

MALWARE DETECTION BY ENSEMBLE CLASSIFIERS

Raj Sahu
BTECH/10363/18
CSE A

Introduction

Every other website in today's age on the internet wants to collect data of its users by tricking them into giving away their credentials for fraud or many such vindictive acts. Naive users using a browser have no idea about the backend of the page. The users might be tricked into giving away their credentials or downloading malicious data. This Problem can be tackled effectively by pre-classifying the website as Phishy or Safe using Ensemble Classifiers .

Dataset : Phishing Websites Dataset - UCI ML REPOSITORY

Features Used :

- Length of url.
- Number of sub domains.
- HTTPS website.
- Domain Registration
- Favicon
- Web Traffic
- Abnormal Url
- Submit to email
- Age of domain
- Indexed by google
- Ip Address.
- Shortening service.
- @ symbol.
- Redirection "//"
- SFH
- % of External Objects.
- DNS
- No of '-' symbol.
- % of External or object links.

Ensemble Learning

Ensemble models in machine learning combine the decisions from multiple models to improve the overall performance.

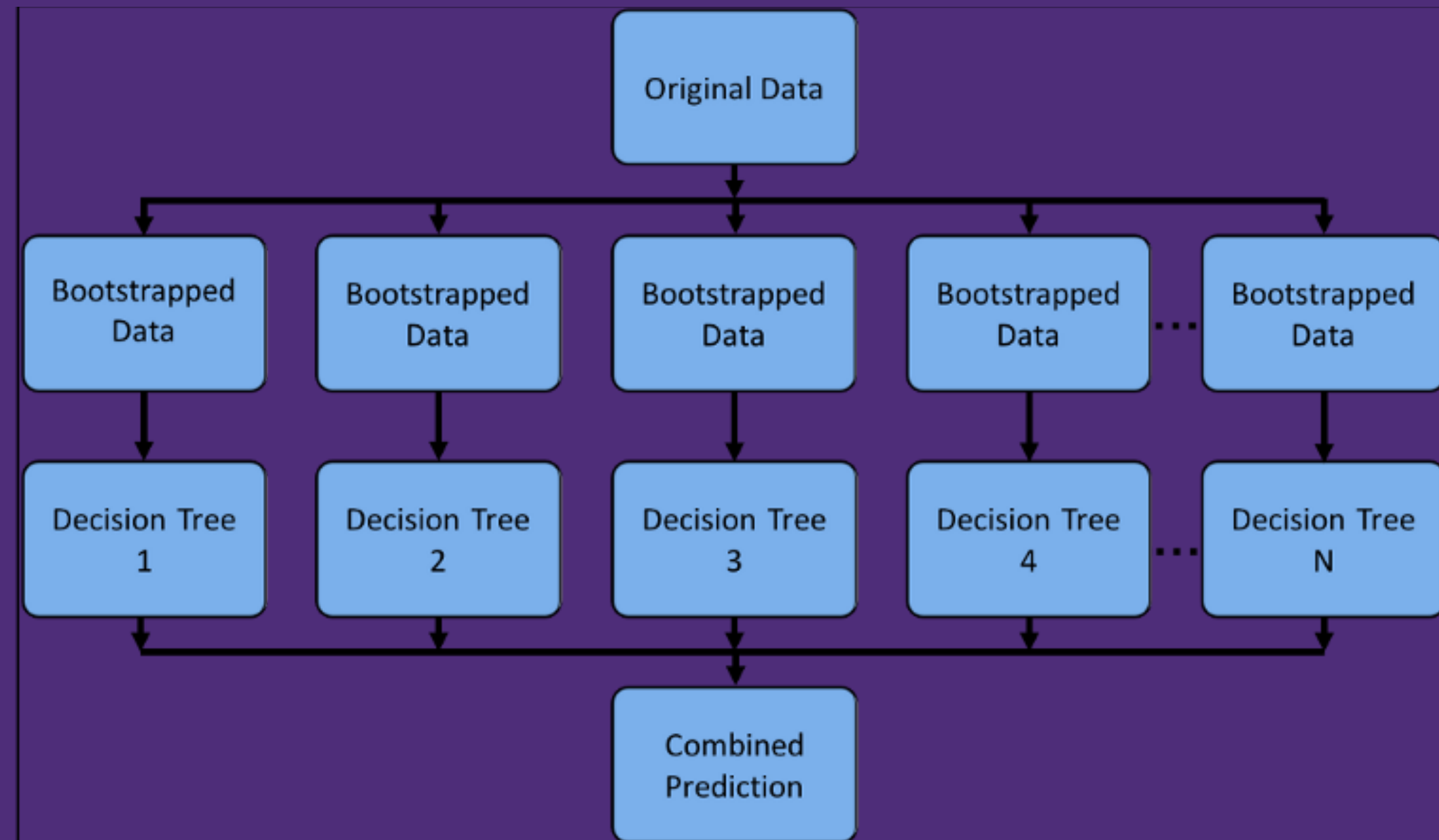
Ensembling Techniques :

- 1. Bagging (models are built independently and the results are aggregated for final predictions).*
- 2. Boosting (models are built sequentially and the results are aggregated for final predictions).*
- 3. Stacking (Independently trained models are stitched together using a meta-learner).*



Random Forests – Bagging

- * Random forest is an ensemble of decision tree algorithms. It is an extension of bootstrap aggregation (bagging) of decision trees.
- * A number of decision trees are created where each tree is created from a different bootstrap sample of the training dataset



Random Forests – Bagging

- * Each decision tree is fit on a slightly different training dataset, and in turn, has a slightly different performance.
- * Unlike normal decision tree models, such as classification and regression trees, trees used in the ensemble are unpruned, making them slightly overfit to the training dataset.
- * This is desirable as it helps to make each tree more different and have less correlated predictions or prediction errors.
- * Final prediction is the majority vote class label predicted across the decision trees.
- * Random Forest was implemented using the scikit-learn Library with 100 base estimators.

GBDT – Boosting



- * Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until the training set is predicted perfectly or a maximum number of models are added.

p = probability

$\frac{p}{1-p}$ = corresponding odds

$$\text{Probability} = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$

1. Initialize model with a constant value:

$$F_0(x) = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, \gamma).$$

2. For $m = 1$ to M :

1. Compute so-called *pseudo-residuals*:

$$r_{im} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \dots, n.$$

2. Fit a base learner (or weak learner, e.g. tree) $h_m(x)$ to pseudo-residuals, i.e. train it using the training set $\{(x_i, r_{im})\}_{i=1}^n$.

3. Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg \min_{\gamma} \sum_{i=1}^n L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)).$$

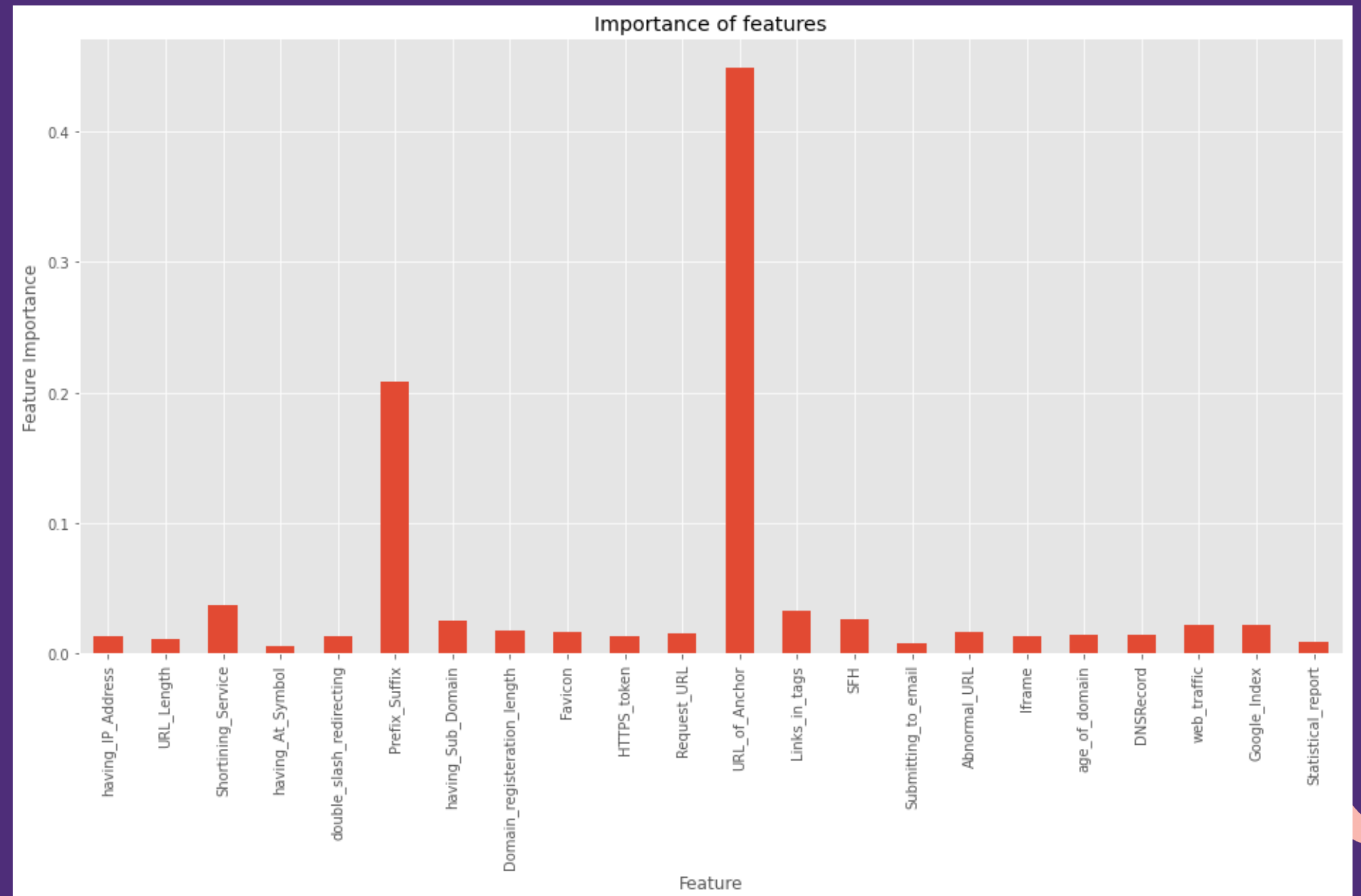
4. Update the model:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x).$$

3. Output $F_M(x)$.

GBDT – Boosting

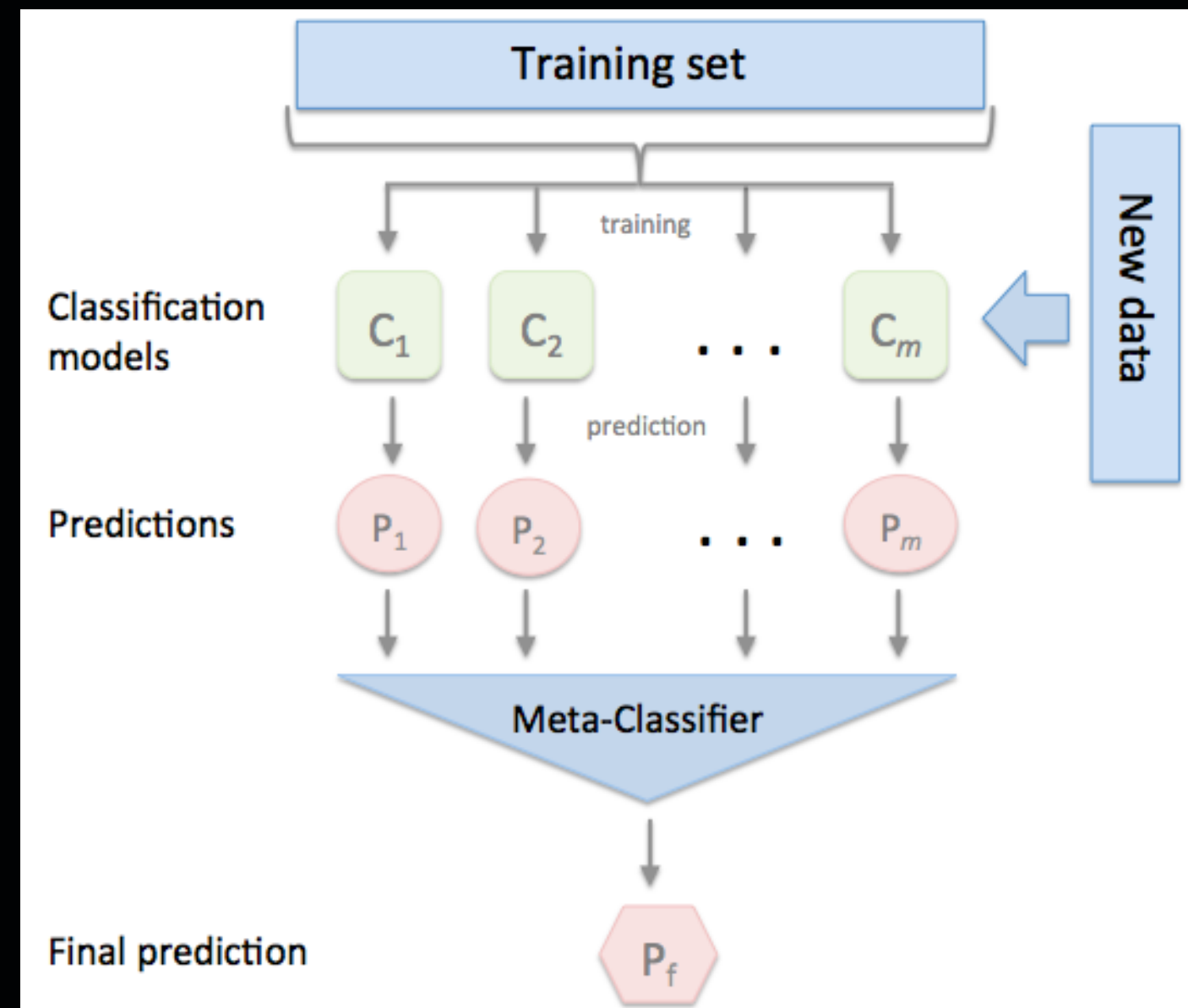
- * Gradient boosting decision tree algorithm was used along with row and column sampling using the Xgboost Library.
- * Type of importance is the average gain of splits which use the feature.

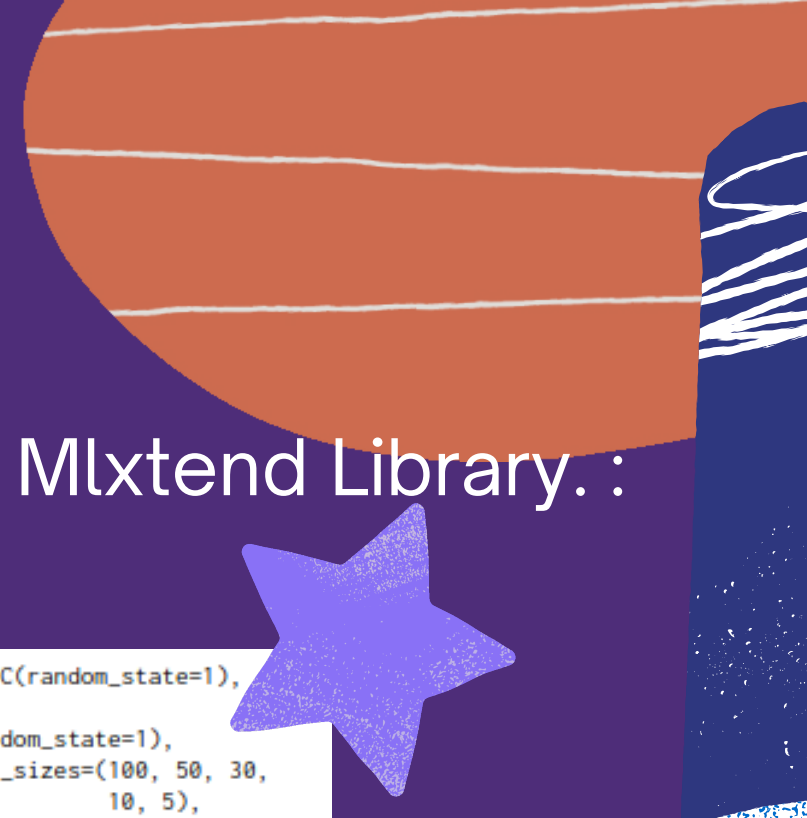


Stacking Classifier :



- * Stacking is an ensemble learning technique to combine multiple classification models which were trained independently using a meta-classifier
- * The individual classification models are trained based on the complete training set; then, the meta-classifier is fitted based on the outputs – meta-features – of the individual classification models in the ensemble.





Mlxtend Library.:

```
C(random_state=1),  
dom_state=1),  
_sizes=(100, 50, 30,  
10, 5),
```

-



Results :

Accuracy Scores :			
Model	Train Accuracy	Test Accuracy	Δ
Random Forests	97.02%	93.89%	3.14%
Stacking Classifier	96.94%	94.28%	2.65%
GBDT	95.92%	94.54%	1.39%
$\Delta \implies Training Accuracy - Test Accuracy$			

Train Accuracy = 96.09% Test Accuracy = 92.44% Δ = 3.65% [KNN]
Train Accuracy = 92.94% Test Accuracy = 91.93% Δ = 1.01% [SVC]
Train Accuracy = 58.65% Test Accuracy = 56.04% Δ = 2.61% [Naive Bayes]
Train Accuracy = 97.02% Test Accuracy = 93.89% Δ = 3.14% [Random Forests]
Train Accuracy = 96.86% Test Accuracy = 94.18% Δ = 2.69% [MLP]
Train Accuracy = 95.92% Test Accuracy = 94.54% Δ = 1.39% [XG Boost]
Train Accuracy = 96.94% Test Accuracy = 94.28% Δ = 2.65% [StackingClassifier]



Thank you!