Project Report: Heart Attack, Accident, and Death Prediction

Raman Y. Avarzaman i6356849 05/09/2024
GitHub

Problem Statement

The goal of this project is to design both a **Rule-Based Decision Support System** and a **Machine Learning Solution** to predict:

1. **Heart Attacks**
2. **Accidents**
3. **Deaths**

The system analyzes patient data and predicts these events based on historical data, considering factors such as hypertension scores, alertness, intoxication levels, activity types (e.g., working, sleeping, drinking coffee, drinking alcohol), and other events like heart attacks, accidents, and deaths.

Data Analysis

The dataset contains **200 hourly 100-day patient histories**, where the following attributes are recorded for each patient:

- **Hypertension score**
- **Alertness level**
- **Intoxication level**
- **Activity**: Work, sleep, drink coffee, drink alcohol, do nothing
- **Events**: Heart attack, accident, death (marked as "patient died" in the `action` column)

The dataset contains real-world, noisy data where deaths are recorded under the `action` column, making it a challenge to extract information.

Here are some **descriptive statistics** for the dataset:

Description	Alertness	Hypertension	Intoxication	Smoker	Overweight	Family history	Goofball
count	344065	344065	344065	344065	344065	344065	344065
mean	0.757026	0.301734	0.024912	0.494478	0.510914	0.502995	0.485465
std	0.914594	0.299578	0.029532	0.280098	0.29417	0.285025	0.297668
min	-26.97758	0	0	0.002733	0.000186	0.005628	0.000335
25%	0.534651	0.052048	0.001039	0.261081	0.267065	0.270245	0.215807
50%	0.875667	0.215006	0.014287	0.477806	0.501043	0.499253	0.489028
75%	1.137058	0.464495	0.038676	0.730244	0.765045	0.75369	0.750683
max	8.005291	1.968238	0.220624	0.996607	0.996473	0.994115	0.999426

- Number of rows where heart_attack is True: 596
- Number of rows where heart_attack is False: 343469
- Number of rows where accident is True: 2606
- Number of rows where accident is False: 341459
- Number of rows where action is not 'patient died': 344015
- Number of rows where action is 'patient died': 50

General Challenges:

- **Event Labeling**: The challenge was to design a system that predicts heart attacks, accidents, and deaths with the available features, where death is not a direct feature but needs to be inferred from the `action` column.
- **Designing the Machine Learning Solution:** Creating an effective machine learning model required addressing issues like feature selection, imbalanced data, and evaluating different algorithms for event detection.
- **Limited Flexibility**: Rule-based systems generally do not handle edge cases well and might miss important patterns in the data that a machine learning model could detect.

Rule-Based Detector Solution

Approach

The rule-based detector was built based on logical assumptions and strict criteria derived from data patterns. It operates as a **bounded rational agent**, meaning that the rules need to be well-defined but can be overly strict, potentially leading to missed positive cases.

Key Rules (extracted by observing the training dataset):

- **Heart Attack Prediction**:
  - If hypertension score > 1.0 and the patient is NOT drinking alcohol or coffee, there is an increased chance of a heart attack.
- **Accident Prediction**:
  - If the intoxication score > 0.1 and alertness score < 0, predict an accident.
- **Death Prediction**:
  - If hypertension score > 1.4 and the patient is NOT drinking alcohol or coffee, and intoxication score > 0.09, there is an increased chance of a death.

Rule-Based Code Fragment:

```
def rule_based_predictions_with_death(row):
   # Initialize predictions
   heart_attack_pred = False
   accident_pred = False
   death_pred = False
   # Rule for predicting heart attack
   def check_heartattack():
       if row['hypertension'] > 1.0 and (row['action'] != "coffee" or row['action'] !=
"alcohol"):
           heart_attack_pred = True
   # Rule for predicting accident
   def check_accident():
       if row['alertness'] < 0.0 and row['intoxication'] > 0.1:
           accident_pred = True

   # Rule for predicting death
   if row['hypertension'] > 1.4 and row['action'] != "coffee" and row['action'] != "alcohol"
and row['intoxication'] > 0.09:
       death_pred = True

   return pd.Series([heart_attack_pred, accident_pred, death_pred],
                    index=['heart_attack_pred', 'accident_pred', 'death_pred'])
```

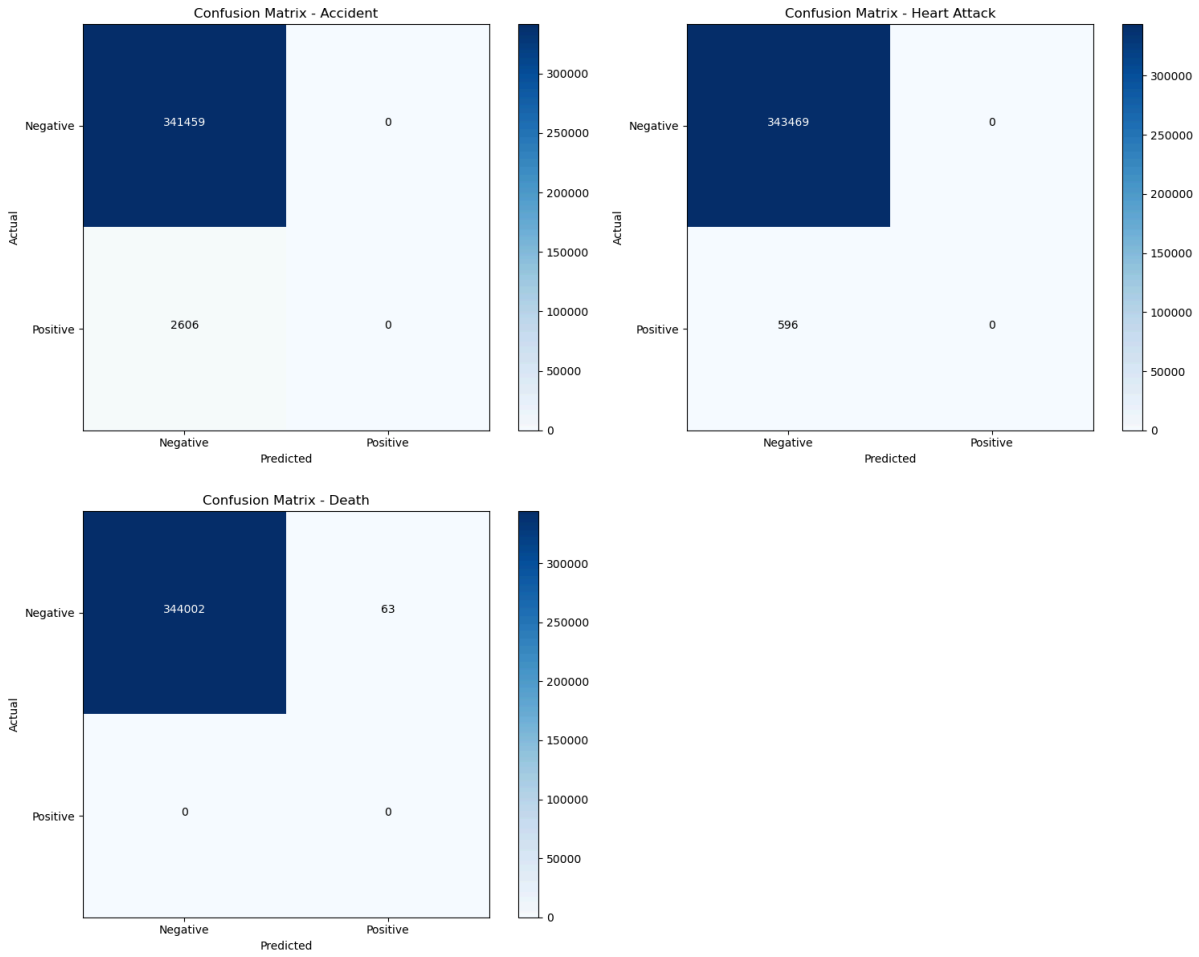
Performance

The rule-based system was **good at detecting negative cases**, but strict rules led to very few positive predictions:

- **False Positives**: Minimal, since the rules were stringent.
- **False Negatives**: High, especially for heart attacks and accidents, due to the strict nature of the rules.

Overall accuracy (all three correct): 0.9910 (the high accuracy is due to the data being mostly negative)

Confusion Matrices:



Machine Learning Solution

Approach

A machine learning-based solution using the **RandomForestClassifier** from the scikit-learn library was designed to predict heart attacks, accidents, and deaths. This approach overcomes the limitations of rule-based systems by learning patterns from the data.

Data Preprocessing:

- The **action** column, which contains activities like "work," "drink coffee," and "drink alcohol," was **One-Hot Encoded** to transform it into numerical features.
- Death information was extracted from the `action` column and treated as a binary target variable.

Feature Engineering:

- The following features were used to train the model:
  - Hypertension score
  - Alertness level
  - Intoxication level
  - Activity types (One-Hot Encoded)
  - Death (extracted from the `action` column)

Performance

The **machine learning model** significantly outperforms the rule-based system:

Heart Attack Prediction:

- **Accuracy**: 1.00
- **Sensitivity (Recall)**: 0.98
- **Specificity**: 1.00
- **Confusion Matrix Summary**: The model correctly predicted 103028 non-heart attack cases and 178 heart attack cases. There were 11 false positives and 3 false negatives.

Accident Prediction:

- **Accuracy**: 1.00
- **Sensitivity (Recall)**: 0.97
- **Specificity**: 1.00
- **Confusion Matrix Summary**: The model correctly predicted 102280 non-accident cases and 769 accident cases. There were 150 false positives and 21 false negatives.

Death Prediction:

- **Accuracy**: 1.00
- **Sensitivity (Recall)**: 0.97
- **Specificity**: 1.00
- **Confusion Matrix Summary**: The model correctly predicted 103029 non-death cases and 175 death cases. There were 10 false positives and 6 false negatives.

Confusion Matrices: