Efficiency Of The Drug For The Asthma Treatment Using Machine Learning Algorithms.

Abstract

Asthma causes numerous hospital encounters annually, including emergency department visits and hospitalizations. To improve patient outcomes and reduce the number of these encounters, predictive models are widely used. Machine Learning Algorithms intends to solve real- world problems in diagnosing and treating diseases.

Objective

The purpose of this case study is to find the efficiency of the drugs which the patients has been taking lately and to find out the effectiveness of the new drug as compared to the old one. I.e. skillful efficiency of the new drug and the old one of asthma using efficient algorithms of machine learning.

Introduction

Asthma is a variable long-term condition, affecting 339 million people worldwide, often with diurnal, seasonal and life-time differences in symptoms and disease burden. Although, for many, asthma symptoms are controlled most of the time, some have on-going poor control and all are at risk of attacks which, at best, are inconvenient and at worst can result in hospitalization or even death. Currently, there is no cure

for asthma, There is a need to understand the extent of machine learning that has been leveraged in the Healthcare industry.

Machine learning (ML) is a subclass of artificial intelligence technology, where algorithms process large data sets to detect patterns, learn from them, and execute tasks autonomously without being instructed on exactly how to address the problem. In recent years, the wide availability of powerful hardware and cloud computing has resulted in the broader adoption of ML in different areas of human lives, from using it for recommendations on social media to adopting it for process automation in factories. And its adoption will only grow further.

Healthcare is an industry that keeps up with the times as well. With the amount of data generated for each patient, machine learning algorithms in healthcare have great potential. So, that's no wonder that there are multiple successful machine learning applications in healthcare right now. So, here we are going to do the same.

The purpose of this operation is the resolution of real- world health problems in diagnosis and disease treatment.

METHODS AND MATERIALS

The process of solving this case study was modeled in steps to obtain the desired output.

Organizing input data:

The dataset which we have consists 21 columns, 18215 entries I.e. rows. Firstly, we will check whether the information have null values or not and we found out that there are no null values in our dataset.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18215 entries, 0 to 18214
Data columns (total 21 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   patid                             18215 non-null  int64
 1   index_age                         18215 non-null  int64
 2   previous_asthma_drugs             18215 non-null  int64
 3   total_pre_index_cannisters_365    18215 non-null  int64
 4   post_index_exacerbations365       18215 non-null  int64
 5   pneumonia                         18215 non-null  int64
 6   sinusitis                         18215 non-null  int64
 7   acute_bronchitis                  18215 non-null  int64
 8   acute_laryngitis                  18215 non-null  int64
 9   upper_respiratory_infection       18215 non-null  int64
 10  gerd                              18215 non-null  int64
 11  rhinitis                          18215 non-null  int64
 12  adherence                         18215 non-null  float64
 13  total_pre_index_charge            18215 non-null  float64
 14  pre_asthma_days                   18215 non-null  int64
 15  pre_asthma_charge                 18215 non-null  float64
 16  pre_asthma_pharma_charge          18215 non-null  float64
 17  drug_s                            18215 non-null  int64
 18  female                            18215 non-null  int64
 19  log_charges                       18215 non-null  float64
 20  log_asthma_charge                 18215 non-null  float64
dtypes: float64(6), int64(15)
memory usage: 2.9 MB
```

Data Pre-Processing and EDA:

Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models.

I have used Pairplot for to understand the best set of features to explain a relationship between two variables. It also helps to form some simple classification models by drawing some simple lines or make linear separation in our data-set. Here in this dataset we have no linear relationship with any of the feature and target column.

Heatmap when something has changed either in your feature column and you want to understand how that affects useability i.e to your target column.

Distplot to represents the overall distribution of continuous data variables you can see that there is a right skewness present in some columns and in some column the data is not normally distributed.

A pie chart to expresses a part-to-whole relationship of different drug consumed and their percentage and the drugs consumed by male and female count in the data.

For the data cleaning,i have created one function to categorized the data into 3 categories i.e 0,1 and 2. so that I can compare my data with pre drug count.

Dropped pid column because I find the relationship between patient's id and the efficiency of the drug are least compatible with each other.

Data sampling:

In the data-sampling step, a subset of data is extracted from the input dataset. This is performed for

sampling and for dividing the data into the two classes of training and testing data. I have kept test size is 0.30 I.e I am sending 30% of the data for testing and remaining 70% data for training.

I have created two models I.e one for Pre drug efficiency and another one for Post drug efficiency.

On Post drug data I have done Random Over Sampling Because the data is highly imbalance i.e it has an unequal number of observations.

Algorithms:

In this step, the parameters of the algorithms are tested and evaluated by using different values in order to find the best possible result in the system output.

I have used Random Forest, Decision Tree and Bagging classifier.

Random Forest Algorithm:

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Why did I choose to use Random Forest?

It takes less training time as compared to other algorithms.

It predicts output with high accuracy, even for the large dataset it runs efficiently.

It can also maintain accuracy when a large proportion of data is missing.

It enhances the accuracy of the model and prevents the overfitting issue.

Comparison of pre-index I.e it refers to activity or characteristics of the patient that existed before they were assigned to one of the drug groups and Post-index refers to measurements of activity after the drug assignment

Pre-Drug Model:

```
mymodel(RandomForestClassifier())
```

	precision	recall	f1-score	support
0	0.67	0.82	0.74	2146
1	0.75	0.70	0.72	2320
2	0.68	0.45	0.54	999
accuracy			0.70	5465
macro avg	0.70	0.65	0.66	5465
weighted avg	0.70	0.70	0.69	5465

Post Drug Model:

```
mymodel1(RandomForestClassifier())
```

	precision	recall	f1-score	support
0	0.98	0.97	0.97	2744
1	0.97	0.98	0.97	2729
accuracy			0.97	5473
macro avg	0.97	0.97	0.97	5473
weighted avg	0.97	0.97	0.97	5473

Decision Tree Algorithm:

It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

Why did I choose to use Decision Tree?

It is simple to understand as it follows the same process which a human follow while making any decision in real-life.

It can be very useful for solving decision-related problems.

It helps to think about all the possible outcomes for a problem.

There is less requirement of data cleaning compared to other algorithms.

Pre-Drug Model:

```
mymodel(DecisionTreeClassifier())
```

	precision	recall	f1-score	support
0	0.67	0.67	0.67	2146
1	0.67	0.67	0.67	2320
2	0.50	0.50	0.50	999
accuracy			0.64	5465
macro avg	0.61	0.61	0.61	5465
weighted avg	0.64	0.64	0.64	5465

Post Drug Model:

```
mymodel1(DecisionTreeClassifier())
```

	precision	recall	f1-score	support
0	0.94	0.95	0.94	2744
1	0.95	0.94	0.94	2729
accuracy			0.94	5473
macro avg	0.94	0.94	0.94	5473
weighted avg	0.94	0.94	0.94	5473

Bagging Algorithm (Ensemble Learning Technique):

Bagging, also known as Bootstrap aggregating, is an ensemble learning technique that helps to improve the performance and accuracy of machine learning algorithms. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model.

Why did I choose to use Bagging (Ensemble Learning Technique)?

Bagging minimizes the overfitting of data

It improves the model's accuracy

It deals with higher dimensional data efficiently

Pre-Drug Model:

```
from sklearn.ensemble import BaggingClassifier
mymodel(BaggingClassifier())
```

	precision	recall	f1-score	support
0	0.67	0.79	0.73	2146
1	0.72	0.70	0.71	2320
2	0.63	0.45	0.52	999
accuracy			0.69	5465
macro avg	0.67	0.64	0.65	5465
weighted avg	0.69	0.69	0.68	5465

Post Drug Model:

```
mymodel1(BaggingClassifier())
```

	precision	recall	f1-score	support
0	0.97	0.97	0.97	2744
1	0.97	0.97	0.97	2729
accuracy			0.97	5473
macro avg	0.97	0.97	0.97	5473
weighted avg	0.97	0.97	0.97	5473

Results:

As we can see from the above records, I have attached that the Efficiency/Accuracy of the Post Drug Consumption I.e patients who has been taking drug_s or drug_d is greater than the Pre Drug Consumption I.e they were taking some other drugs before.

So, the previously taken drugs by the patient were less effective than the newly taken drugs by the patients.

Relevant Literature:

Artificial intelligence and machine learning (ML) promise to transform cancer therapies by accurately predicting the most appropriate therapies to treat individual patients. Here, we present an approach, named Drug Ranking Using ML (DRUML), which uses omics data to produce ordered lists of >400 drugs based on their anti-proliferative efficacy in cancer cells. To reduce noise and increase predictive robustness, instead of individual features, DRUML uses internally normalized distance metrics of drug response as features for ML model generation.