A

Major Project Report

On

# MACHINE LEARNING APPROACH FOR CLICK FRAUD DETECTION

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By


| | |
|---|---|
| **K. Teja Niranjan** | **(187R1A0589)** |
| **N. Madhuri** | **(187R1A05A3)** |
| **A. Vamshidhar Reddy** | **(197R5A0502)** |


Under the Guidance of

**Dr. G. Madhukar**

Associate Professor


**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.


**2018-2022**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# CERTIFICATE

This is to certify that the project entitled "**MACHINE LEARNING APPROACH FOR CLICK FRAUD DETECTION**" being submitted by **K. Teja Niranjan (187R1A0589), N. Madhuri (187R1A05A3), A. Vamshidhar Reddy(197R5A0502)** in partial fulfillment of the requirements for the award of the degree of B. Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2021-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

**Dr. G. Madhukar**
**Associate Professor**
**INTERNAL GUIDE**

**Dr. A. Raji Reddy**
**DIRECTOR**

**Dr. K. Srujan Raju**
**HOD**

**EXTERNAL EXAMINER**

**Submitted for viva voice Examination held on** ____________________

# ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express my profound gratitude and deep regard to my guide. **Dr. G. Madhukar**, Associate Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. J. Narasimha Rao, Dr. G. Madhukar, Dr. T. S. Mastan Rao, Dr. Suwarna Gothane, Mr. A. Uday Kiran, Mr. A. Kiran Kumar** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

**K. Teja Niranjan (187R1A0589)**

**N. Madhuri (187R1A05A3)**

**A. Vamshidhar Reddy (197R5A0502)**

# ABSTRACT

Click Fraud is a fraudulent act of clicking on pay-per-click advertisements to increase the site's revenue or to drain revenue from the advertiser. This illegal act has been putting commercial industries in a dilemma for quite some time. These industries think twice before advertising their products on websites and mobile-apps, as many parties try to exploit them. To safely promote their products, there must be an efficient system to detect click fraud. The proposed model, classified under supervised machine learning, is a combination of two learning models used for feature transformation and classification. We showcase its superior performance compared to other related models, and make a comparison with multiple click fraud datasets with varying sizes.

Fraud users just visit website and don't do any operation such as app downloading or form filling or any other process to make more money. To detect such fraud click we are implementing machine learning approach

# LIST OF FIGURES/TABLES

# LIST OF SCREENSHOTS

# TABLE OF CONTENTS

# 1. INTRODUCTION

# 1. INTRODUCTION

## 1.1 PROJECT SCOPE

The explosive growth in the size and use of the World Wide Web continuously creates new great challenges and needs. One such need is dealing with click fraud, which aims at increasing clicks on certain ads and thus the profit of the websites which display them. In this work, we extend the concept of click fraud and redefine it as any pattern of clicks whose goal is to alternate the normal operation of a website in order to produce specific results.

An indication of a click fraud may be a burst of clicks that can be simulated by an automated program or script. We deal with the problem of efficient real-time Click Fraud detection utilizing advanced data structures and exploiting their advantages concerning space and time required.

## 1.2 PROJECT PURPOSE

Click fraud is the illegal clicking of advertisements that leads to the wasted funds of the advertisers, and to counter this issue, several methods to detect click fraud have been devised. Click fraud detection is used to protect the advertiser by classifying clicks into valid and fraudulent clicks

## 1.3 PROJECT FEATURES

Machine learning and data-driven approaches are becoming very important in many areas. Smart spam classifiers protect our email by learning from massive amounts of spam data and user feedback; advertising systems learn to match the right ads with the right context; fraud detection systems protect banks from malicious attackers; anomaly event detection systems help experimental physicists to find events that lead to new physics.

There are two important factors that drive these successful applications: usage of effective (statistical) models that capture the complex data dependencies and scalable learning systems that learn the model of interest from large datasets

# 2. SYSTEM ANALYSIS

# 2. SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analysed. The system analyst plays an important role of an interrogator and wells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, "what must be done to solve the problem?" The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

## 2.1 PROBLEM DEFINITION

Click fraud refers to the fraudulent clicking on a pay-per-click advert, which is designed to divert or negatively impact the advertiser's budget. Several parties commit click fraud, and to understand who might be fraudulently clicking on ads, we look at the three most common offenders: Competitors, Webmasters and Fraud rings. Google AdWords is an advertising network that has a system in place to detect click fraud. Google performs an in-depth investigation of the complaints from advertisers. As seen by these steps, the whole process of detecting click fraud is not entirely automated

## 2.2 EXISTING SYSTEM

Due to the growth in web technologies and media, advertising companies have shifted focus from conventional newspapers and televised advertisements to online and in-app advertisements in order to attract new customers. For Internet giants such as Google, Yahoo and Facebook, the largest revenue source is Online Advertising. These giants are advertising networks.

However, in this payment model, there exists a security risk called Click Fraud. In 2017, about 1 in 5 clicks were fraudulent clicks and in smartphones, they increased by two times in four months (ppc, 2019). These click fraud statistics show that the practice has only been growing and that a significant chunk of internet traffic is fraudulent. Whatever form click fraud takes; the result is that advertisers are always under financial loss.

### 2.2.1 DISADVANTAGES OF EXISTING SYSTEM

- We cannot find genuine users for the website

- It causes a revenue loss for the website
- There will not be any growth in the website

## 2.3 PROPOSED SYSTEM

The proposed model, classified under supervised machine learning, is a combination of two learning models used for feature transformation and classification. We showcase its superior performance compared to other related models, and make a comparison with multiple click fraud datasets with varying sizes. Click fraud refers to the fraudulent clicking on a pay-per-click advert, which is designed to divert or negatively impact the advertiser's budget. Several parties commit click fraud, and to understand who might be fraudulently clicking on ads, we look at the three most common offenders: Competitors, Webmasters and Fraud rings

### 2.3.1 ADVANTAGES OF PROPOSED SYSTEM

- We can find genuine users for the website
- It causes high revenue for the website
- There will be growth in the website

## 2.4 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is done. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Behavioural Feasibility

### 2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefit in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

### 2.4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### 2.4.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

## 2.5 HARDWARE AND SOFTWARE REQUIREMENTS

### 2.5.1 HARDWARE REQUIREMENTS

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements.

- System: Intel Quad Core@ CPU 2.90GHz.
- Hard Disk: 120 GB
- Input Devices: Keyboard, Mouse
- Ram: 2GB

## 2.5.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

- Operating system: Windows 7,8,10
- Languages: Python
- IDE: Jupiter notebook

# 3. ARCHITECTURE

# 3. ARCHITECTURE

## 3.1 PROJECT ARCHITECTURE

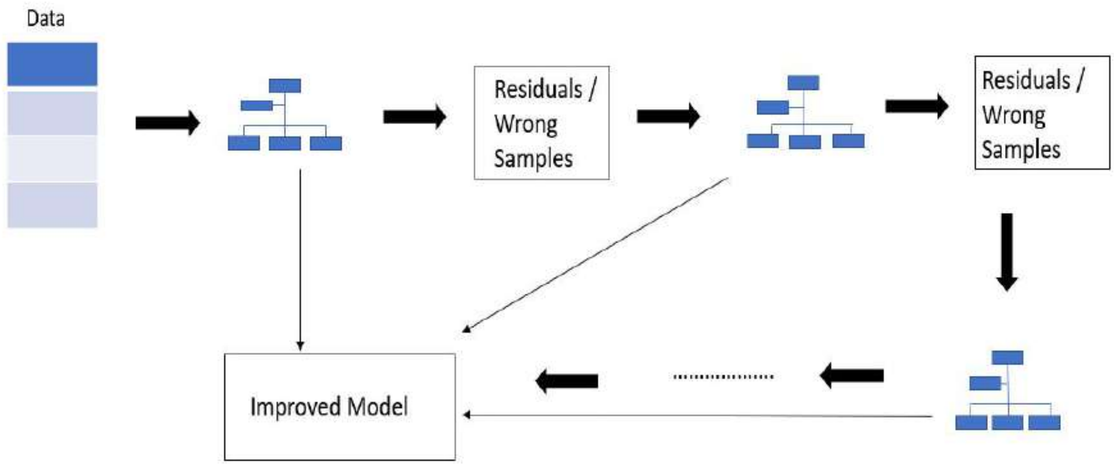



Figure 3.1 Project Architecture for click fraud detection

The project architecture represents the full functionality of the click fraud detection project program. First, we collect data from various sources such as websites and Kaggle. Then remove the noisy data and try to pre-process the data. After the pre-processing is complete, it tries to apply the decision tree algorithm to the dataset. Therefore, after application, you will get two results. For example, if you get the correct results, try applying a decision tree algorithm to this data. They are added to the improved result collection, incorrect samples are reprocessed, and the process continues until reasonable accuracy is found.

## 3.2 USE CASE DIAGRAM

Its purpose is to present a graphical overview of the functionality provided by a system in terms of actions, their goals represented as use cases and any dependencies between those use cases. Here the functionality of the model is to collect the data from a dataset for training the data. The developer will do all these use cases like collecting data, create model, fit model, access server to predict and end user will deploy the model.

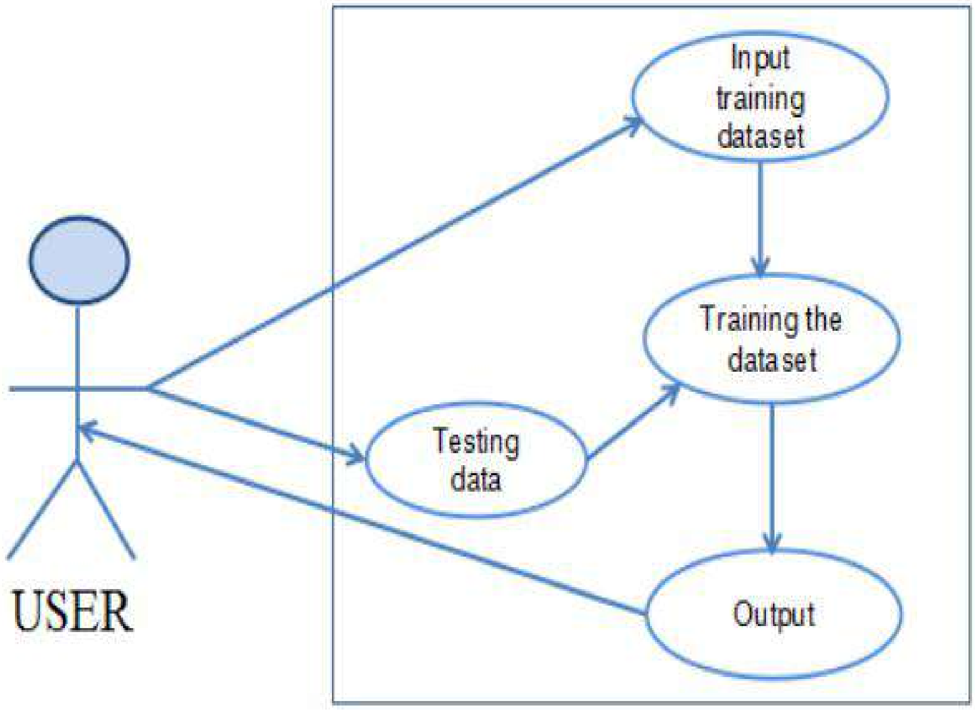



Figure 3.2 use case diagram for click fraud detection

## 3.3 CLASS DIAGRAM

It describes about the structure by showing the system classes, their attributes, operations and the relationship among the classes. It explains about the information of the classes.

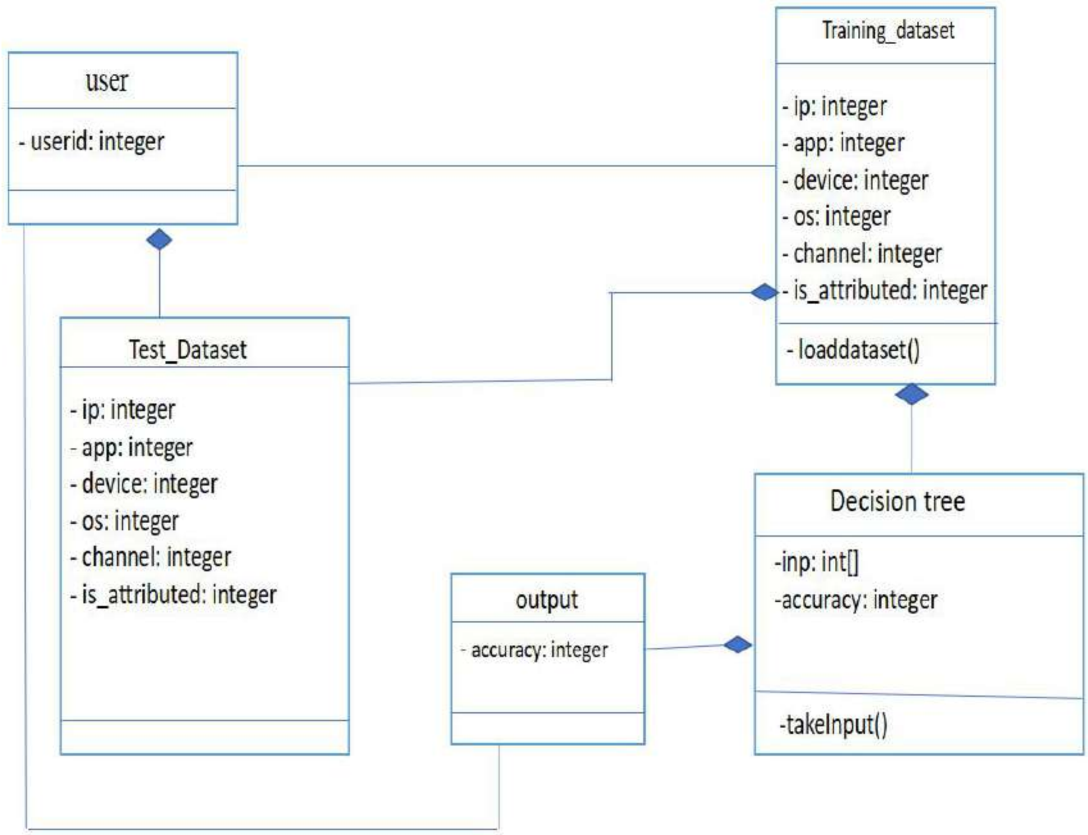



Figure 3.3 class diagram for click fraud detection

## 3.4 SEQUENCE DIAGRAM

It is used to represent the objects that are participating the interaction horizontally and time vertically. The sequence of the messages between the objects will show the functionality carried out in the model. Each use case specifies some behaviour, possibly including variants that the subject can perform in collaboration with one or more.

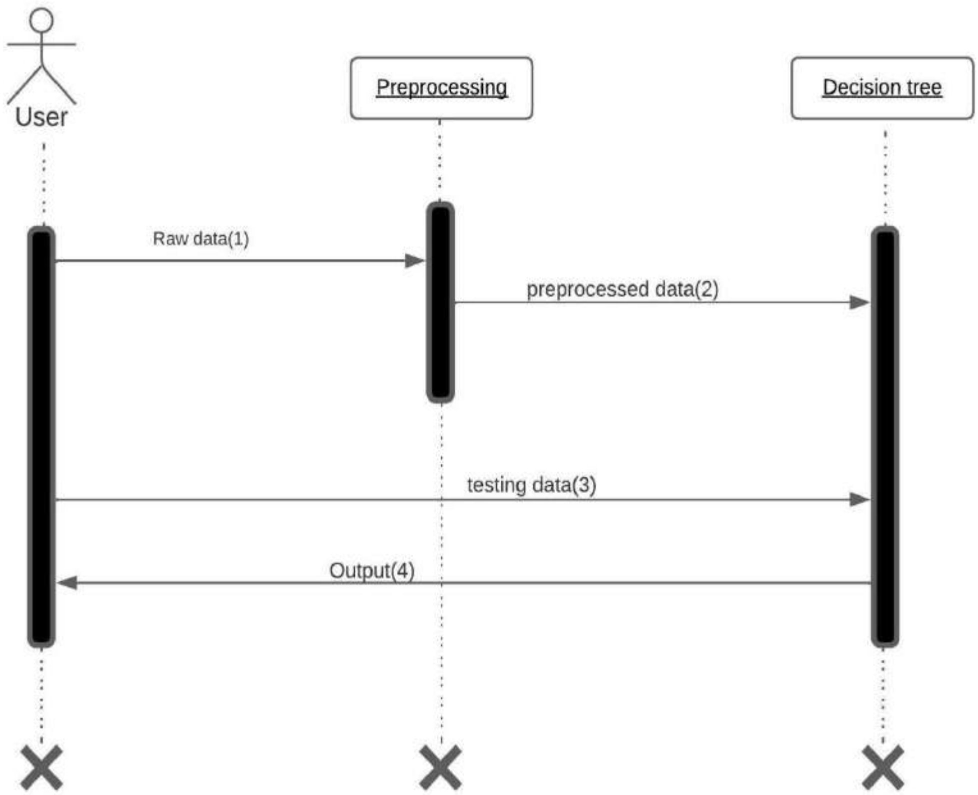



Figure 3.4 sequence diagram for click fraud detection

## 3.5 ACTIVITY DIAGRAM

The diagram basically describes a program control overflow. The first step in your project is to fetch the dataset and remove all kinds of errors, missing values and noisy data. This is sometimes referred to as data pre-processing. After the data has been processed, it will try to split the data. Training and test datasets that try to apply the decision tree algorithm individually. After applying these algorithms, you will get two types of results for both the test and training datasets, and these results will be compared in the next step. These steps of applying the algorithm to get the values continue until you have the accuracy you need for your project.

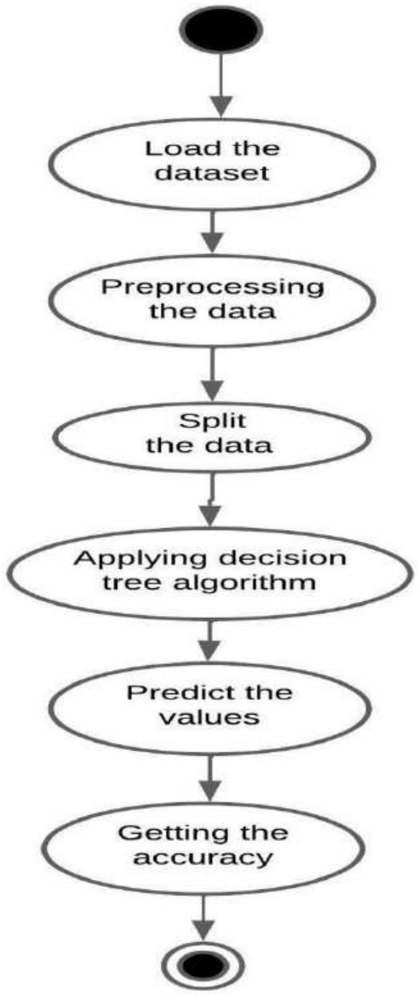



Figure 3.5 activity diagram for click fraud detection

# 4. IMPLEMENTATION

# 4. IMPLEMENTATION

## 4.1 SAMPLE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
from sklearn.preprocessing import LabelEncoder
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import ExtraTreesClassifier
import xgboost as xgb
from xgboost import XGBClassifier
from xgboost import plot_importance
import gc
import os
import warnings
warnings.filterwarnings('ignore')
dtypes={
'ip':'uint16','app':'uint16','device':'uint16','os':'uint16','channel':'uint16','ips_attributed':'uint
8','click_id':'uint32'
}
testing = True
if testing:
   train_path = "C:/Users/tejan/OneDrive/Desktop/Click/input/train_sample.csv"
   skiprows = None
   nrows = None
   colnames=['ip','app','device','os', 'channel', 'click_time', 'is_attributed']
else:
```

```
    train_path = "C:/Users/tejan/OneDrive/Desktop/Click/input/train.csv"
    skiprows = range(1, 144903891)
    nrows = 10000000
    colnames=['ip','app','device','os', 'channel', 'click_time', 'is_attributed']
train_sample = pd.read_csv(train_path, skiprows=skiprows, nrows=nrows, dtype=dtypes,
usecols=colnames)
len(train_sample.index)
print(train_sample.memory_usage())
print('Training dataset uses {0}
MB'.format(train_sample.memory_usage().sum()/1024**2))
train_sample.head()
train_sample.info()
def fraction_unique(x):
    return len(train_sample[x].unique())
number_unique_vals={x:fraction_unique(x) for x in train_sample.columns}
number_unique_vals
train_sample.dtypes
plt.figure(figsize=(14, 8))
sns.countplot(x="app", data=train_sample)
plt.figure(figsize=(14, 8))
sns.countplot(x="device", data=train_sample)
plt.figure(figsize=(14, 8))
sns.countplot(x="channel", data=train_sample)
plt.figure(figsize=(14, 8))
sns.countplot(x="os", data=train_sample)
100*(train_sample['is_attributed'].astype('object').value_counts()/len(train_sample.index))
app_target = train_sample.groupby('app').is_attributed.agg(['mean', 'count'])
app_target
frequent_apps=train_sample.groupby('app').size().reset_index(name='count')
frequent_apps=frequent_apps[frequent_apps['count']>frequent_apps['count'].quantile(0.8
0)]
frequent_apps=frequent_apps.merge(train_sample,on='app',how='inner')
frequent_apps.head()
plt.figure(figsize=(10,10))
```

```
sns.countplot(y="app", hue="is_attributed", data=frequent_apps);
def time_features(df):
   df['datetime']=pd.to_datetime(df['click_time'])
   df['day_of_week']=df['datetime'].dt.dayofweek
   df['day_of_year']=df['datetime'].dt.dayofyear
   df['month']=df['datetime'].dt.month
   df['hour']=df['datetime'].dt.hour
   return df
#Extra Tree Classifier with AdaBoost Classifier
tree = ExtraTreesClassifier(n_estimators=100, random_state=0)
adaboost_model_1=AdaBoostClassifier(
   base_estimator=tree,
   n_estimators=600,
   learning_rate=1.54,
   algorithm="SAMME")
adaboost_model_1.fit(X_train,y_train)
predictions=adaboost_model_1.predict_proba(X_test)
predictions[:10]
Extra_acc = metrics.roc_auc_score(y_test,predictions[:,1])
Extra_acc
param_grid={"base_estimator__max_depth":[2,5],
         "n_estimators":[200,400,600]
         }
```

**#Decision Tree With AdaBoost Classifier and Kfolds = 3**

```
tree1 = DecisionTreeClassifier()
ABC=AdaBoostClassifier(
   base_estimator=tree1,
   learning_rate=0.6,
   algorithm="SAMME")
folds=3
grid_search_ABC=GridSearchCV(ABC,
                     cv=folds,
```

```
                    param_grid=param_grid,
                    scoring='roc_auc',
                    return_train_score=True,
                    verbose=1)
grid_search_ABC.fit(X_train,y_train)
predictions=grid_search_ABC.predict_proba(X_test)
predictions[:10]
DT_acc = metrics.roc_auc_score(y_test,predictions[:,1])
DT_acc
cv_results = pd.DataFrame(grid_search_ABC.cv_results_)
cv_results
# plotting AUC with hyperparameter combinations
plt.figure(figsize=(16,6))
for n, depth in enumerate(param_grid['base_estimator__max_depth']):
    # subplot 1/n
    plt.subplot(1,3, n+1)
    depth_df = cv_results[cv_results['param_base_estimator__max_depth']==depth]
    plt.plot(depth_df["param_n_estimators"], depth_df["mean_test_score"])
    plt.plot(depth_df["param_n_estimators"], depth_df["mean_train_score"])
    plt.xlabel('n_estimators')
    plt.ylabel('AUC')
    plt.title("max_depth={0}".format(depth))
    plt.ylim([0.60, 1])
    plt.legend(['test score', 'train score'], loc='upper left')
    plt.xscale('log')
```

**#Random Forest Classifier**

```
RFC = RandomForestClassifier(max_depth=2, random_state=0)
RFC.fit(X_train, y_train)
predictions = RFC.predict_proba(X_test)
predictions[:10]
RF_acc = metrics.roc_auc_score(y_test,predictions[:,1])
```

```
RF_acc
plt.figure(figsize=(16,6))
for n, subsample in enumerate(param_grid['subsample']):
    # subplot 1/n
    plt.subplot(1,len(param_grid['subsample']), n+1)
    df = cv_results[cv_results['param_subsample']==subsample]
    plt.plot(df["param_learning_rate"], df["mean_test_score"])
    plt.plot(df["param_learning_rate"], df["mean_train_score"])
    plt.xlabel('learning_rate')
    plt.ylabel('AUC')
    plt.title("subsample={0}".format(subsample))
    plt.ylim([0.60, 1])
    plt.legend(['test score', 'train score'], loc='upper left')
    plt.xscale('log')
```

**#XGB Classifier**

```
model = XGBClassifier()
model.fit(X_train, y_train)
y_pred = model.predict_proba(X_test)
y_pred[:10]
xgb_acc = metrics.roc_auc_score(y_test, y_pred[:, 1])
print("AUC: %.2f%%" % (xgb_acc * 100.0))
```

**#XGbooster with Kfolds**

```
folds = 3
# specify range of hyperparameters
param_grid = {'learning_rate': [0.2, 0.6],
             'subsample': [0.3, 0.6, 0.9]}
# specify model
xgb_model = XGBClassifier(max_depth=2, n_estimators=200)
# set up GridSearchCV()
model_cv = GridSearchCV(estimator = xgb_model,
                        param_grid = param_grid,
                        scoring= 'roc_auc',
```

```
                    cv = folds,
                    verbose = 1,
                    return_train_score=True)
model_cv.fit(X_train, y_train)
cv_results = pd.DataFrame(model_cv.cv_results_)
cv_results
cv_results['param_learning_rate'] = cv_results['param_learning_rate'].astype('float')
cv_results.head()
plt.figure(figsize=(16,6))
param_grid = {'learning_rate': [0.2, 0.6],
          'subsample': [0.3, 0.6, 0.9]}
for n, subsample in enumerate(param_grid['subsample']):
   plt.subplot(1,len(param_grid['subsample']), n+1)
   df = cv_results[cv_results['param_subsample']==subsample]
   plt.plot(df["param_learning_rate"], df["mean_test_score"])
   plt.plot(df["param_learning_rate"], df["mean_train_score"])
   plt.xlabel('learning_rate')
   plt.ylabel('AUC')
   plt.title("subsample={0}".format(subsample))
   plt.ylim([0.60, 1])
   plt.legend(['test score', 'train score'], loc='upper left')
   plt.xscale('log')
```

**#Naive Bayes**

```
 from sklearn.naive_bayes import GaussianNB
GNb = GaussianNB()
GNb.fit(X_train, y_train)
predictions=GNb.predict_proba(X_test)
predictions[:10]
GB_acc = metrics.roc_auc_score(y_test,predictions[:,1])
GB_acc
plt.figure(figsize=(16,6))
```

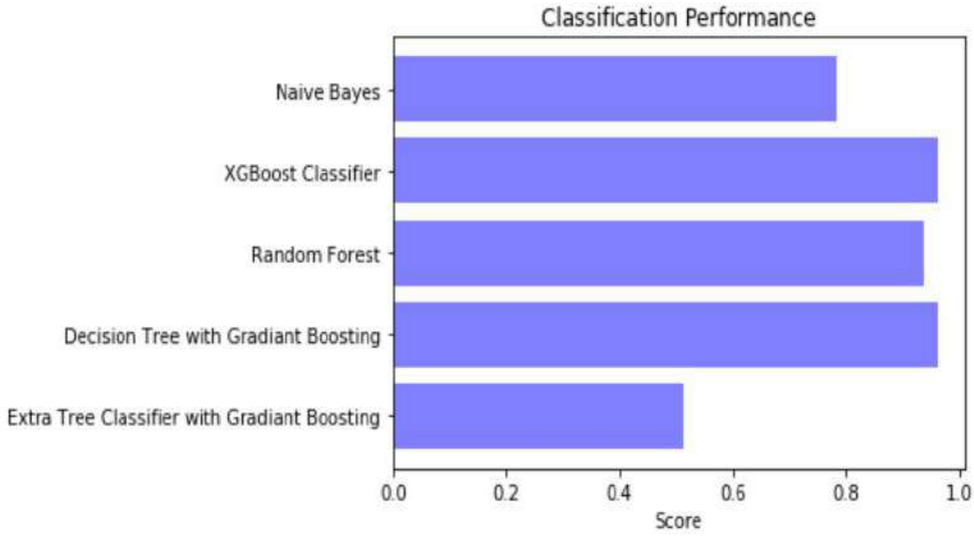
```
for n, subsample in enumerate(param_grid['subsample']):
    # subplot 1/n
    plt.subplot(1,len(param_grid['subsample']), n+1)
    df = cv_results[cv_results['param_subsample']==subsample]
    plt.plot(df["param_learning_rate"], df["mean_test_score"])
    plt.plot(df["param_learning_rate"], df["mean_train_score"])
    plt.xlabel('learning_rate')
    plt.ylabel('AUC')
    plt.title("subsample={0}".format(subsample))
    plt.ylim([0.60, 1])
    plt.legend(['test score', 'train score'], loc='upper left')
    plt.xscale('log')
```

**#Accuracy Comparsion**

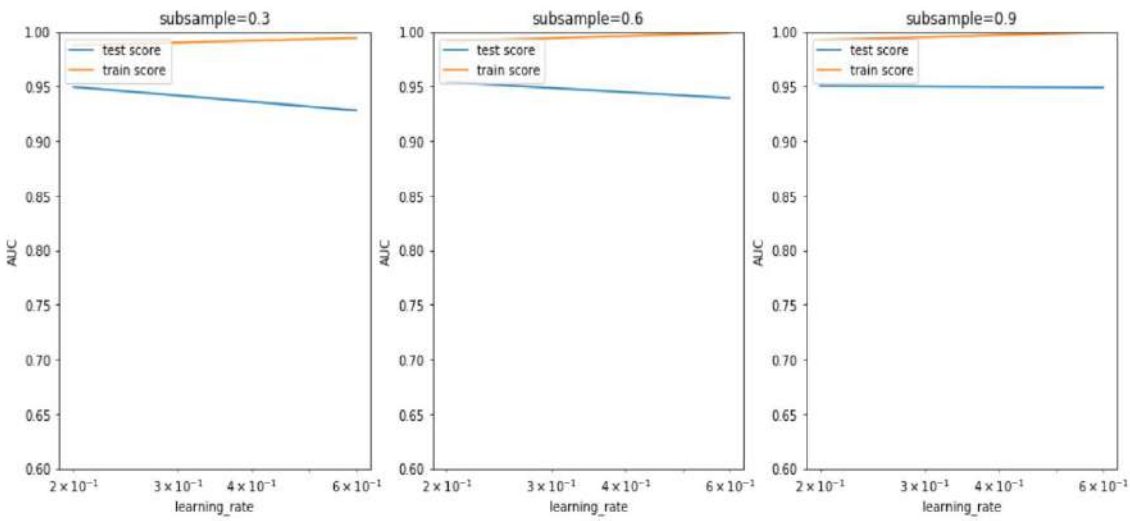
```
score = [Extra_acc,DT_acc,RF_acc,xgb_acc,GB_acc]
#make variabel for save the result and to show it
classifier = ('Extra Tree Classifier with Gradiant Boosting','Decision Tree with Gradiant Boosting','Random Forest','XGBoost Classifier','Naive Bayes')
y_pos = np.arange(len(classifier))
print(y_pos)
print(score)
import matplotlib.pyplot as plt2
plt2.barh(y_pos, score, align='center', alpha=0.5,color='blue')
plt2.yticks(y_pos, classifier)
plt2.xlabel('Score')
plt2.title('Classification Performance')
plt2.show()
import joblib
filename = 'model.sav'
joblib.dump(model, filename)
```

# 5. SCREENSHOTS

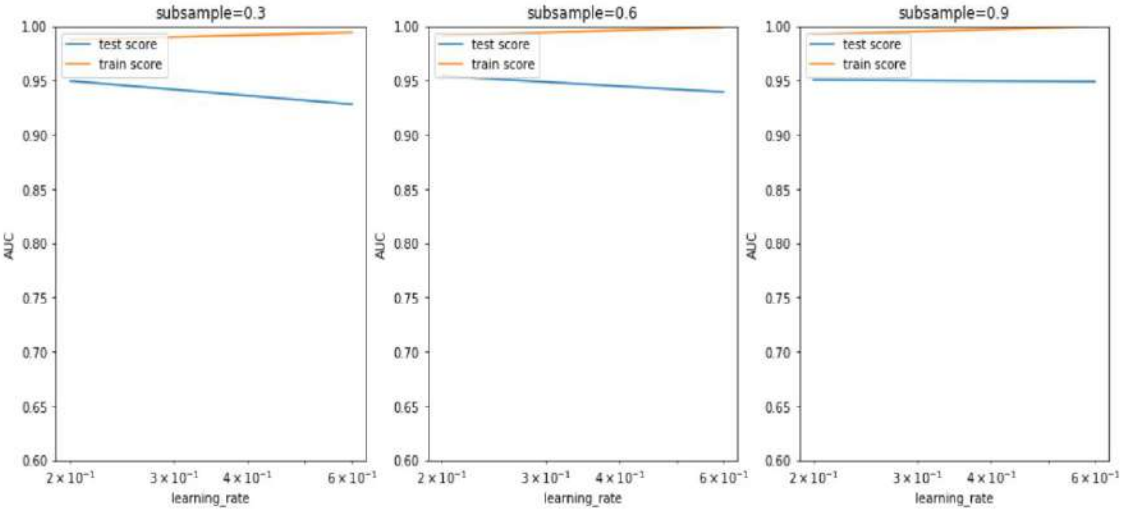
# 5. SCREENSHOTS



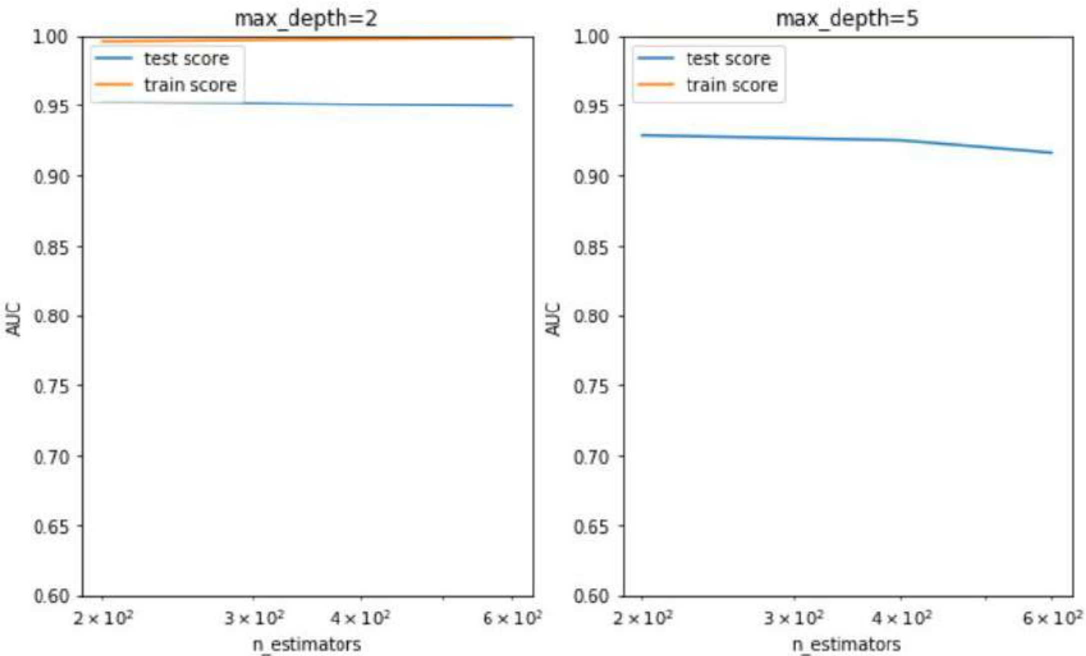

screenshot 5.1 Bar graph showing classification performance of algorithms




Screenshot 5.2 Graph for Naive Bayes




Screenshot 5.3 Graph for Random Forest




Screenshot 5.4 Graph for Decision tree with Adaboost classifier

# 6. TESTING

# 6. TESTING

## 6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

## 6.2 TYPES OF TESTING

### 6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### 6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests 29 demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

### 6.2.3 VALIDATION TESTING

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

### 6.2.4 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input: identified classes of valid input must be accepted.

Invalid Input: identified classes of invalid input must be rejected.

Functions: identified functions must be exercised.

Output: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

# 7. CONCLUSION

# 7. CONCLUSION

## 7.1 PROJECT CONCLUSION

The financing of millions of websites and mobile apps on-line ads is a template. Digital advertising with special purpose attack methods, called click malware, is constantly targeted by criminals. An important security challenge is click fraud created via malware. The state-of - the-art techniques can easily detect static attacks involving large attack volumes. Nonetheless, current methods fail to detect complex attacks involving steady click-spam that match the app user's actions. Timing analysis has been found to have a crucial role to play in isolating click scams, both static and dynamic. This research paper applies a technique that detects click-spam using relative uncertainty between click-spam and valid clicks-streams. It does this by identifying repeated patterns from valid click-spam in the ad network. A malware corpus is also analysed in an instrumented environment which can handle click-spam generation by exposing malware to legitimate click-spams.

## 7.2 FUTURE SCOPE

Future enhancement future improvements to the process that can be made. The adaptive character of the system means that the learning data are continually improved. Nevertheless, there are additional ways of improving identification system accuracy. In this study, we covered a wide range of classification algorithms to classify who you are. characteristics, such as consumer geographical location, were not included in the existing classification process. To this end, training data would need to be developed for every campaign, so that a warning flag is lifted if most viewers for an ad suddenly comes from a new location. We think these ideas should be discussed further as they may be helpful input attributes to the classification system.

# 8. BIBLIOGRAPHY

# 8. BIBLIOGRAPHY

## 8.1 REFERENCES

[1] Mouawi, R., El Hajj, I.H.: Towards a machine learning approach for detecting click fraud mobile advertising. In: 13th International Conference on Innovations in Information Technology (2018)

[2] Blizard, T., Livic, N.: Click-fraud monetizing malware: a survey and case study. In: 2012 7th International Conference on Malicious and Unwanted Software (MALWARE), pp. 67–72, Puerto Rico, USA, IEEE

[3] Reference Wang R., Fu B., Fu G., Wang M. Deep & cross network for ad click predictions Proceedings

[4] Thejas G.S., Boroojeni K.G., Chandna K., Bhatia I., Iyengar S.S., Sunitha N.R. Deep learning-based model to fight against ad click fraud Proceedings of the 2019 ACM southeast conference, ACM SE '19, Association for Computing Machinery, New York, NY, USA (2019), pp. 176-181

## 8.2 WEBSITES

[1]. https://fruition.net/about/blog/types-click-fraud-detect/

[2]. https://www.clickcease.com/blog/click-fraud-statistics/

[3].https://www.semanticscholar.org/paper/Prediction-of-click-frauds-in-mobile-advertising-Taneja-Garg/de4eaff56fa60e83f238d81a52e9dc5f3fd33f18 .

## 8.3 GITHUB LINK

https://github.com/TejaNiranjan/Machine-learning-approach-for-click-fraud-detection

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

# MACHINE LEARNING APPROACH FOR CLICK FRAUD DETECTION

Dr. G. Madhukar, Associate Professor, Dept of CSE, CMR Technical Campus

K Teja Niranjan, Dept of CSE, CMR Technical Campus

N Madhuri, Dept of CSE, CMR Technical Campus

A Vamshidhar Reddy, Dept of CSE, CMR Technical Campus

---

**ABSTRACT -** Mobile advertising has gained popularity in recent years as a means for publishers to monetize their free applications due to the increase of Internet usage. Click fraud is one of the main concerns in the in-app advertising industry. Click fraud involves online advertisements that have been clicked on. Pay-per-click fraud involves online advertisements that have been clicked on. Advertisements that pay per click typically target potential customers by charging a fee per click. With machine learning as a solution, we designed the system to detect click fraud using naive bayes, xgboost classifier, random forest, decision tree with gradient boosting, extra tree classifier with gradient boosting, and we observed decision tree with gradient boosting outperformed other algorithms with 96.07% accuracy.

**Key Words:** click fraud detection, online advertisement, Real time fraud detection

## 1. INTRODUCTION

Fraudulent clicks on pay-per-click ads are designed to divert the budgets of advertisers. There are several parties who are engaging in click fraud. Consider the top three criminals, competitors, webmasters, and fraud circles to understand who is clicking on your ad fraudulently. They serve ads to users and agree on a price per action. According to the frequency of visitors to the advertiser, the ad network pays the content publisher. With this payment model, however, there are security risks, such as click fraud. The number of fraudulent clicks for smartphones doubled in four months (ppc, 2019) from 1 in 5 in 2017.

There is a significant portion of web traffic that is fraudulent based on these click fraud statistics. Click fraud always results in financial losses for the advertiser, regardless of its form. Most ad click fraud is committed by competitors. Make yourself more competitive by wasting your competitor's click billing budget. When webmasters commit click fraud, they display ads on their sites to generate fraudulent revenue. To increase sales, they choose to click on these ads instead of creating and developing their website. Click farms are a way to trick people into clicking on ads all day long to make money on click fraud. Compared to automated scripts, we find it more beneficial to use real people, as compelling click performers can lead to clicks on your advertisement.

## 2. METHODOLOGY

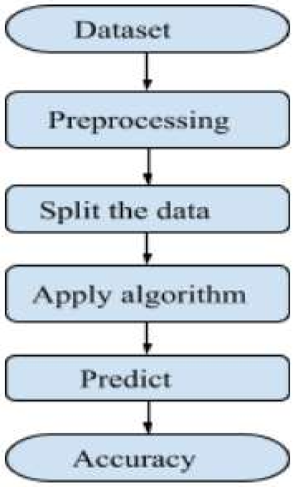



Figure- 2.1 Data Flow Diagram

The diagram basically describes a program control overflow. The first step in your project is to fetch the dataset and remove all kinds of errors, missing values and noisy data. This is sometimes referred to as data pre-processing. After the data has been processed, it will try to split the data. Training and test datasets that try to apply the decision tree algorithm individually. After applying these algorithms, you will get two types of results for both the test and training datasets, and these results will be compared in the next step. These steps of applying the algorithm to get the values continue until you have the accuracy you need for your project.

**Dataset**

Talking Data, China's largest independent big data service platform, covers more than 70% of active mobile devices nationwide. It processes 3 billion clicks per day, 90% of which are potentially fraudulent. The current approach to prevent click fraud by app developers is to measure user click behaviour across the portfolio and flag IP addresses that generate a lot of clicks but don't install the app. I used this information to create an IP blacklist and a device blacklist.

The dataset contains 100001 records, column are 8 and label is 0

Attribute information:

1)Ip

2)App

3)Device

4)Os

5)channel

6) click time

7) is attribute

## 3. MODELING AND ANALYSIS

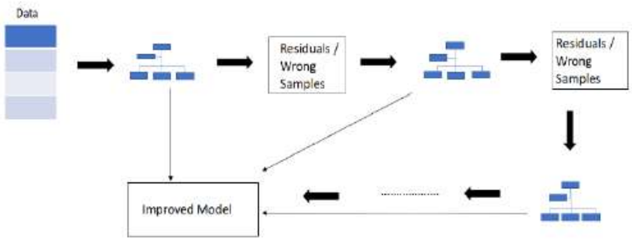



**FIGURE 3.1:** Architecture Diagram

The project architecture represents the full functionality of the click fraud detection project program. First, we collect data from various sources such as websites and Kaggle. Then remove the noisy data and try to pre-process the data. After the pre-processing is complete, it tries to apply the decision tree algorithm to the dataset. Therefore, after application, you will get two results. For example, if you get the correct results, try applying a decision tree algorithm to this data. They are added to the improved result collection, incorrect samples are reprocessed, and the process continues until reasonable accuracy is found.

## 4. RESULTS AND DISCUSSION

**Evaluation metrics**

True Positive: That is when we anticipate Jesus and the actual result is also Yes.

True Negative: In this case, we are predicting "no" and the actual output is also "no".

False positives: If you predicted "yes", it was actually "no".

False Negatives: If I expected it to be no, it wasn't.

accuracy=TP+TN/TP+FP+TN+FN

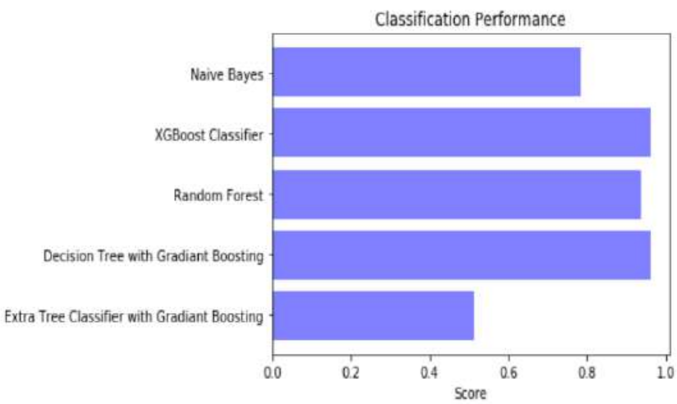



Figure 4.1: Bar Graph

We have trained 5 machine learning algorithms and the above bar graph accuracy comparison is given below

| sno | Algorithm names | Accuracy |
|---|---|---|
| 1 | Naive Bayes | 78.21 % |
| 2 | XGBoost Classifier | 96.06 % |
| 3 | Random forest | 93.62 % |
| 4 | Decision tree with gradient boosting | 96.07 % |
| 5 | Extra tree classifier with Gradient boosting | 51.10 % |

**Table 4.2:** Accuracy Comparison of Algorithms

We have observed that decision tree algorithm has performed better than other algorithms so we finalized decision tree.

## 5. CONCLUSION

We have developed a click fraud detection mechanism that can be used in the real world. You used a dataset with different attributes. We have used many click fraud detection algorithms such as Naive Bayes, xgboost classifier, decision tree gradient boosting, additional tree classifier with gradient boosting, and random forest. Of all these algorithms, xgBoost works very well with a project accuracy of 0.9606. This machine learning template can be used to identify real and fake users.

## 6. FUTURE SCOPE

If many resources are available, you can increase the number of decision trees to get accurate results. You can also apply multi-grain scans to improve data preprocessing. You can also add the consumer's geographic location as an attribute to analyze and customize the results. Also, if you use this geographic location to see if a person or bot is trying to click from the new location, you'll see a warning flag telling you that the new user is clicking. I think these ideas need further discussion as they are input attributes that are useful for classification systems and projects.

## 7. ACKNOWLEDGEMENT

The success of this paper includes help from our guide as well. We are grateful to our guide, Dr. G. Madhukar, Associate professor, CMR Technical Campus, for his expertise that assisted us in our research.

## 8. REFERENCES

[1]. Antoniou, D., Paschou, M., Sakkopoulos, E., Sourla, E., Tzimas, G., Tsakalidis, A., & Viennas, E. (2011). Exposing click-fraud using a burst detection algorithm. In 2011 IEEE

[2]. symposium on computers and communications (ISCC) (pp. 1111–1116). IEEE.

[3]. Arisoy, E., Sainath, T. N., Kingsbury, B., & Ramabhadran, B. (2012). Deep neural network language models. In Proceedings of the NAACL-HLT 2012 workshop: will we ever really replace the N-gram model? on the future of language modeling for HLT (pp. 20–28). Association for Computational Linguistics.

[4]. Breiman, L. (1996). Stacked regressions. Machine Learning, 24(1), 49–64. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

[5]. Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd Acm Sigkdd international conference on knowledge discovery and data mining (pp. 785–794). ACM.

[6]. Chenoweth, T., ObradoviĆ, Z., & Lee, S. S. (2017). Embedding technical analysis into neural network-based trading systems. In Artificial intelligence applications on wall street (pp. 523–541). Routledge.

[7]. Click fraud statistics: The click fraud blog. (2019). Click Cease. URL: https://www. lickcease.com/blog/click-fraud-statistics/.

[8]. Click fraud - The 5 most common forms of click fraud. (2019). URL: https://fruition. net/about/blog/types-click-fraud-detect/.

ISSN: 2582-3930

IJSREM e-Journal

International Journal of Scientific Research in Engineering and Management

*is hereby awarding this certificate to*

# *K Teja Niranjan*

*in recognition the publication of manuscript entitled*

MACHINE LEARNING APPROACH FOR CLICK FRAUD DETECTION

*published in Ijsrem Journal Volume 06 Issue 06 June 2022*

www.ijsrem.com

Editor in Chief
E-mail: editor@ijsrem.com

ISSN: 2582-3930

International Journal of Scientific Research in Engineering and Management

*is hereby awarding this certificate to*

# *N Madhuri*

*in recognition the publication of manuscript entitled*

MACHINE LEARNING APPROACH FOR CLICK FRAUD DETECTION

*published in Ijsrem Journal Volume 06 Issue 06 June 2022*

Editor in Chief

www.ijsrem.com

E-mail: editor@ijsrem.com

ISSN: 2582-3930

IJSREM e-Journal

IJSREM

# International Journal of Scientific Research in Engineering and Management

*is hereby awarding this certificate to*

# *A Vamshidhar Reddy*

*in recognition the publication of manuscript entitled*

## MACHINE LEARNING APPROACH FOR CLICK FRAUD DETECTION

*published in Ijsrem Journal Volume 06 Issue 06 June 2022*

www.ijsrem.com

Editor in Chief
E-mail: editor@ijsrem.com