain.csv** that downloaded to your computer from the Google Colab code.
10. Navigate to the **Classify.** Under Classifier click Choose > classifiers > trees > RandomForest.
11. Click **Supplied test set** under Test options and select **CorrelationAttributeEvaltest.csv**, downloaded from the python code.
12. Hit **Start**. If a popup appears informing you that the train and test set are not compatible, hit **yes**.
13. Model can be found in our directory: **Classification/CorrelationAttributeEvalWithRandomForest.model**

Part 7 – Team Members and Tasks Performed

Finding the Data & Building Proposal: Chetan Maviti and Kanishk Sivanandam
Preprocessing Initial Attempt: Chetan Maviti and Kanishk Sivanandam
Preprocessing & Project Update: Chetan Maviti and Kanishk Sivanandam
Non-Weka Attribute Selection Algorithm: Kanishk Sivanandam
Attribute Selection Algorithms and Classifiers: Chetan Maviti
Results Output: Chetan Maviti and Kanishk Sivanandam
Results Analysis: Chetan Maviti and Kanishk Sivanandam
Building Final Report: Chetan Maviti and Kanishk Sivanandam

Part 8 – References

Data: https://data.world/rohit0308/stroke-prediction-23
https://weka.sourceforge.io/doc.dev/weka/attributeSelection/CorrelationAttributeEval.html
https://www.saedsayad.com/oner.htm
https://rasbt.github.io/mlxtend/user_guide/classifier/OneRClassifier/
https://www.sartorius.com/en/knowledge/science-snippets/what-is-principal-component-analysis-pca-and-how-it-is-used
https://builtin.com/data-science/step-step-explanation-principal-component-analysis
https://www.mathworks.com/help/stats/relieff.html
https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://medium.com/nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell
https://weka.sourceforge.io/doc.dev/weka/classifiers/rules/DecisionTable.html
https://www.cs.cornell.edu/courses/cs4780/2024sp/lectures/pdfs/lecturenote05.pdf
https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0