A

Major Project

On

PREDICTING MOVIE PROFITABILITY USING DECISION TREES

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

BY

Gaddam Mithila Reddy	(177R1A05D6)
Dugyala Nimisha	(177R1A05D3)
Polagouni Swetha	(177R1A05G1)
Vasala Sri Kavya	(177R1A05H3)

Under the Guidance of

DR. M. VARAPRASAD RAO

(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 UGC Act.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2017-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

This is to certify that the project entitled "**PREDICTING MOVIE PROFITABILITY USING DECISION TREES**", being submitted by **GADDAM MITHILA REDDY (177R1A053D6)**, **DUGYALA NIMISHA (177R1A05D3)**, **POLAGOUNI SWETHA (177R1A05G1)** & **VASALA SRI KAVYA (177R1A05H3)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafied work carried out under our guidance and supervision during the year 2020-2021. It is certified that they have completed the project satisfactorily.

Dr. M. Varaprasad Rao
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voce 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. M. Varaprasad Rao, Professor for his 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) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane** and **Mr. B. Ramji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to the Head of the Department **Dr. K. Srujan Raju** for providing excellent infrastructure and a nice atmosphere for completing this project successfully.

We are obliged to our Director **Dr. A. Raji Reddy** for being cooperative throughout the course of this project. We would like to express our sincere gratitude to our Chairman Sri. **Ch. Gopal Reddy** for his encouragement throughout the course of this project

The guidance and support received from all the members of **CMR TECHNICAL CAMPUS** who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

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

GADDAM MITHILA REDDY (177R1A05D6)

DUGYALA NIMISHA (177R1A05D3)

POLAGOUNI SWETHA (177R1A05G1)

VASALA SRI KAVYA (177R1A05H3)

ABSTRACT

The movie industry is one of the most important branches of the entertainment industry, which generates a lot of revenue. The person playing a big role in this aspect is the producer as they are in charge of funding needed to produce the movie. However, producing a movie has its risks; one being that there is a chance of the movie not covering production costs. A producer relies on tools to predict profitability in movies for decision making with regards to whether or not to produce a movie project. For several years now, researchers have used different approaches to collect information that would be used as variables when predicting the success of a movie, but very few have explored using attributes directly related to a movie.

This paper focuses on using decision trees to characterize and predict movie profitability. Decision tree classifiers are relatively fast compared to other classification methods and are easily interpreted by humans. For our project, we want to see the difference between using Gini Index and Entropy for the selection of the best split point based on an attribute using an impurity function. The decision tree will be used to forecast the profitability of a movie before its production. Decision trees are models commonly used as decision support tools and its results show that the resulting model predicts whether or not a movie will be profitable with an average accuracy of 63.79%. Keeping in mind that the approach presented in this paper is not a standalone tool, it should, however, be able to round out forecasting methods such as the producer’s foresight and judgment.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This Project is titled as “Predicting Movie Profitability Using Decision Trees”. This project focuses on using decision trees to characterize and predict movie profitability. Decision tree classifiers are relatively fast compared to other classification methods and are easily interpreted by humans. For our project, we want to see the difference between using Gini Index and Entropy for the selection of the best split point based on an attribute using an impurity function. The decision tree will be used to forecast the profitability of a movie before its production.

1.2 PROJECT PURPOSE

This project has been developedto predict movie’s profitability before production in order to help investors and producers to make a more informed decision where to invest in a movie or the effect of the budget on the retains from revenue.The decision tree will be used to forecast the profitability of a movie before its production.

1.3 PROJECT FEATURES

Classification is the task of assigning objects to one of several predefined categories. In classification, there is a given set of sample records called the training dataset with each record containing attributes. An attribute can be numerical or categorical. One of the categorical attributes is called the classification attribute and its values are called class labels. The class labels indicate the class to which a record belongs. For this project, the expected revenue will be divided into six class labels whereby a movie classified in the category 5 is the highest profitable movie and the one in 0 has no profit.

2. SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

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 dwells 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 isto be done.

2.1 PROBLEM DEFINITION

The movie industry is one of the most important branches of the entertainment industry, which generates a lot of revenue. The person playing a big role in this aspect is the producer as they are in charge of funding needed to produce the movie. However, producing a movie has its risks; one being that there is a chance of the movie not covering production costs. A producer relies on tools to predict profitability in movies for decision making with regards to whether or not to produce a movie project.This project focuses on using decision trees to characterize and predict movie profitability. Decision tree classifiers are relatively fast compared to other classification methods and are easily interpreted by humans. For our project, we want to see the difference between using Gini Index and Entropy for the selection of the best split point based on an attribute using an impurity function. The decision tree will be used to forecast the profitability of a movie before its production.

2.2 EXISTING SYSTEM

Large quantities of data regarding movies are generated and stored for analytical reasons and this shows the agency in the movie industry. The way in which success is defined is of paramount importance to the problem, but past works have focused primarily on gross box office revenue while some used the number of admissions. There are several related works involving the prediction of movie success based on reviews and box office. The basic assumption for using the two as success metrics is simple, a movie that sells well at the box office is considered a success. However, the two metrics ignore how much it costs to produce a movie.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Movie profitability is predicted just by considering the gross amount collected and the total investment.
- The prediction is done only using attributes that are obtained after release of movie.

2.3 PROPOSED SYSTEM

Classification is the task of assigning objects to one of several predefined categories. In classification, there is a given set of sample records called the training dataset with each record containing attributes. An attribute can be numerical or categorical. One of the categorical attributes is called the classification attribute and its values are called class labels. The class labels indicate the class to which a record belongs. For this project, the expected revenue will be divided into six class labels whereby a movie classified in the category 5 is the highest profitable movie and the one in 0 has no profit.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The project aims to predict movie's profitability before production in order to help investors and producer make a more informed decision where to invest in a movie or the effect of the budget on the retains from revenue.

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 to be carried out. 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

• Social 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, isthe 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 benefits 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 give 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 & 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. Thefollowing are some hardware requirements.

- Processor : i3 (or) Higher
- RAM : 4GB (or) Higher
- HDD : 500 GB

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 : Microsoft Windows, Linux or Mac.
- Python – PyCharm

3. ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCITECTURE

This project architecture shows the procedure followed for breed detectionusing machine learning, starting from input to final prediction.

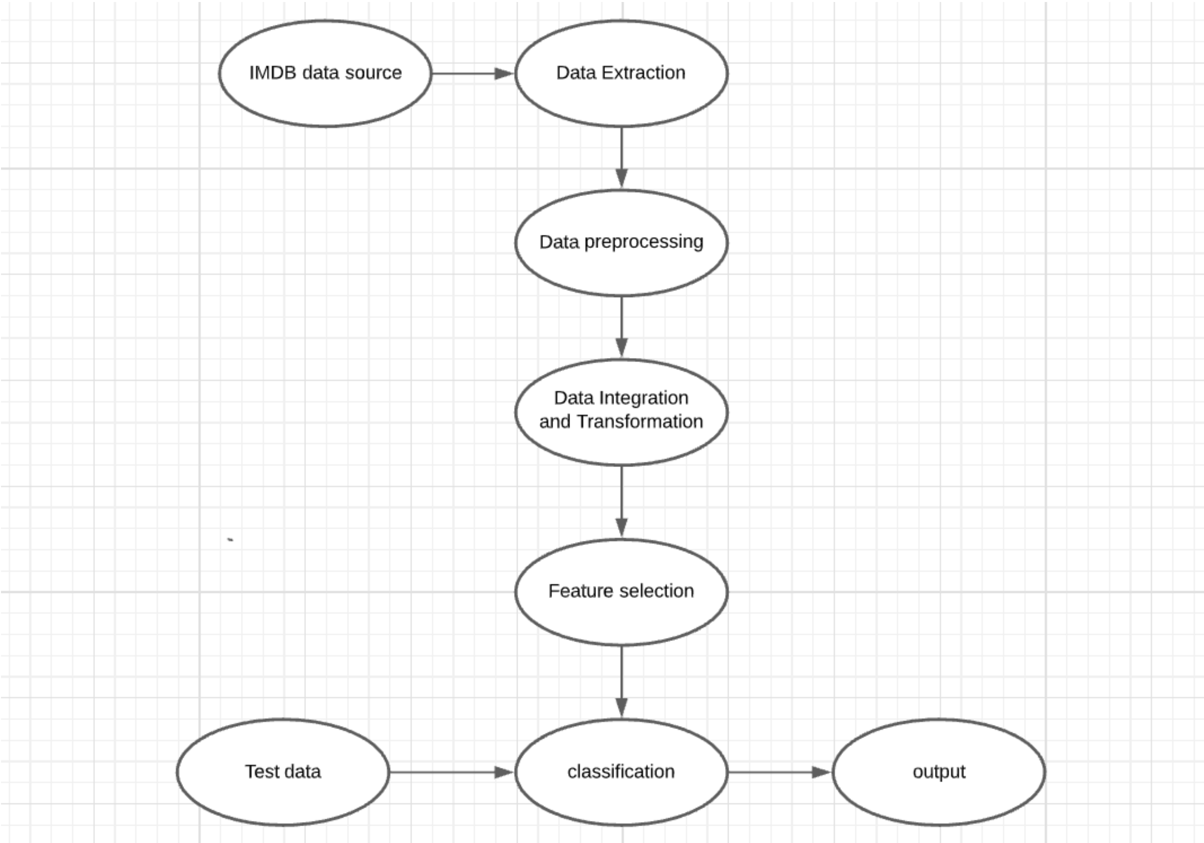
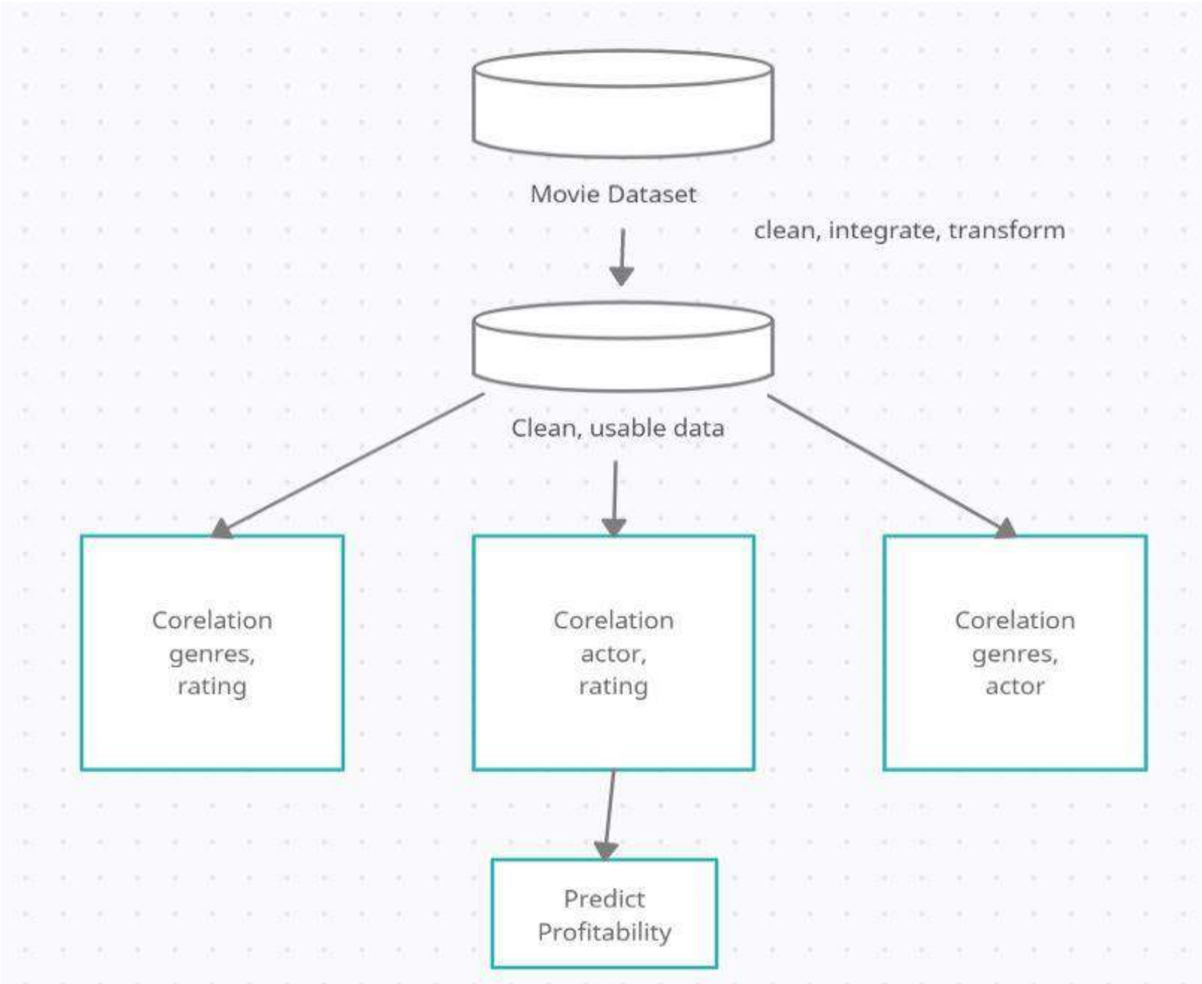


Figure 3.1: Architecture to predict movie profitability.



3.2 DESCRIPTION

Input Data: Input data is generally given in .csv format where the data is fetched and mapped in the data framed from the source columns.

Reading Data: library files are used to read the data into the data frame.

Separating Features: In this following step we are going to separate the features which we take to train the model by giving the target value i.e. 1/0 for the particular of features.

Normalization: Normalization is a very important step while we are dealing with the large values in the features as the higher bit integers will cost high computational power and time. To achieve the efficiency in computation we are going to normalize the data values.

Training and test data: Training data is passed to the Decision tree classifier to train the model. Test data is used to test the trained model whether it is making correct predictions or not.

Decision Tree Classifier: the purpose of choosing the decision tree classifier for this project the efficiency and accuracy that we have observed when compared to other classifiers.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically two actors who are the user and the administrator. The user has the rights to login, access to resources and to view the details. Whereas the administrator has the login, access to resources of the users and also the right to update and remove the crime details, and he can also view the user files.

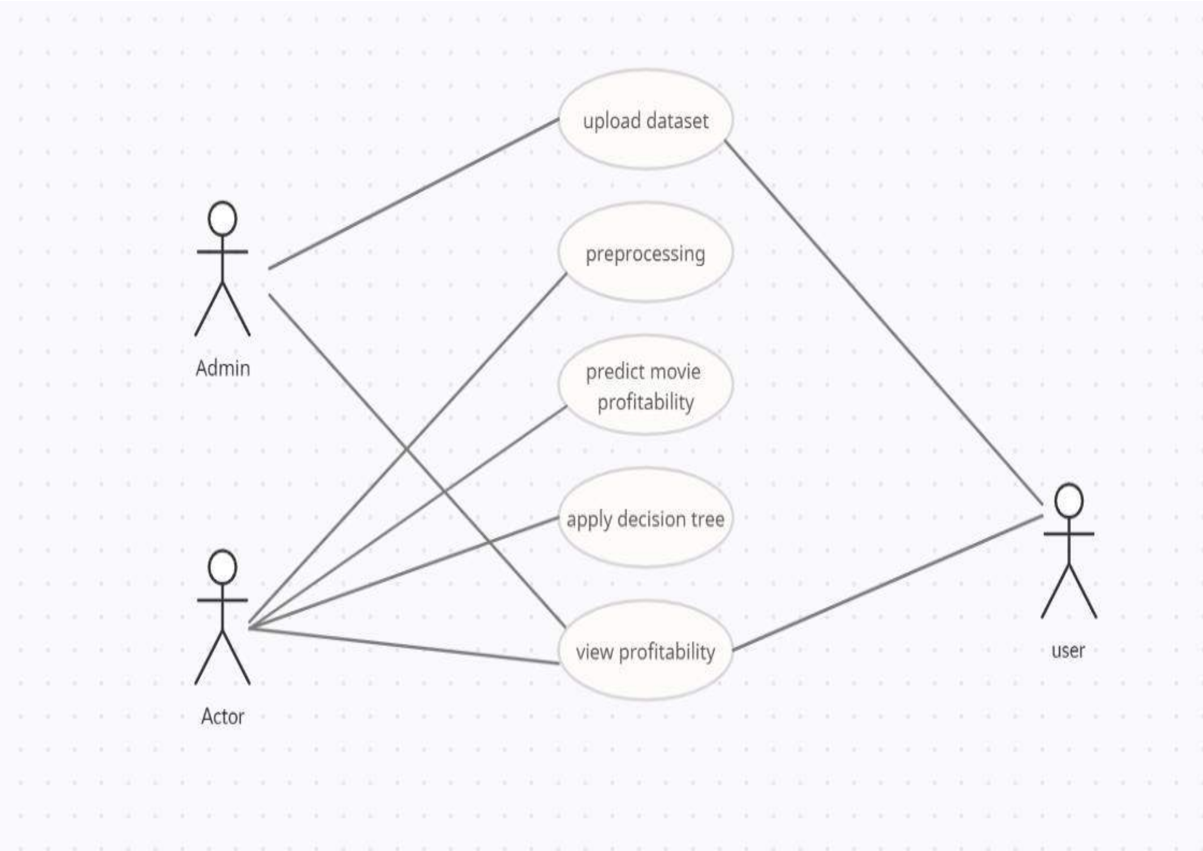


Figure 3.2: Use case diagram to predict movie profitability

3.4 CLASS DIAGRAM

Class Diagram is a collection of classes and objects.

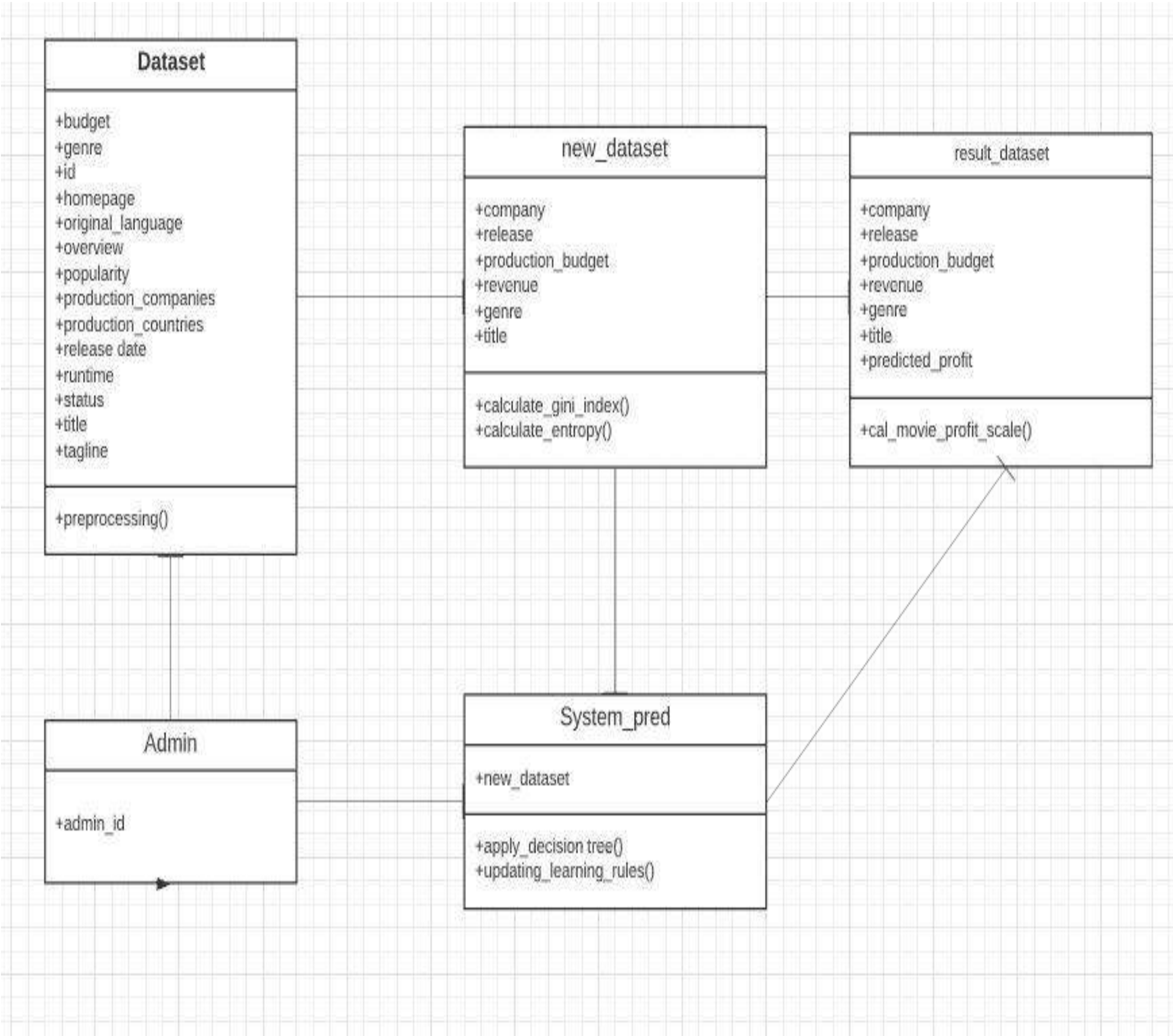


Figure 3.3: Class diagram to predict movie profitability

3.5 SEQUENCE DIAGRAM

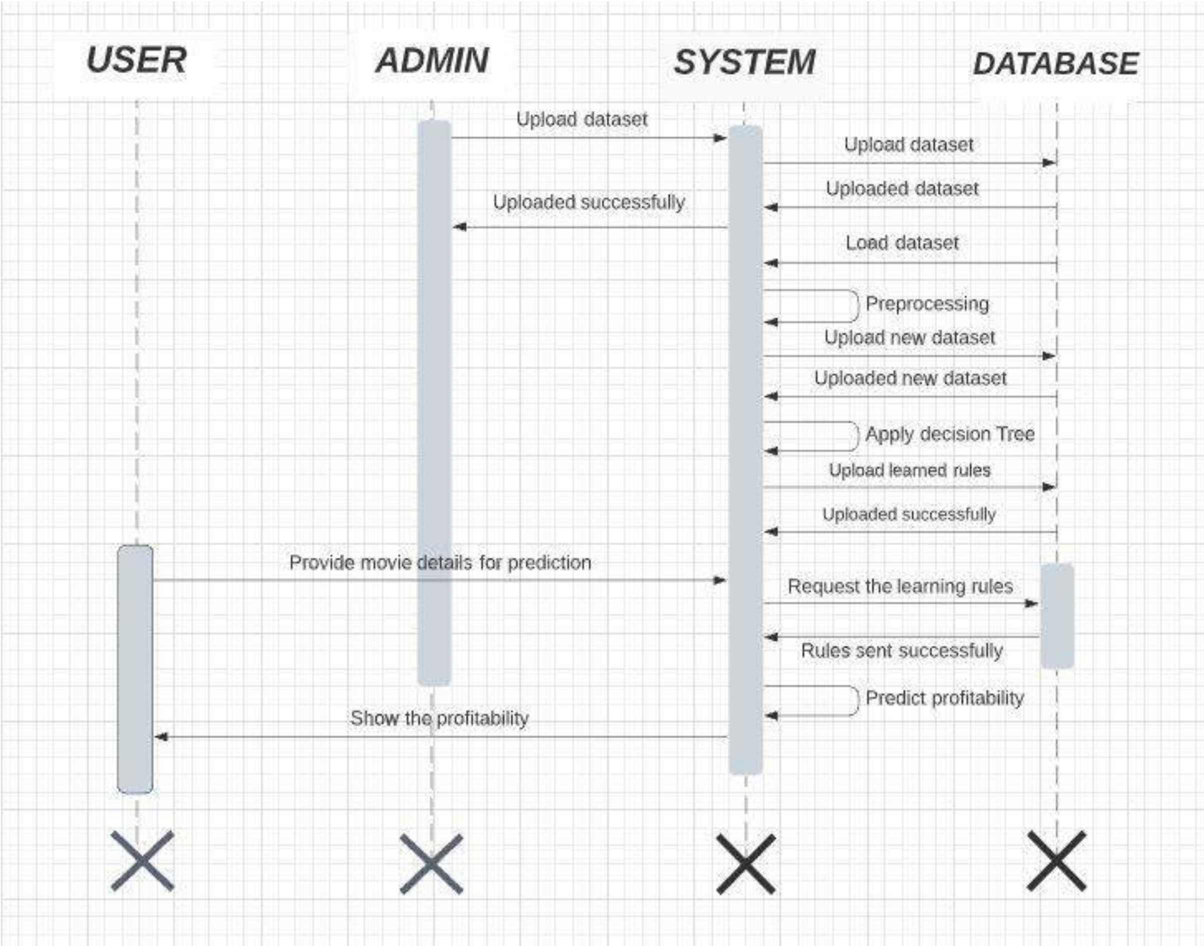


Figure 3.4 : Sequence diagram to predict movie profitability

3.6 ACTIVITY DIAGRAM

It describes about flow of activity states.

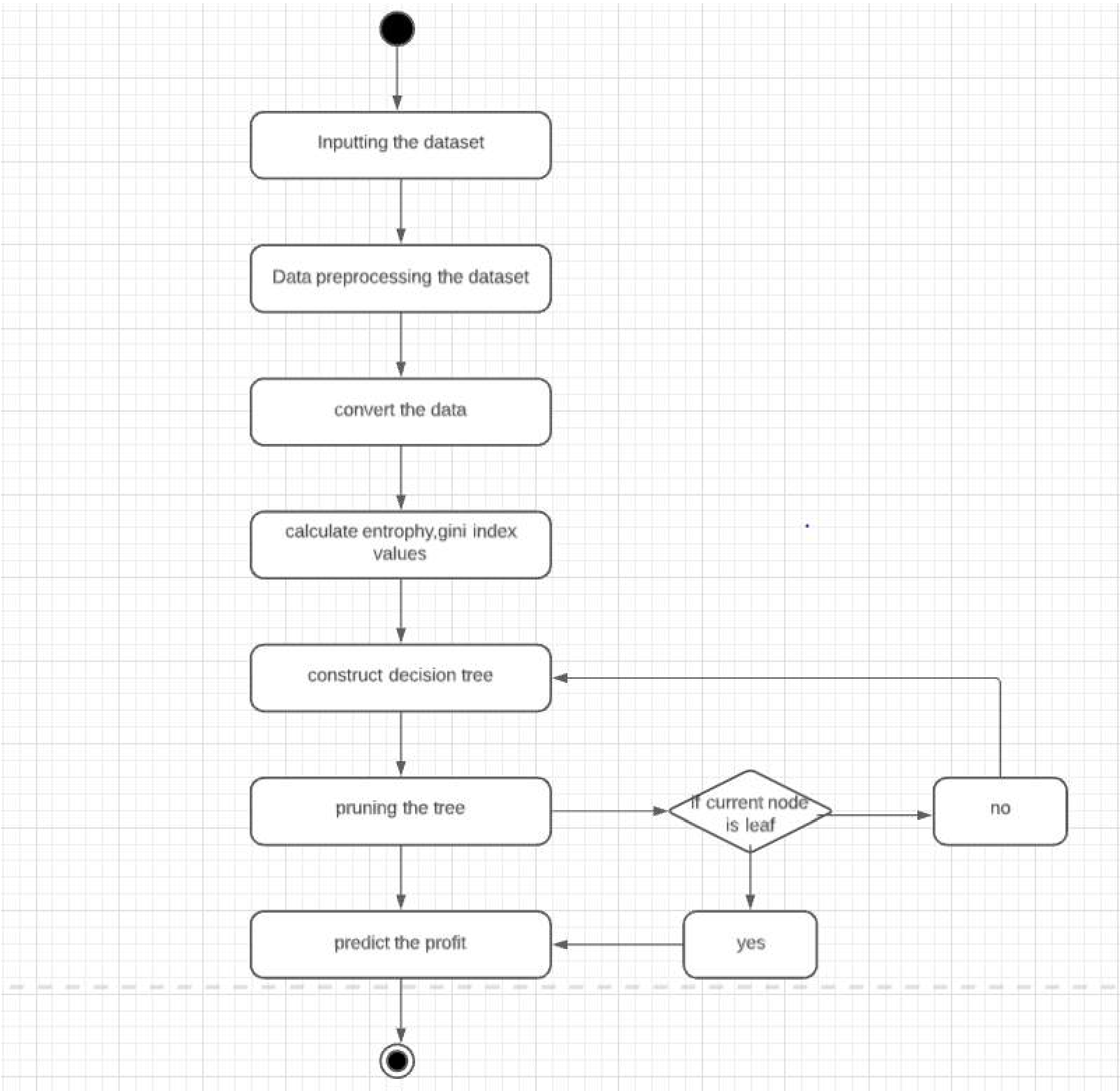


Figure 3.5: Activity diagram to predict movie profitability

4. IMPLEMENTATION

4. IMPLEMENTATION

4.1 SAMPLE CODE

main.py

```
from logic import id3

from logic import cart

import data_cleanup

print("Parsing data and correcting for inflation...")

data_cleanup.clean_data()

print("\nBuilding decision tree using ID3 algorithm...\n")

id3.run_id3('data/new_database2.csv', 50)

cart.run_cart('data/new_database2.csv', 50)
```

id3.py

```
import os

import sys

import math

from operator import itemgetter

from itertools import groupby

from collections import Counter

sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))

from classes.dataset import Dataset

from classes.node import Node

from classes.tree import Tree

TARGET = 'revenue'

def init_dataset(database_name):

newdata = Dataset()

newdata.get_data(database_name)

    return newdata

def id3_tree(learn_set, attribute_set):
```

```
#Create a node, label and attach to tree later
curr_node = Node('Empty')
    attributes = attribute_set.copy()
attribute_set_length = len(attributes)
revenue_class = learn_set[0][TARGET]
revenue_all_same = True
list_revenues = []
    for movie in learn_set:
curr_revenue = movie[TARGET]
list_revenues.append(curr_revenue)
        if revenue_class != curr_revenue:
revenue_all_same = False
    if revenue_all_same:
curr_node.update_node_label(revenue_class)
elifattribute_set_length == 0:
revenue_counter = Counter(list_revenues)
revenue_majority = revenue_counter.most_common(1)
curr_node.update_node_label(revenue_majority[0][0])
    else:
attribute_name = find_information_gain(learn_set, attributes)
attributes.discard(attribute_name)
        groups = []
        keys = []
learn_set = sorted(learn_set, key=itemgetter(attribute_name))
        for group in groupby(learn_set, itemgetter(attribute_name)):
keys.append(group[0])
groups.append(list(group[1]))
curr_node.update_node_label(attribute_name)
        for i in range(0, len(keys)):
partition_size = len(groups[i])
```

```
            if partition_size == 0:
majority_counter = Counter(list_revenues)
common_rev = majority_counter.most_common(1)
curr_node.update_node_label(common_rev[0][0])
            else:
curr_node.new_branch(keys[i])
curr_node.new_child(id3_tree(groups[i], attributes)
    return curr_node
def find_information_gain(learn_set, attribute_set):
info_gain = 0
best_attribute = ''
info_of_class = calculate_entropy(learn_set, TARGET)
 for attribute in attribute_set:
info_of_attribute = calculate_info(learn_set, attribute, TARGET)
new_info_gain = info_of_class - info_of_attribute
        if new_info_gain>= info_gain:
info_gain = new_info_gain
best_attribute = attribute
    return best_attribute
def calculate_entropy(learn_set, target_attribute):
total_size = len(learn_set)
list_target = []
    for movie in learn_set:
list_target.append(movie[target_attribute])
    counter = Counter(list_target)
    common = counter.most_common()
    entropy = 0
    for label in common:
count_label = label[1]
        portion = count_label/total_size
```

```
        entropy -= ((portion)*math.log2(portion))
return entropy
def calculate_info(learn_set, attribute, target_attribute):
    info = 0
    keys = []
    groups = []
total_size = len(learn_set)
learn_set = sorted(learn_set, key=itemgetter(attribute))
    for group in groupby(learn_set, itemgetter(attribute)):
keys.append(group[0])
groups.append(list(group[1]))
    for g in groups:
        count = len(g)
        portion = count/total_size
        info += portion * calculate_entropy(g, target_attribute)
    return info
def run_id3(database_name, num_trials):
    total = 0
    for x in range(0, num_trials):
mydata = init_dataset(database_name)
decision_tree = Tree(id3_tree(mydata.learn_set, mydata.attribute_set))
decision_tree.insert_rules()
num_correct = 0
        for movie in mydata.test_set:
            for rule in decision_tree.rules:
conditions_met = True
                key = ''
                value = ''
                for i in range(0, len(rule) - 1, 2):
                    key = rule[i]
```

```
                    value = rule[i + 1]
                    if movie[key] != value:
conditions_met = False
                        break;
                if conditions_met:
                    if rule[-1] == movie[TARGET]:
num_correct += 1
                        break;
        total += (num_correct/len(mydata.test_set)) * 100
print("Test #" + str(x) + ", accuracy = " +
              str((num_correct/len(mydata.test_set)*100)))
    total /= num_trials
print("Average over " + str(num_trials) + " trials: " + str(total) + "%")
```

cart.py

```
import csv
from random import shuffle
class _SplittingCriterion:
    def __init__(self, attr_col_num, value):
self.attr_col_num = attr_col_num
self.value = value
    def match(self, row):
        return self.value == row[self.attr_col_num]
    def __str__(self):
        condition = '=='
        return "{0} {1} {2}".format(header[self.attr_col_num], condition, self.value)
class _Leaf:
    def __init__(self, rows):
self.predictions = _count_class_values(rows)
class _SplittingNode:
```

```
    def __init__(self, split_crit, true_branch, false_branch):
self.split_crit = split_crit
self.true_branch = true_branch
self.false_branch = false_branch
def _get_unique_values(rows, attr_num):
    return set([row[attr_num] for row in rows])
def _count_class_values(rows):
    counter = {}  # save it as label -> count
    for row in rows:
class_label = row[revenue_pos]  # class label at the last column
        if class_label not in counter:
            counter[class_label] = 0
        counter[class_label] += 1
    return counter
def _partition(rows, split_crit):
true_rows, false_rows = [], []  # to hold the partitions
    for row in rows:
        if split_crit.match(row):
true_rows.append(row)
        else:
false_rows.append(row)
    return true_rows, false_rows
def _gini(rows):
class_values = _count_class_values(rows)
gini_impurity = 1
    for class_val in class_values:
        prob = class_values[class_val] / float(len(rows))
gini_impurity -= prob ** 2  # Gini = 1 - (sum of prob^2)
    return gini_impurity
def _info_gain(left, right, current_uncertainty):
```

```
    prob = float(len(left)) / (len(left) + len(right))
    return current_uncertainty - prob * _gini(left) - (1 - prob) * _gini(right)
def _get_best_split(rows):
best_gain = 0  # to hold the best gain to split
best_split_crit = None  # to hold the best splitting criterion
current_uncertainty = _gini(rows)
    size = len(header)  # number of columns
    for col_num in range(size):
        if header[col_num] == 'revenue' or header[col_num] == 'title':
            continue
        else:
unique_values = _get_unique_values(rows, col_num)
        for val in unique_values:
split_crit = _SplittingCriterion(col_num, val)
true_rows, false_rows = _partition(rows, split_crit)
            if len(true_rows) == 0 or len(false_rows) == 0:
                continue
            gain = _info_gain(true_rows, false_rows, current_uncertainty)
            if gain >best_gain:
best_gain, best_split_crit = gain, split_crit
    return best_gain, best_split_crit
def _build_tree(rows):
    gain, split_crit = _get_best_split(rows)
    if gain == 0:
        return _Leaf(rows)
true_rows, false_rows = _partition(rows, split_crit)
true_branch = _build_tree(true_rows)
false_branch = _build_tree(false_rows)
    return _SplittingNode(split_crit, true_branch, false_branch)
def classify(row, node):
```

```
    if isinstance(node, _Leaf):
        return node.predictions
    if node.split_crit.match(row):
        return classify(row, node.true_branch)
    else:
        return classify(row, node.false_branch)
def predict(leaf):
    return max(leaf.keys(), key=(lambda key: leaf[key]))
def split_dataset(dataset, train_ratio):
    size = len(dataset)  # size of dataset
    shuffle(dataset)
train_data = dataset[:int(train_ratio * size)]
test_data = dataset[int(train_ratio * size):]
    return train_data, test_data
def _get_accuracy(tree, test):
    size = len(test)
    correct = 0.0
    with open('results.csv', 'w', newline='\n', encoding='utf-8') as resultFile:
        writer = csv.writer(resultFile, delimiter=',')
header_row = []
        for r in header:
header_row.append(r)
header_row.append('Prediction')
writer.writerow(header_row)
        for row in test:
result_row = []
            for r in row:
result_row.append(r)
predict_leaf = predict(classify(row, tree))  # Predict the data
result_row.append(predict_leaf)
```

```
writer.writerow(result_row)
if row[revenue_pos] == predict_leaf:
        correct += 1
  accuracy = correct / size * 100
  return accuracy
def _get_av_accuracy(dataset, train_ratio, n):
av_accuracy = 0
  for i in range(n):
    train, test = split_dataset(dataset, train_ratio)
    tree = _build_tree(train)
    accuracy = _get_accuracy(tree, test)
print('Test #{0}, accuracy = {1}'.format(i, accuracy))
av_accuracy += accuracy
av_accuracy /= n
print('Average over {0} trials: {1}%'.format(n, av_accuracy))
def run_cart(filename, n):
  with open(filename, 'r') as file:
    reader = csv.reader(file)
    dataset = list(reader)
  global header
  header = dataset[0]  # get column names
  global revenue_pos
revenue_pos = header.index('revenue')  # get position of class label
  dataset = dataset[1:]  # exclude column names from dataset
print('\nBuilding decision tree using CART algorithm....\n')
  _get_av_accuracy(dataset, 0.5, n)
```

dataset.py

```
import csv
import random
```

```
class Dataset:
    def __init__(self):
        self.name = "MOVIES"
self.learn_set = []
self.test_set = []
self.attribute_set = set()
    def get_data(self, database_name):
        read = open(database_name, 'r', encoding='utf-8')
        reader = csv.DictReader(read)
self.attribute_set = set(reader.fieldnames)
self.attribute_set.discard('title')
self.attribute_set.discard('revenue')
        for row in reader:
movie_dict = { 'revenue': 'rv' + row['revenue'],
                    'release': 're' + row['release'],
                    'prod_budget': 'b' + row['prod_budget'],
                    'genre': row['genre'],
                    'company': row['company']}
coin_toss = random.randint(0, 1)
            if coin_toss == 0:
self.learn_set.append(movie_dict)
elifcoin_toss == 1:
self.test_set.append(movie_dict)
```

tree.py

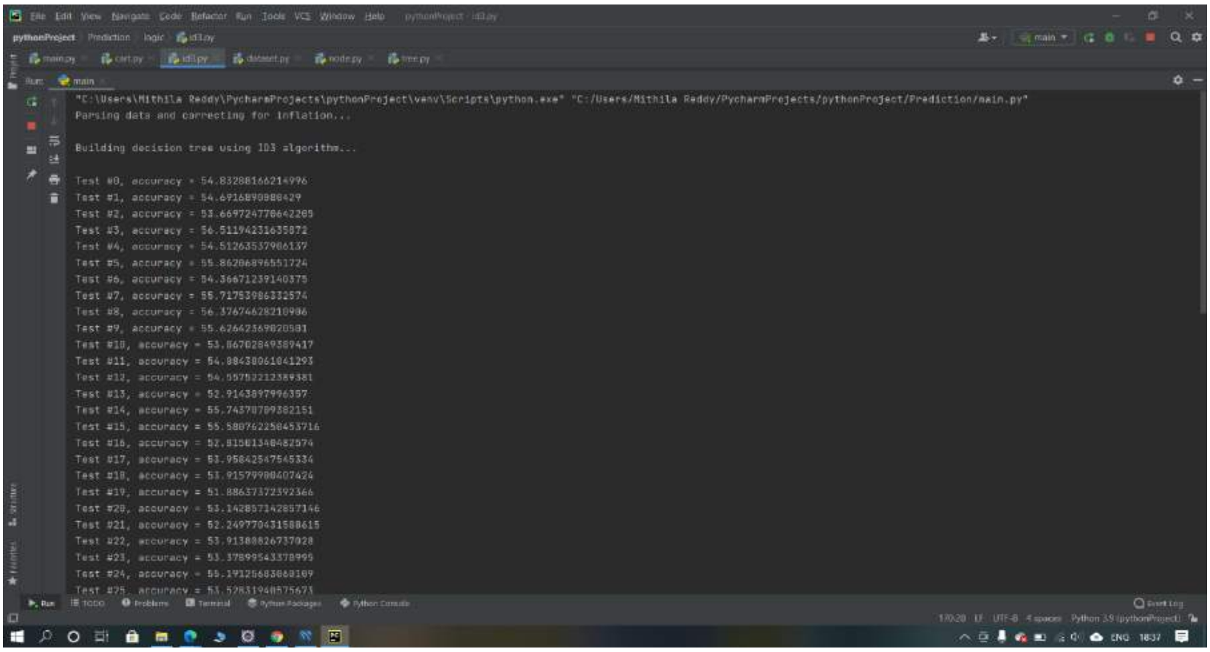
```
class Tree:
    def __init__(self, tree_root):
self.root = tree_root
self.rules = []
    def insert_rules(self):
```

```
self.read_in_rules(self.root, '')
    def read_in_rules(self, node, parent):
branch_length = len(node.branches)
curr = parent + "," + node.label + ","
        if branch_length != 0:
            for i in range(0, branch_length):
rule_part = curr + node.branches[i]
self.read_in_rules(node.children[i], rule_part)
        else:
curr = curr[1:-1]
self.rules.append(curr.split(','))
```

5. SCREENSHOTS

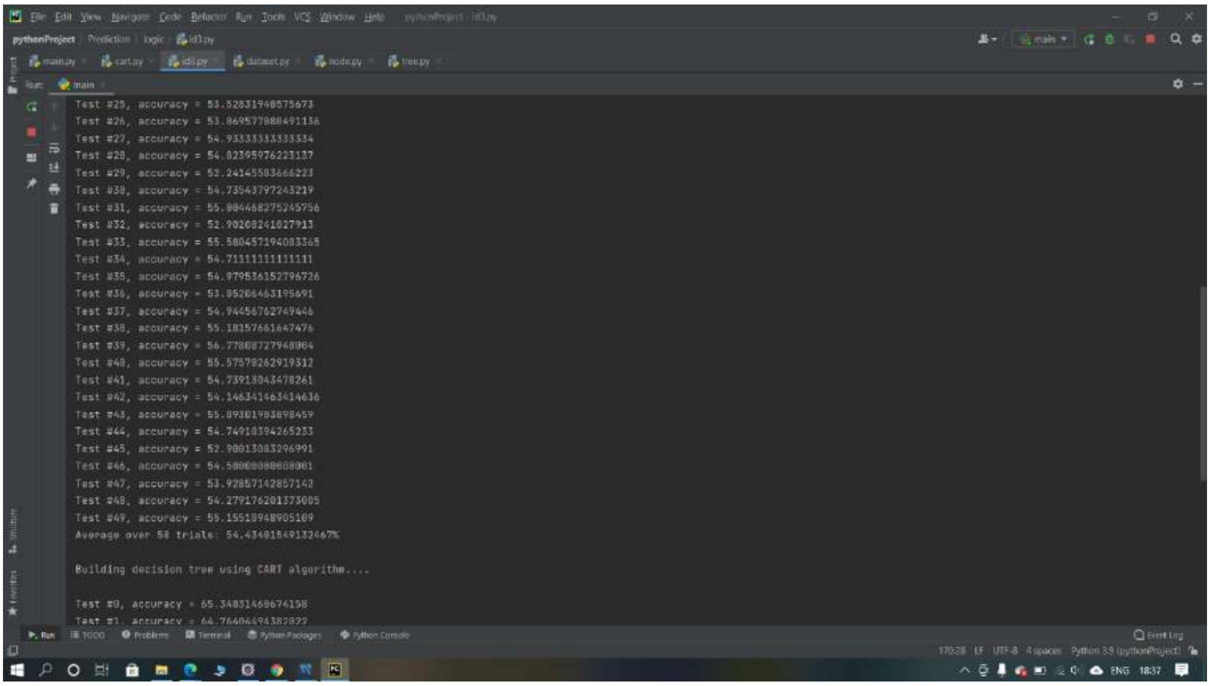
5. SCREENSHOTS

5.1 BUILDING TREE USING ID3 ALGORITHM



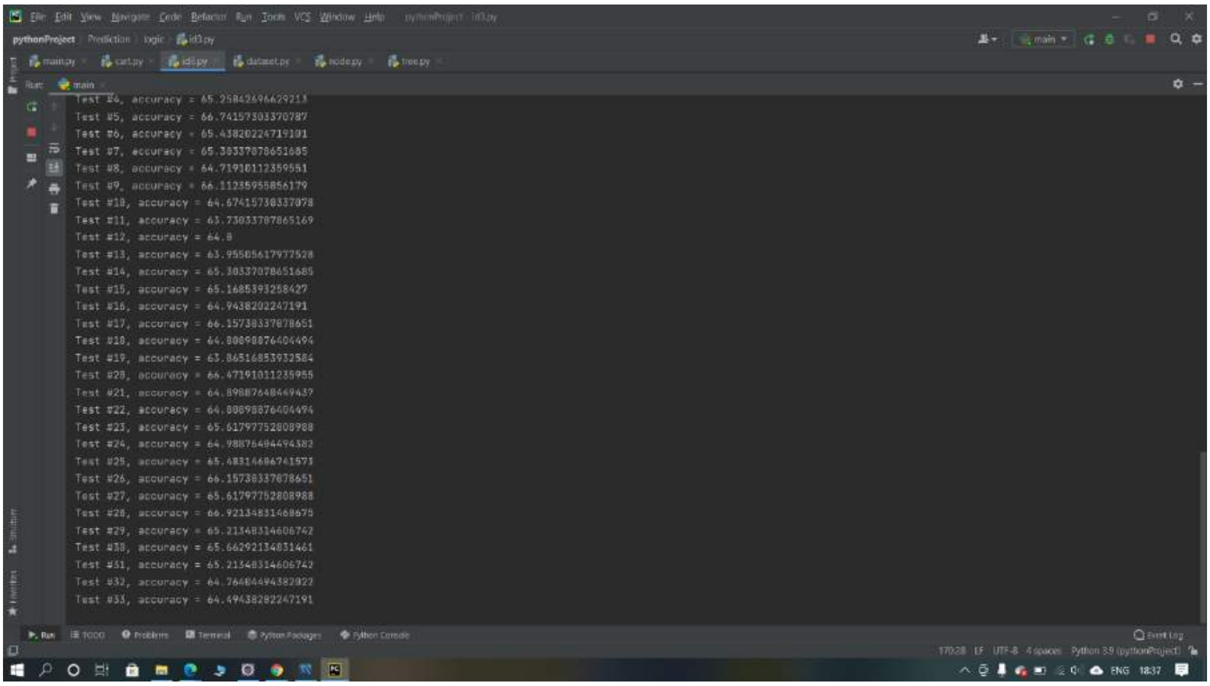
Screenshot 5.1: Building Decision Tree using ID3 Algorithm

5.2 TEST ACCURACY USING ID3 ALGORITHM



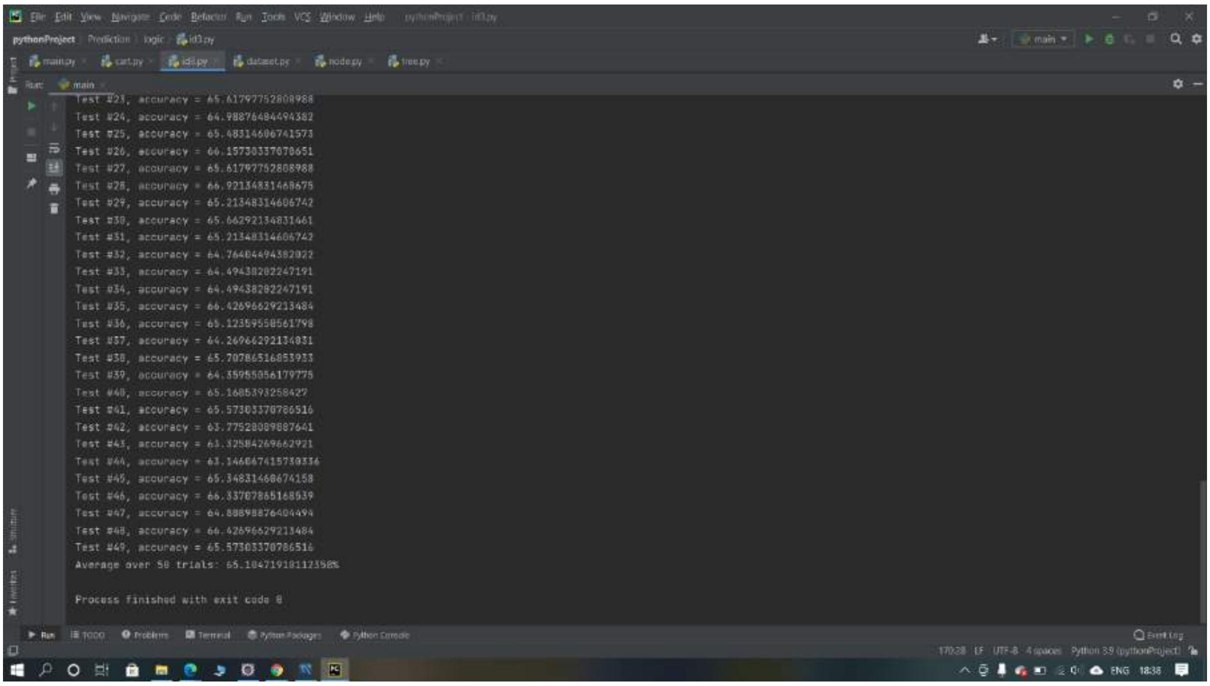
Screenshot 5.2: Test accuracy using ID3 Algorithm

5.3 BUILDING TREE USING CART ALGORITHM



Screenshot 5.3: Building Decision Tree using CART Algorithm

5.4 TEST ACCURACY USING CART ALGORITHM



Screenshot 5.4: Test Accuracy using CART Algorithm

5.5 THE MOVIE DATABASE(TMDB)

	A	B	C	D	E	F	G	H	I	J
1	budget	genres	homepage	id	keywords	original_la	original_ti	overview	popularity	productior
2	2.37E+08	[{"id": 28, '	http://ww	19995	[{"id": 146	en	Avatar	In the 22n	150.4376	[{"name":
3	3E+08	[{"id": 12, '	http://disr	285	[{"id": 270	en	Pirates of	Captain Ba	139.0826	[{"name":
4	2.45E+08	[{"id": 28, '	http://ww	206647	[{"id": 470	en	Spectre	A cryptic n	107.3768	[{"name":
5	2.5E+08	[{"id": 28, '	http://ww	49026	[{"id": 849	en	The Dark K	Following	112.313	[{"name":
6	2.6E+08	[{"id": 28, '	http://mo	49529	[{"id": 818	en	John Carte	John Carte	43.927	[{"name":
7	2.58E+08	[{"id": 14, '	http://ww	559	[{"id": 851	en	Spider-Ma	The seemi	115.6998	[{"name":
8	2.6E+08	[{"id": 16, '	http://disr	38757	[{"id": 156	en	Tangled	When the	48.68197	[{"name":
9	2.8E+08	[{"id": 28, '	http://mai	99861	[{"id": 882	en	Avengers:	When Ton	134.2792	[{"name":
10	2.5E+08	[{"id": 12, '	http://han	767	[{"id": 616	en	Harry Pott	As Harry b	98.88564	[{"name":
11	2.5E+08	[{"id": 28, '	http://ww	209112	[{"id": 849	en	Batman v	Fearing the	155.7905	[{"name":
12	2.7E+08	[{"id": 12, '	http://ww	1452	[{"id": 83, '	en	Superman	Superman	57.92562	[{"name":
13	2E+08	[{"id": 12, '	http://ww	10764	[{"id": 627	en	Quantum c	Quantum c	107.9288	[{"name":
14	2E+08	[{"id": 12, '	http://disr	58	[{"id": 616	en	Pirates of	Captain Ja	145.8474	[{"name":
15	2.55E+08	[{"id": 28, '	http://disr	57201	[{"id": 155	en	The Lone I	The Texas	49.04696	[{"name":
16	2.25E+08	[{"id": 28, '	http://ww	49521	[{"id": 83, '	en	Man of Ste	A young bo	99.39801	[{"name":
17	2.25E+08	[{"id": 12, "name": "A		2454	[{"id": 818	en	The Chron	One year a	53.9786	[{"name":
18	2.2E+08	[{"id": 878	http://mai	24428	[{"id": 242	en	The Aveng	When an u	144.4486	[{"name":
19	3.8E+08	[{"id": 12, '	http://disr	1865	[{"id": 658	en	Pirates of	Captain Ja	135.4139	[{"name":
20	2.25E+08	[{"id": 28, '	http://ww	41154	[{"id": 437	en	Men in Bla	Agents J (V	52.03518	[{"name":

	K	L	M	N	O	P	Q	R	S	T
1	productior	release_da	revenue	runtime	spoken_la	status	tagline	title	vote_aver	vote_count
2	[{"iso_316	########	2.79E+09	162	[{"iso_639	Released	Enter the W	Avatar	7.2	11800
3	[{"iso_316	########	9.61E+08	169	[{"iso_639	Released	At the end	Pirates of	6.9	4500
4	[{"iso_316	########	8.81E+08	148	[{"iso_639	Released	A Plan No	Spectre	6.3	4466
5	[{"iso_316	########	1.08E+09	165	[{"iso_639	Released	The Legen	The Dark K	7.6	9106
6	[{"iso_316	########	2.84E+08	132	[{"iso_639	Released	Lost in our	John Carte	6.1	2124
7	[{"iso_316	########	8.91E+08	139	[{"iso_639	Released	The battle	Spider-Ma	5.9	3576
8	[{"iso_316	########	5.92E+08	100	[{"iso_639	Released	They're tal	Tangled	7.4	3330
9	[{"iso_316	########	1.41E+09	141	[{"iso_639	Released	A New Age	Avengers:	7.3	6767
10	[{"iso_316	########	9.34E+08	153	[{"iso_639	Released	Dark Secre	Harry Pott	7.4	5293
11	[{"iso_316	########	8.73E+08	151	[{"iso_639	Released	Justice or	Batman v	5.7	7004
12	[{"iso_316	########	3.91E+08	154	[{"iso_639	Released		Superman	5.4	1400
13	[{"iso_316	########	5.86E+08	106	[{"iso_639	Released	For love, fo	Quantum c	6.1	2965
14	[{"iso_316	########	1.07E+09	151	[{"iso_639	Released	Jack is bac	Pirates of	7	5246
15	[{"iso_316	########	89289910	149	[{"iso_639	Released	Never Tak	The Lone I	5.9	2311
16	[{"iso_316	########	6.63E+08	143	[{"iso_639	Released	You will be	Man of Ste	6.5	6359
17	[{"iso_316	########	4.2E+08	150	[{"iso_639	Released	Hope has a	The Chron	6.3	1630
18	[{"iso_316	########	1.52E+09	143	[{"iso_639	Released	Some asse	The Aveng	7.4	11776
19	[{"iso_316	########	1.05E+09	136	[{"iso_639	Released	Live Forev	Pirates of	6.4	4948
20	[{"iso_316	########	6.24E+08	106	[{"iso_639	Released	They are b	Men in Bla	6.2	4160

Screenshot 5.5: The Movie Database(TMDB)

5.6 NEW_DATABASE GENERATED

	A	B	C	D	E	F	G
1	release	genre	revenue	prod_budg	title	company	
2	12	Action	5	3	Avatar	Other	
3	5	Adventure	5	3	Pirates of	Walt Disney Pictures	
4	10	Action	4	3	Spectre	Columbia Pictures	
5	7	Action	5	3	The Dark K	Other	
6	3	Action	2	3	John Carte	Walt Disney Pictures	
7	5	Fantasy	5	3	Spider-Ma	Columbia Pictures	
8	11	Animation	3	3	Tangled	Walt Disney Pictures	
9	4	Action	5	3	Avengers:	Other	
10	7	Adventure	5	3	Harry Pott	Warner Bros.	
11	3	Action	4	3	Batman v	Other	
12	6	Adventure	2	3	Superman	Other	
13	10	Adventure	3	2	Quantum c	Other	
14	6	Adventure	5	2	Pirates of	Walt Disney Pictures	
15	7	Action	1	3	The Lone F	Walt Disney Pictures	
16	6	Action	3	2	Man of Ste	Other	
17	5	Adventure	2	3	The Chron	Other	
18	4	Science Fic	5	2	The Aveng	Paramount Pictures	
19	5	Adventure	5	4	Pirates of	Walt Disney Pictures	
20	5	Action	3	2	Men in Bla	Other	

Screenshot 5.6: new_database2 generated from TMDB

5.7 RESULT DATASET

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	release	genre	revenue	prod_budg	title	company	Prediction						
2	10	Comedy	0	0	The Swindl	Other	0						
3	6	Drama	1	0	The Art of	Other	0						
4	3	Crime	1	1	Chappie	Columbia I	1						
5	1	Comedy	0	1	Down to Y	Miramax F	1						
6	6	Comedy	2	1	The Truma	Paramoun	1						
7	9	Drama	0	1	Treading V	Other	1						
8	10	Drama	1	1	The Color	Other	1						
9	3	Fantasy	2	2	Home	Twentieth	2						
10	12	Animation	1	1	Ernest & C	Other	1						
11	6	Adventure	1	1	The Beast	Other	1						
12	6	Thriller	1	1	The Neon	Other	1						
13	9	Action	1	1	The Train	Other	1						
14	7	Adventure	1	1	Red Cliff	Other	2						
15	3	Adventure	2	2	Wrath of t	Other	2						
16	12	Drama	1	1	Any Given	Other	1						
17	7	Romance	3	1	The Mask	New Line (	3						
18	2	Horror	0	0	Valentine	Village Roa	0						
19	4	Crime	1	1	The Interp	Universal I	1						
20	2	Romance	1	1	Against the	Paramoun	2						

Screenshot 5.7: result dataset generated with prediction values

6. TESTING

6. TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discoverevery 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 test. 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 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 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 centered 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.

6.3 TEST CASES

6.3.1 UPLOADING DATASET

Test case ID	Test case name	Purpose	Test case	Output
1	User uploads movie data	Use it for predicting profit	The user uploads the movie data	Uploaded successfully
2	User uploads 2nd movie data	Use it for predicting profit	The user uploads the movie data	Uploaded successfully

6.3.2 PREDICTION

Test case ID	Test case name	Purpose	Input	Output
1	Prediction test 1	To check if the predictor performs its task.	Movie data is given.	Profitability of movie data was predicted between 0 to 5.
2	Prediction test 2	To check if the predictor performs its task.	Movie data is given	Profitability of movie data was predicted between 0 to 5.
3	Prediction test 3	To check if the predictor performs its task.	Movie dataset is given(2000 movie data).	Profitability of each movie in dataset was predicted between 0 to 5.

7. CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

It is clear that predicting profit of a movie with a 100% accuracy can be difficult and with a large amount of data collected it becomes unclear which criteria are the best for-profit prediction. The project aims to predict movie's profitability before and after production in order to help investors and producer make a more informed decision where to invest in a movie or the effect of the budget on the retains from revenue. Our research aims to improve previous research by using a different type of classifier but based on previous related work, this might not be the case. For the rest of the report, we framed this problem to try and find the effective way to calculate the best splitting point using Gini index and Entropy. In general, we found that the decision tree based on the Gini index was a better classifier with an average accuracy of 63.79%, suitable to solve our problem. The lack of a pruning method proved to be a weakness in our implementations. For future work, we might consider using a pruning method in order to provide a more rigorous safeguard against high-variance or overfitting. This approach can be used as a support tool for prediction for upcoming movies.

7.2 FUTURE SCOPE

In future we can use other convolutional neural networks by downloading the modules directly into the project files. The software can be developed further to include lot of modules because the proposed system is developed on the view of future. We can connect to other data bases by including them.

8. BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

1. “ Theatrical Market Statistics Report”, Motion Picture Association of America, Inc., 2012.
2. Simonoff, J.S., Sparrow, I.R., Predicting movie grosses: winners and losers, blockbusters and sleepers, chance, 13(3), 2000.
3. Im, D., Nguyen, M. T., Predicting box-office success of movies in the U.S. market, CS 229, Univ. Stanford, 2011.10.
4. Walls, W. D. Modeling Movie Sucess When Nobody knows Anything: Conditional Stable Distribution Analysis Of Film Returns. Journal of Cultural Economics, 29, 3(August 2005),pp 177-190.
5. Han, Jiawei, Micheline Kamber, and Jian Pei. Data mining : concepts and techniques. Amsterdam: Elsevier/Morgan Kaufmann, 2012.

8.2 WEBSTIES

1. http://www.businessinsider.com/google-study-can-predict-sucess-of-movies-2013-6.
2. http://www.tmdb.com/interfaces/
3. http://www.mpaa.org/wp-content/uploads/2014/03/2012-Theatrical-Market-Ststistics-report.pdf.