

Artificial Intelligence Project Report on Akinator

AUTHORS (alphabetical order):

K B Poovanna

Praharsh G

Tanya Tripathi

COURSE NAME:

AI Project (Akinator)

PROGRAM:

BTech CSE (CPS)

SCHOOL:

SCOPE

FACULTY:

Prof. R. Vedhapriyavadhana

DATE OF SUBMISSION:

08-02-2023

Table of Contents

S.No.	Description	Page no.
1.	Introduction & Overview of AI	3
2.	Problem Definition	3
3.	Source Code	4
4.	Output	8
5.	Conclusion	10

Introduction & Overview of AI

Since the invention of computers or machines, their capability to perform various tasks went on growing exponentially. Humans have developed the power of computer systems in terms of their drivers' working domains, their increasing speed, and reducing size with respect to time.

A branch of Computer Science named Artificial Intelligence pursues creating computers or machines as intelligent as human beings. The research field of Artificial Intelligence (AI) is concerned with making machines - Particularly computers - do things that require intelligence when done by humans. In the 60 years of its existence, it has celebrated dramatic successes and equally dramatic failures. Today, AI has become an important and essential part of technology and industry and provides solutions to some of the most complex problems in computer science. Nevertheless, in terms of its original goal to create true human-level intelligence in machines - Strong AI has not succeeded yet and perhaps never will. Today, AI researchers are able to create computers that can perform jobs that are difficult for persons like logic, algebra problem solving, and path planning, or playing chess.

Problem Definition

Akinator is an app that uses artificial intelligence to guess something by asking a series of questions with "Yes", "No", "Probably", "Probably not" and "Don't know" as possible answers, to narrow down the potential item. If the answer is narrowed down to a single likely option before 25 questions are asked, the program will automatically ask whether the item it chose is correct.

We have designed an akinator which guesses the movie you're thinking about. It basically rules out all the options by the way you answer the asked questions and display the one you're thinking about. The questions are about the genre, director, actors and rating of the movie. If the asked questions are accurately answered, then the akinator shows the desired output.

Source Code

HTML index page

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>IMDB AKINATOR</title>
    <link rel="stylesheet" href="style.css">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/boo
tstrap.bundle.min.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIax
VXM" crossorigin="anonymous"></script>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/b
ootstrap.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfSpd3yD65VohhpuaC0mLA
SjC" crossorigin="anonymous">
<link rel="stylesheet"
href="https://pyscript.net/latest/pyscript.css" />
<script defer
src="https://pyscript.net/latest/pyscript.js"></script>
  </head>

  <header>

    <!-- navbar -->

    <nav class="navbar fixed-top navbar-expand-lg navbar-dark
bg-dark ">
```

```

    <a class="navbar-brand" href="#" id="NavbarTitle">
      <button type="button" class="btn btn-warning">IMDB
AKINATOR</button>
    </a>
    <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse"
id="navbarNavAltMarkup">
      <div class="navbar-nav">
        <a class="nav-item nav-link active"
onClick="document.getElementById('welcome').scrollIntoView();"
" href="#">Home </a>
        <a class="nav-item nav-link"
href="akinator.html">Akinator</a>
        <a class="nav-item nav-link"
onClick="document.getElementById('end').scrollIntoView();"
href="#">How it works</a>
        <a class="nav-item nav-link" href="#">About Us</a>
      </div>
    </div>
</nav>

</header>

<body>

```



```

    </div>

</div>

</div>

</section>

<!-- description -->

<section id="description">
  <div class="row-6 row">
    <div class="col-lg-4">
      
    </div>
    <div class="col-lg-8">

      <p id="descriptPara">Akinator is an app that uses
artificial intelligence to guess something by asking a series
of questions with "Yes", "No", "Probably", "Probably not" and
"Don't know" as possible answers, to narrow down the
potential item. If the answer is narrowed down to a single
likely option before 25 questions are asked, the program will
automatically ask whether the item it chose is correct. We
have designed an akinator which guesses the movie you're

```

thinking about. It basically rules out all the options by the way you answer the asked questions and display the one you're thinking about. The questions are about the genre, director, actors and rating of the movie. If the asked questions are accurately answered, then the akinator shows the desired output.</p>

</div>

</div>

</section>

</body>

<footer>

<!-- status bar -->

<div class="footer" id="end">

(c) 2023 Copyright: IMDB Akinator.com ♥

</div>

</footer>

</html>

HTML akinator

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>OLLA</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body  >

    <div id="code">
      <iframe
src="https://trinket.io/embed/python3/3a0d1bfb55?outputOnly=true&runOption=run" width="50%" height="300" frameborder="0"
marginwidth="300" marginheight="500"
allowtransparency="true"></iframe>
    </div>

  </body>
</html>
```

CSS

```
#NavbarTitle{
  margin-left: 30px;
  margin-right:800px;
  font-size: 2em;
}

.navbar
{
  position: fixed;
}

.nav-item
{
  margin-left:5px;
  margin-right:20px;
}

.nav
{
  background-color: #000;
}

#welcome
{
  padding:15% 2% 7% 0%;
  background-color: #FED049;
}

#hello
{
  padding: 0% 5% 10% 3%;
}
```

```
}

#helloTitle
{
  padding:0% 15%;
  text-align:left;
  color: #000;
  font-size:3.5rem;
}

#button
{
  padding:2% 1.5%;
  margin-top:4%;
  margin-left:15%;
}

#poster
{
  height:15rem;
  width:10rem;
}

#description
{
  padding:7% 2%;
}

#descript
```

```
{  
  height: 20rem;  
  width: 26rem;  
}  
  
#descriptPara  
{  
  text-align: center;  
  padding: 5rem;
```

```
}  
  
#creators
```

```
{  
  padding: 7% 2%;  
  background-color: #FED049;  
}  
  
#creatorPic  
{  
  border-radius: 50%;  
  height: 15rem;  
  width: 13rem;  
  margin-left: 7em;  
}  
  
#creatorID  
{
```

```
margin:1.5rem 5rem;
text-align: center;

}

.footer
{

background-color: #000;
color: #FFF;
padding:1% 0% 1% 40%;

}
```

Python for functionality

```
import numpy as np
import pandas as pd
import random

df =
pd.read_csv("/Users/praharshgurudatta/Desktop/imdb_top_1000.csv")

Genres = {}
actors = {}
certificate = {}
directors = {}

def take_out():
    genres = df["Genre"].str.split(expand = True)
    genres = genres.replace(',', '', regex = True)
    Genres =
set(genres[0]).union(set(genres[1])).union(set(genres[2]))
    Genres.remove(None)

    certificate = set(df["Certificate"])
    directors = set(df["Director"])

    actors = df[['Star1', 'Star2', 'Star3', 'Star4']]
    actors =
set(actors['Star1']).union(set(actors['Star2'])).union(set(ac
```

```

tors['Star3'])).union(set(actors['Star4']))
    return actors,certificate,directors,Genres,genres

def questioning():

    #actors,certificate,directors,Genres = take_out()
    actor = random.choice(tuple(actors))
    cf = random.choice(tuple(certificate))
    director = random.choice(tuple(directors))
    genre = random.choice(tuple(Genres))
    Questions = ["Is "+str(actor)+" in the movie?",
                  "Is the movie rated "+str(cf)+"?",
                  "Is "+str(director)+" The Director of the
movie?",
                  "Does "+str(genre)+" describe your movie?"]
    i = Questions.index(random.choice(Questions))
    return i,Questions[i],actor,cf,director,genre

asf = 0
cf = 0
dirf = 0
gen = 0

while(len(df.axes[0])>1):

    actors,certificate,directors,Genres,genres = take_out()
    ind,question,actor,certi,director,genr = questioning()
    if((asf>4 and ind == 0) or (cf>0 and ind == 1) or (dirf>0
and ind == 2) or (gen>3 and ind == 3)):
        continue
    print(question)

```

```

ans = input("Yes or No: ")
if(ind == 0):
    actors.remove(actor)
    if(ans == "Yes"):
        for ind in df.index:
            if(df['Star1'][ind]!=actor and
df['Star2'][ind]!=actor and df['Star3'][ind]!=actor and
df['Star4'][ind]!=actor):
                df = df.drop(ind)
                asf = asf+1
        else:
            for ind in df.index:
                if(df['Star1'][ind]==actor and
df['Star2'][ind]==actor and df['Star3'][ind]==actor and
df['Star4'][ind]==actor):
                    df = df.drop(ind)

elif(ind == 1):
    certificate.remove(certifi)
    if(ans == "Yes"):
        for ind in df.index:
            if(df['Certificate'][ind]!=certifi):
                df = df.drop(ind)
                cf = cf+1
        else:
            for ind in df.index:
                if(df['Certificate'][ind]==certifi):
                    df = df.drop(ind)

elif(ind == 2):
    directors.remove(director)

```



```

    if(ans == "Yes"):
        for ind in df.index:
            if(df['Director'][ind]!=director):
                df = df.drop(ind)
                dirf = dirf+1
        else:
            for ind in df.index:
                if(df['Director'][ind]==director):
                    df = df.drop(ind)

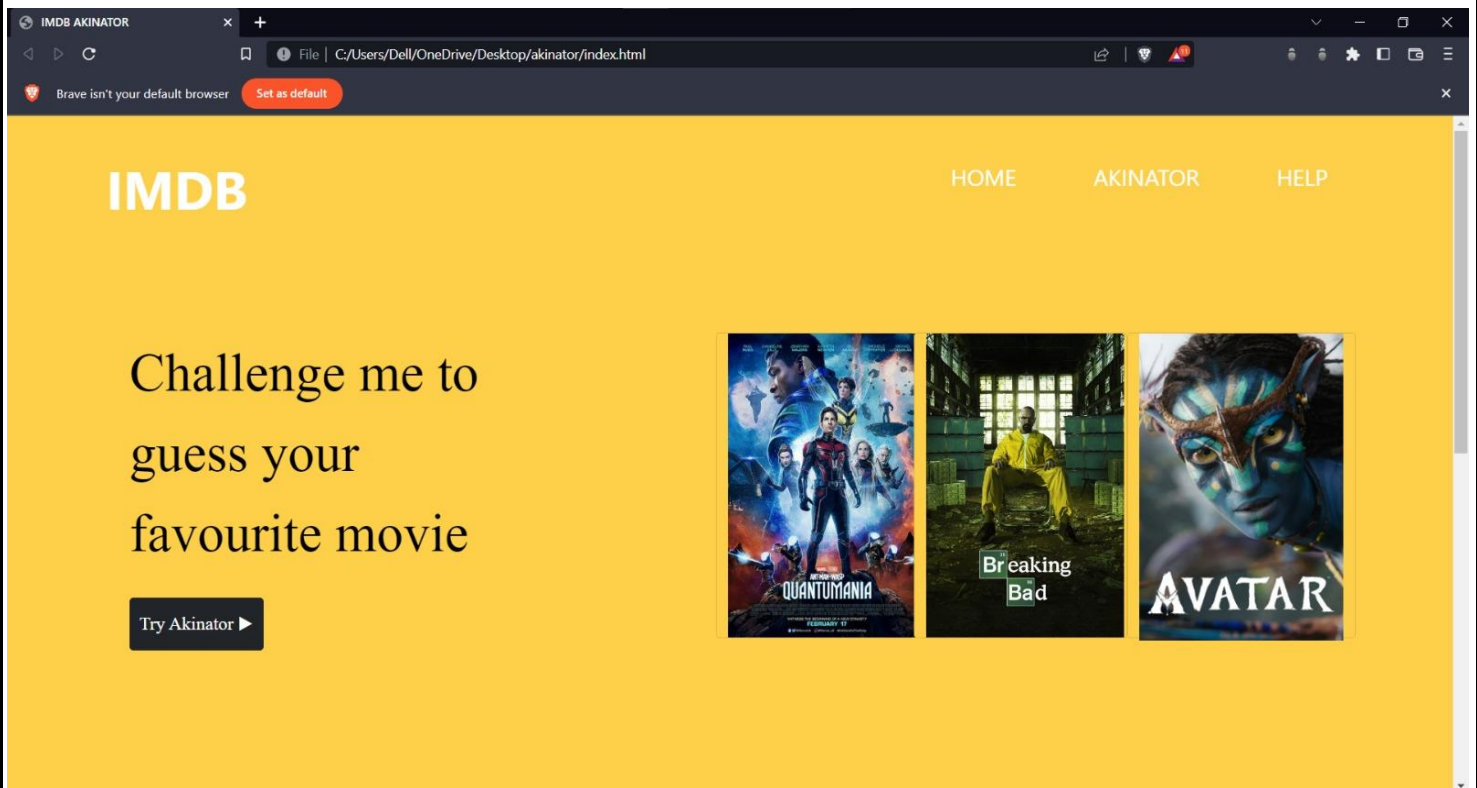
elif(ind == 3):
    Genres.remove(genr)
    if(ans == "Yes"):
        for ind in df.index:
            if(genres[0][ind]!=genr and
genres[1][ind]!=genr and genres[2][ind]!=genr):
                df = df.drop(ind)
                genres = genres.drop(ind)
        else:
            for ind in df.index:
                if(genres[0][ind] == genr and genres[1][ind]
== genr and genres[2][ind] == genr):
                    df = df.drop(ind)
                    genres = genres.drop(ind)

if df.empty:
    print("There is no movie like this")
else:
    print(df['Series_Title'])

```

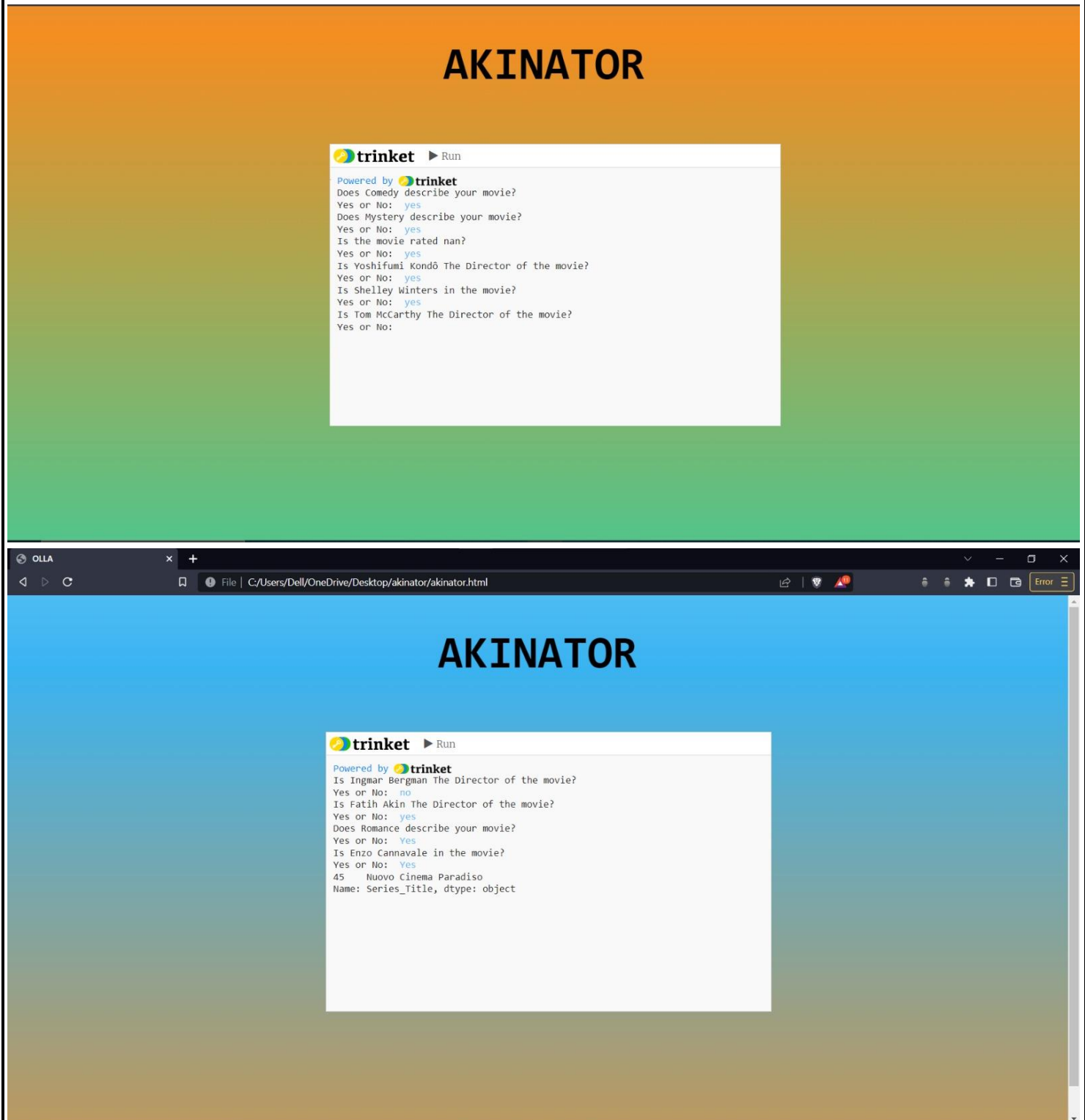
Output

1.This is the opening page which greets the user when entered.



Akinator is an app that uses artificial intelligence to guess something by asking a series of questions with "Yes", "No", "Probably", "Probably not" and "Don't know" as possible answers, to narrow down the potential item. If the answer is narrowed down to a single likely option before 25 questions are asked, the program will automatically ask whether the item it chose is correct. We have designed an akinator which guesses the movie you're thinking about. It basically rules out all the options by the way you answer the asked questions and display the one you're thinking about. The questions are about the genre, director, actors and rating of the movie. If the asked questions are accurately answered, then the akinator shows the desired output.

2. When the user clicks the “Try Akinator” it takes to the akinator page where a terminal is provided .
3. Based on the questions the user has to provide Yes or No answers based on which the backend will throw the right movie name.



Conclusion

To conclude, Akinator is a fun game to play even though all it does is guess the movie by asking a bunch of questions about it. Similarly, many such recreational games can be created, based on the basic AI concepts.

It can also be used as a movie recommender, it recommends the movie based on the answers given.
