



Pattern Detection and Classifying



Background

According to the CDC:

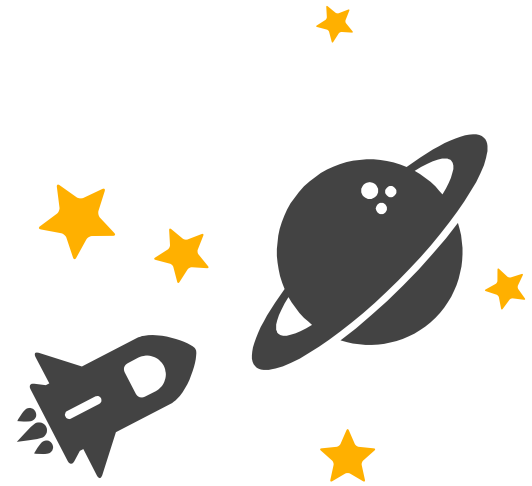
- Heart Disease is the leading cause of death in the US
- Finding symptoms can be difficult
- Can be dependent on one's lifestyle

Dataset Used



- Initial Dataset from CDC, surveying the American public.
- Too vast to be optimal.
- A preprocessed version of the same data found on Kaggle was chosen.

Problem



Use pattern detection and classification algorithms to predict the chances of getting a heart related disease given ones conditions

Preprocessing



Describing

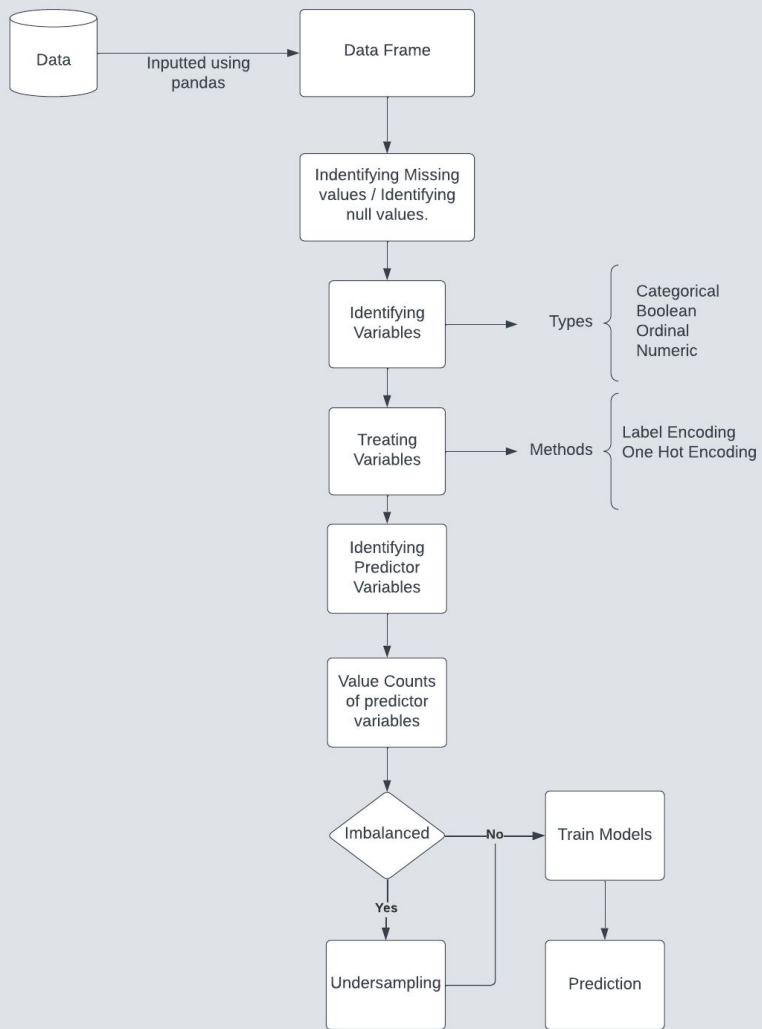
Functions from SKLearn was used to visualize and clearly describe the chosen data.

Treating Variables

Several encoding techniques from SKlearn was used to encode nominal and Ordinal data. Numerical data was normalized.

Sampling

Because the data was not balanced, Sampling Techniques was done to improve the final results.

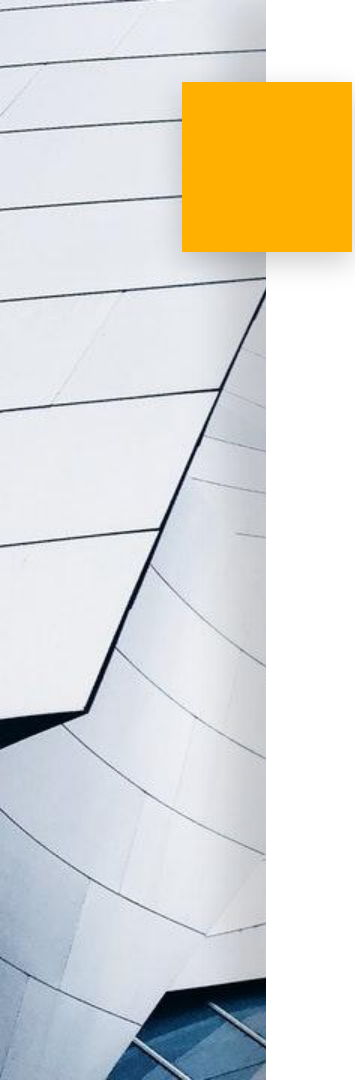


Flow chart



Logistic Regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable.

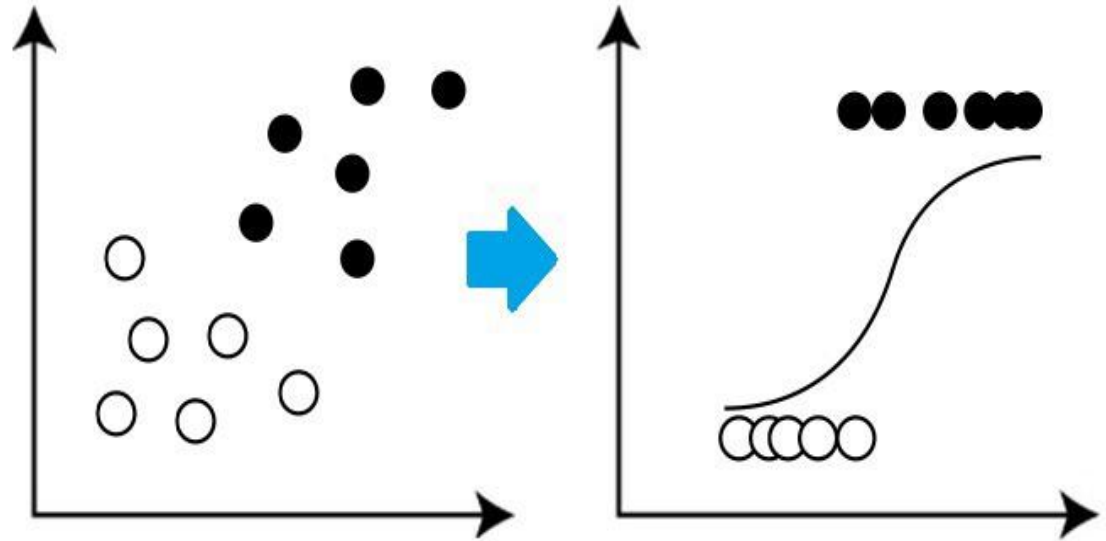


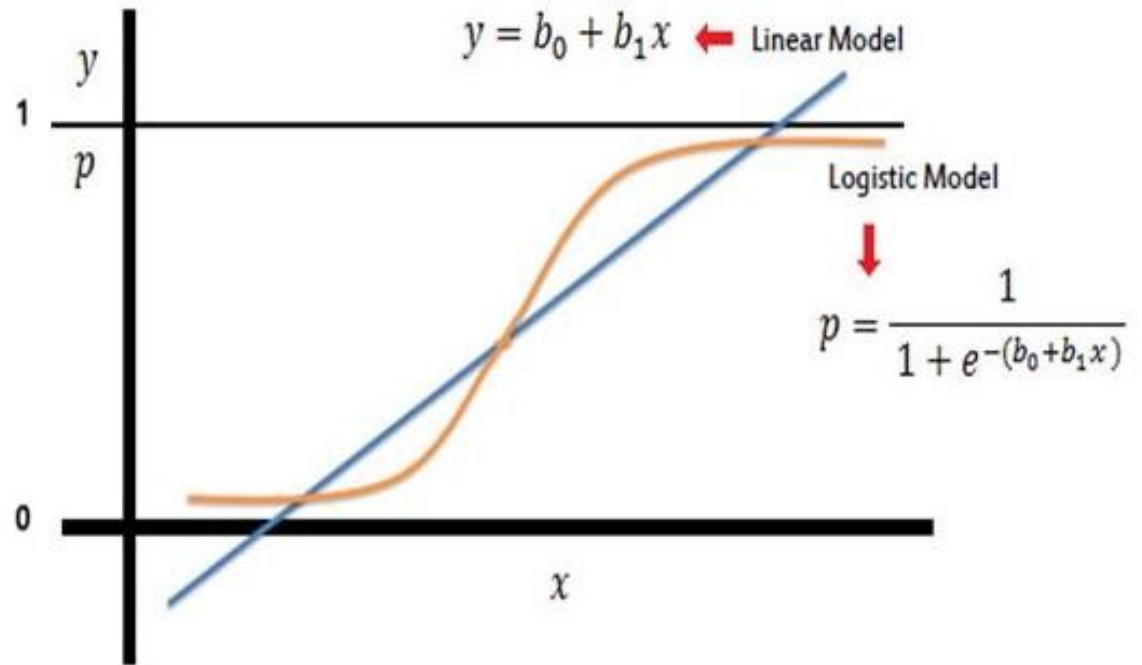
It is used to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities.

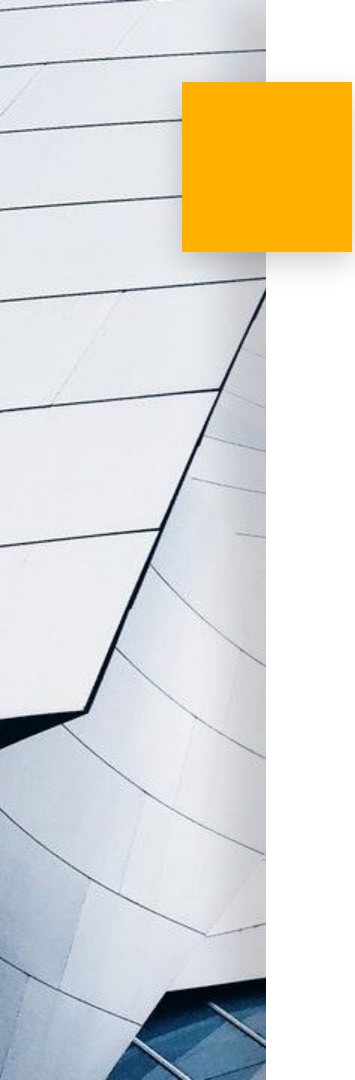
The most common logistic regression models a binary outcome.

It is effective in handling the outliers compared to the linear regression model.

LOGISTIC REGRESSION







The precision values obtained using the logistic regression

	Precision (Yes)	Precision (No)
Initial Dataset	0.52	0.92
Preprocesses d Dataset	0.76	0.77



Decision tree

- Decisions tree is popular in machine learning, due to its simple way of structuring it's model.
- Decision tree models require less data cleaning in comparison to other machine learning models.
- Features are selected on nodes, based on the statistic measure like information gain



Decision tree:

- Major drawback for decision tree is overfitting
- Decision tree needs pruning to refine decision trees and overcome the potential of overfitting.
- Prone to bias in machine learning models if the training dataset isn't balanced or representative.

Decision tree

	Precision (Yes)	Precision (No)
Initial Dataset	0.55	0.92
Pre Processed Dataset	0.76	0.7

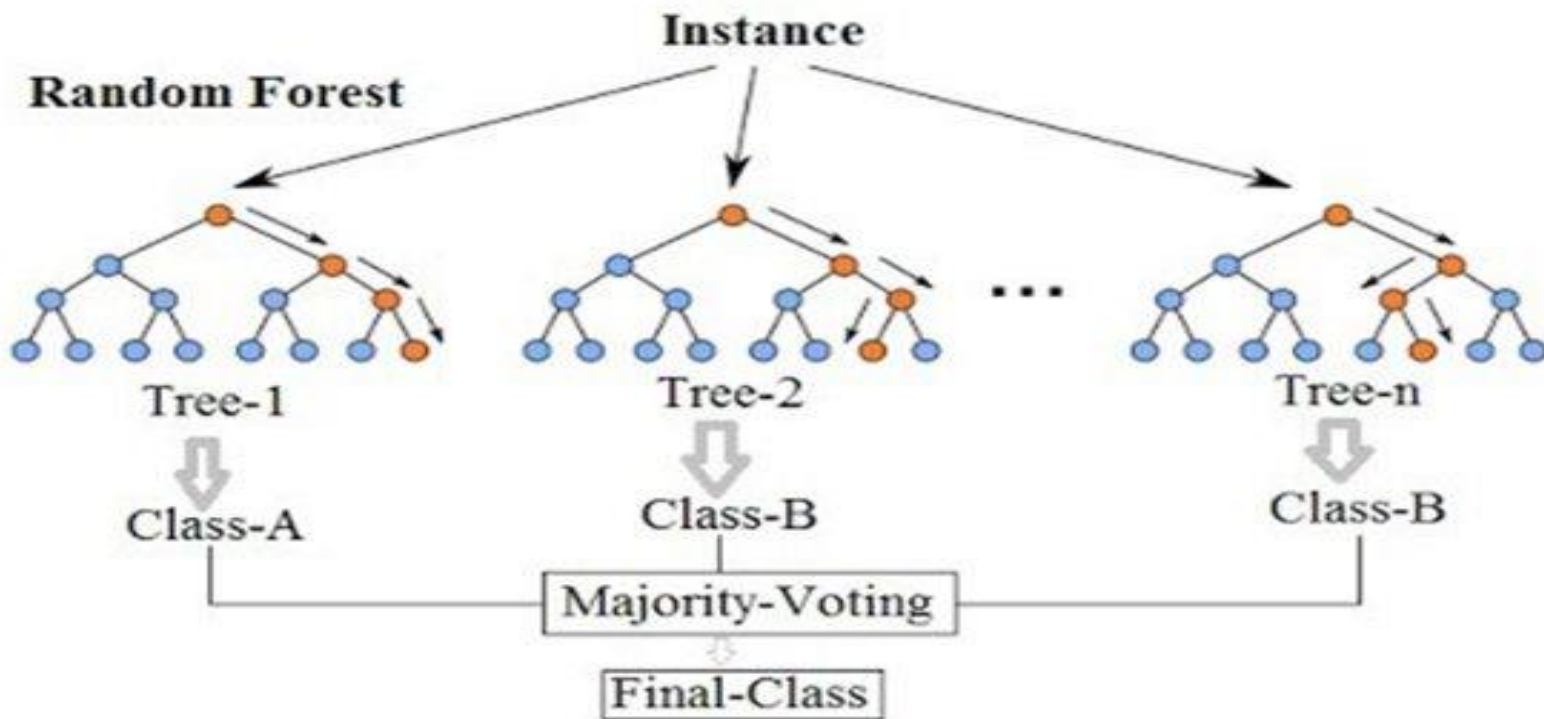
Decision tree

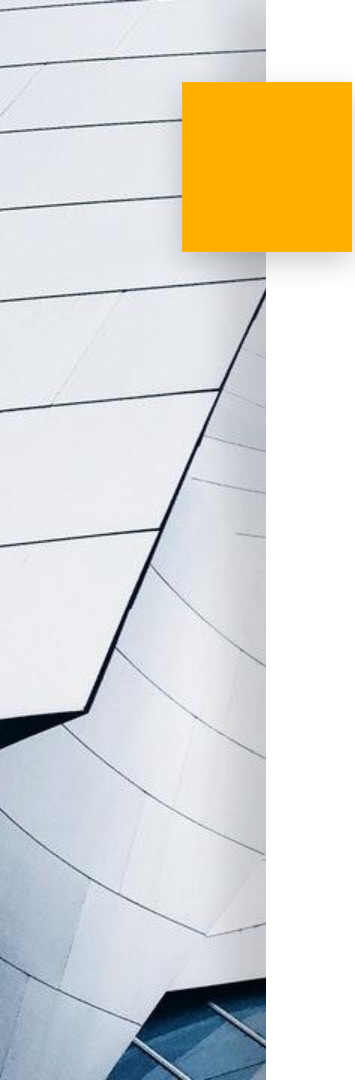
	Recall(Yes)	Recall(No)
Initial Dataset	0.06	0.9
Pre Processed Dataset	0.77	0.73

Random forest Algorithm

- ❑ Random forest is identified as a collection of decision trees. Each tree estimates a classification, and this is called a “vote”. Ideally, we consider each vote from every tree and chose the most voted classification (Majority-Voting).
- ❑ Random Forests produce many unique trees.

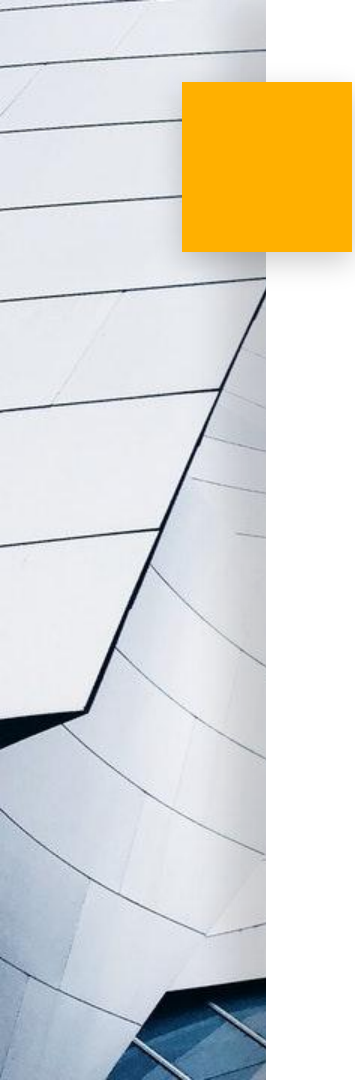
Random Forest Simplified





The precision values obtained using the Random Forest Classifier

	Precision (Yes)	Precision (No)
Initial Dataset	0.36	0.92
Preprocessed Dataset	0.93	1



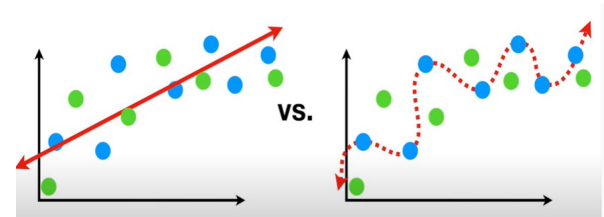
We have used Bagging technique to enhance the results of classification

	Precision	Recall
Yes	0.98	1
No	1	0.98

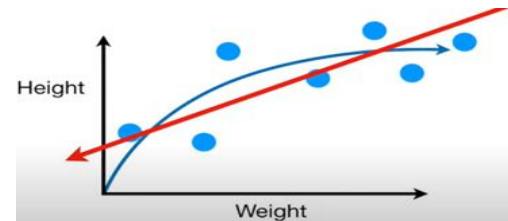
Support Vector Implementation



- Bias : The inability of model to capture true relationship
- Variance: Difference in Fits on data
- Soft Margin: The difference between observations and threshold
- Various Kernels
- Kernel Function



Images Referred from stat quest channel (youtube)



Support Vector Implementation



- Test_train_split from sklearn was used to split data into test and training datasets.
- Instance of sklearn.svm was created.
- Fit train data into the classifier.
- Measure the accuracy.
- Undersampling used
- Tune the Hyper parameters using Grid search CV

	Precision (Yes)	Precision (No)
Initial Dataset	0.64	0.92
Pre Processed Dataset	0.76	0.79

```
print(grid.best_params_)
```

```
print(grid.best_estimator_)
```

```
{'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}  
SVC(C=10, gamma=0.01)
```



Naive Bayes



A simple probability classifying algorithm, based on the Bayes Theorem.

- Find the probability of something (A) given that (B) has occurred.
- Assumed that features are independent.
- All features are considered to have equal weight.
- Runs comparatively faster than other algorithms.

Naive Bayes Implementation



- Test_train_split from sklearn was used to split data into test and training datasets.
- Instance of gaussian naive bayes was created.
- Fit train data into the classifier.
- Measure the accuracy.

	Precision (Yes)	Precision (No)
Initial Dataset	0.26	0.95
Preprocesses d Dataset	0.77	0.67

Model	Accuracy	Precision for No(0)	Precision for Yes(1)	Recall for No(0)	Recall for Yes(1)
Logistic regression	0.91	0.92	0.52	0.99	0.1
Decision tree	0.91	0.92	0.55	1	0.06
Support Vector Machine	0.92	0.92	0.64	1	0.02
Naive Bayes	0.84	0.95	0.26	0.88	0.47
Random Forest	0.91	0.92	0.36	0.98	0.12

Model	Accuracy	Precision No	Precision Yes	Recall No	Recall Yes
Logistic regression	0.76	0.77	0.76	0.75	0.78
Decision tree	0.75	0.76	0.74	0.73	0.77
Support Vector Machine	0.77	0.79	0.76	0.75	0.80
Naive Bayes	0.71	0.67	0.77	0.82	0.60
Random Forest	0.96	1	0.93	0.93	1



Thank You