Chapter 1

COMPANY PROFILE

Varcons Technologies Private Limited is a Private incorporated on 11 July 2022. It is classified as Non-govt Company and is registered at Registrar of Companies, Bangalore. Its authorized share capital is Rs. 1,000,000 and its paid up capital is Rs. 10,000. It is involved in other computer related activities [for example maintenance of websites of other firms/ creation of multimedia presentations for other firms etc.

Directors of Varcons Technologies Private Limited are Chikaegowdanadoddi Kariyappa Somalatha and Haralahalli Chandraiah Spoorthi

Varcons Technologies Private Limited's Corporate Identification Number is (CIN) U72900KA2022PTC163646 and its registration number is 163646.Its Email address is ca.mittalankushjain@gmail.com and its registered address is #8/9, 5th Main, 3rd Cross road, Beside Sachidananada Nagar, R R Nagar Bangalore KA 560098

Chapter 2

About the company

Varcons Technologies is a leading provider of cutting-edge technologies and services, offering scalable solutions for businesses of all sizes. Founded by a group of friends who started by scribbling their ideas on a piece of paper, today we offer smart, innovative services to dozens of clients. We develop SaaS products, provide Corporate Seminars, Industrial trainings and much more

At VCT, We make sure every product/service that we offer is built keeping in mind the practical usability of the product/Service, We're a startup focused on Creativity and Customizability, and We also provide subscription models for Software that we have already built, Since the application is already configured, the user has a ready-to-use application. This not only reduces installation and configuration time but also cuts down the time wasted on potential glitches linked to software deployment

2.1 Services provided by varcons Technologies

VCA provides a host of services to its customers/users/clients, enabling business success driven by technology Harnessing the power of technology, we create a measurable difference for our clients across various industries & multiple geographies.

Website as Software

We develop websites that behave and interact similar to sophisticated software. Search Engine Optimization we help you manage your SEO campaign more efficiently and effectively. We help you gain market share by leveraging our expertise. Our holistic approach to identify anything that may be hurting your traffic or rankings and show you just how to outrank the competition.

Comprehensive Customer Support

With a comprehensive range of services, we guarantee your technology needs are not just met, but exceeded. We shall work with your customers/users closely to understand the way your users/customers use/make use of products/services Branding and Design We offer professional Graphic design, Brochure design & Logo design.

Analytics and Research

We analyse the way your users/customers interact with you/your business by gathering, studying and understanding the consumer voice and their perception of the product/service

Embedded Systems and IOT

We work with Consumer Electronics, Lighting, Home Automation, Metering, Sensor-Technology, Home Appliance and Medical Device companies to help them create smart and connected products. Through its integrated Embedded and IoT services, VCA helps build intelligent & connected devices that can be remotely monitored and controlled while leveraging edge and cloud computing for a host of intelligent applications and analytics.

Chapter 3

INTRODUCTION

The target for this Industrial Training is those who wish to quickly get started in the area of data science and machine learning. We got an overview of the current and most popular libraries with a focus on Python, however we will mention alternatives in other languages where appropriate. All tools presented here are free and open source, and many are licensed under very flexible terms (including, for example, commercial use). Each library will be introduced, code will be shown, and typical use cases will be described.

Machine learning (ML) is a branch of artificial intelligence (AI) that enables computers to “self-learn” from training data and improve over time, without being explicitly programmed. Machine learning algorithms are able to detect patterns in data and learn from them, in order to make their own predictions. In short, machine learning algorithms and models learn through experience. In traditional programming, a computer engineer writes a series of directions that instruct a computer how to transform input data into a desired output. Instructions are mostly based on an IF-THEN structure: when certain conditions are met, the program executes a specific action.

Machine learning itself is a fast-growing technical field and is highly relevant topic in both academia and in the industry. It is therefore a relevant skill to have in both academia and in the private sector. It is a field at the intersection of informatics and statistics, tightly connected with data science and knowledge discovery. The prerequisites for this training are therefore a basic under-standing of statistics, as well as some experience in any C-style language. Some knowledge of Python is useful but not a must.

Chapter 4

RELATED WORKS IN THE FIELD

There have been many research studies and practical applications of movie recommendation systems.

Netflix Prize: In 2006, Netflix launched a competition, known as the Netflix Prize, to improve its movie recommendation algorithm. The competition offered a $1 million prize for the team that could improve the algorithm's accuracy by 10%. The competition lasted for three years and led to significant advancements in the field of recommendation systems.

Content-based recommendation system: A significant related work in the field is the content-based recommendation system developed by Pazzani and Billsus in 2007. Their system utilized user ratings and reviews, along with movie metadata such as genre, cast, director, and plot, to recommend movies to users. Collaborative filtering recommendation system: Another important work is the collaborative filtering recommendation system developed by Resnick and Varian in 1997. Their system utilized user ratings to recommend movies to users with similar tastes.

Hybrid recommendation system: A recent study by Jin and Park in 2019 developed a hybrid recommendation system that combines both content-based and collaborative filtering approaches. Their system utilized a neural network to learn the relationships between movies and users and provide personalized recommendations. Deep learning-based recommendation system: Another significant related work is the deep learning-based recommendation system developed by The et al. in 2017. Their system utilized a deep neural network to learn the user-item relationships and provided more accurate recommendations compared to traditional collaborative filtering methods.

These works have significantly advanced the field of movie recommendation systems and have led to the development of more accurate and personalized recommendation algorithms.

4.1 Machine Learning Packages Used

Fig.1 Machine Learning Packages Used

- ### Pandas

Pandas are a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

- ### NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- ### Ast

The Ast library in Python helps us work with abstract syntax trees (ASTs). An AST is a way of representing the structure of code in a computer program. Think of it as a map or a blueprint of a program.

- ### CountVectorizer

The CountVectorizer library in Python helps us convert text into numbers that a computer can understand. It's like a translator that turns words into numbers.

- **Cosine similarity**

  Cosine similarity is a way of measuring how similar two things are. It's often used in computer science to measure the similarity between two pieces of text or two sets of data.The basic idea behind cosine similarity is to imagine the data as vectors in space. A vector is just a fancy word for an arrow with a direction and a magnitude (or length). The direction of the vector represents the relationship between different parts of the data, and the magnitude represents the strength of that relationship.

- **Pickle**

  The Pickle library in Python is used to save and load data in a way that makes it easy to use later. Think of it like a way to save your progress in a video game so you can pick up where you left off later.

- **Streamlit**

  Streamlit is a library in Python that makes it easy to build interactive web applications. Think of it like building your own website or app, but without needing to know a lot of complex coding languages.

- **Requests**

  The Requests library is a tool in Python that makes it easy to send and receive information from websites and web applications. Think of it like sending a message to a friend and getting a response back

- **Instaloader**

  Instaloader is a tool in Python that helps you download pictures, videos, and other content from Instagram. Think of it like saving a picture or video that you like on Instagram to your phone, but on your computer instead.

Chapter 5

METHODOLOGY

5.1 Data Collection

1. **Identify sources of movie data:** The first step in data collection is to identify sources of movie data. This can include public movie databases, social media platforms, movie review websites, and other online sources. Some popular sources of movie data include IMDb, Rotten Tomatoes, and The Movie Database (TMDb).
2. **Collect data about movies**: Once the sources of movie data are identified, the next step is to collect data about the movies. This can include information such as movie titles, genres, release dates, actors, directors, and plot summaries.
3. **Filter and clean the data**: After collecting the data, it needs to be filtered and cleaned to remove any inconsistencies, duplicates, or missing values. For example, some movie databases may contain multiple entries for the same movie, or may have missing data for some movies.
4. **Organize the data**: Once the data is filtered and cleaned, it needs to be organized in a suitable format. This can involve structuring the data into tables or spreadsheets, or using a database management system to store the data.This data needs to be filtered and cleaned to ensure that the dataset is accurate and complete.
5. **Store the data**: The final step in data collection is to store the data in a format that can be easily accessed and analyzed. This can involve using a cloud-based storage system, or storing the data on a local server or computer.,

.

5.2 Data Preprocessing

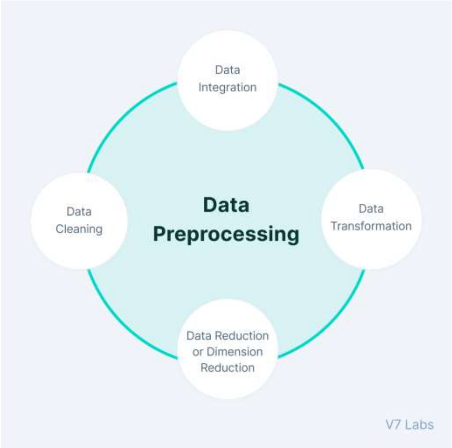
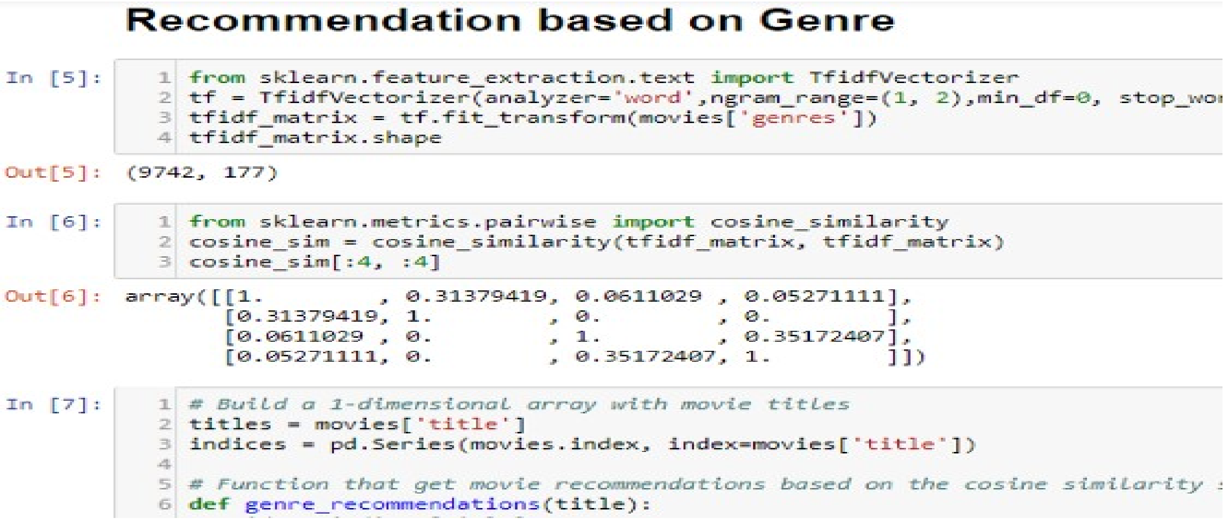


Fig.2 Data Preprocessing

Data cleaning the first step in data pre-processing is data cleaning, which involves identifying and handling missing or erroneous data. This can be done by removing duplicate records, filling in missing values, or removing records with incomplete data. For example, if the dataset contains records with missing movie genres, the missing values. Data Transformation The next step in data pre-processing is data transformation, which involves converting the data into a suitable format for analysis. This can involve scaling the data, converting categorical data into numerical data, or normalizing the data to remove any biases or outliers. For example, if the dataset contains movie release dates in different formats, the dates can be transformed into a uniform date format for easier analysis.

Feature Selection After data transformation, the next step is feature selection, which involves selecting the most relevant features for analysis. This can involve using techniques such as principal component analysis (PCA) or feature importance scores to identify the most important features that contribute to a movie's similarity. For example, if the dataset contains movie features such as title, genre, director, and cast, feature selection can be used to identify the most important features for movie similarity calculation. Feature Encoding: The final step in data pre-processing is feature encoding, which involves converting categorical data into numerical data for analysis. This can involve techniques such as one-hot encoding or label encoding to convert categorical data into numerical data that can be used in similarity calculations. For example, if the dataset contains movie genres, feature encoding can be used to convert the genre data into numerical data that can be used to calculate similarity scores between movies.

5.3 Similarity calculation



Recommendation based on Genre

```
In [5]:  from sklearn.feature_extraction.text import TfidfVectorizer
         tf = TfidfVectorizer(analyzer='word',ngram_range=(1, 2),min_df=0, stop_wor
         tfidf_matrix = tf.fit_transform(movies['genres'])
         tfidf_matrix.shape

Out[5]:  (9742, 177)

In [6]:  from sklearn.metrics.pairwise import cosine_similarity
         cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
         cosine_sim[:4, :4]

Out[6]:  array([[1.        , 0.31379419, 0.0611029 , 0.05271111],
                [0.31379419, 1.        , 0.        , 0.        ],
                [0.0611029 , 0.        , 1.        , 0.35172407],
                [0.05271111, 0.        , 0.35172407, 1.        ]])

In [7]:  # Build a 1-dimensional array with movie titles
         titles = movies['title']
         indices = pd.Series(movies.index, index=movies['title'])

         # Function that get movie recommendations based on the cosine similarity :
         def genre_recommendations(title):
```

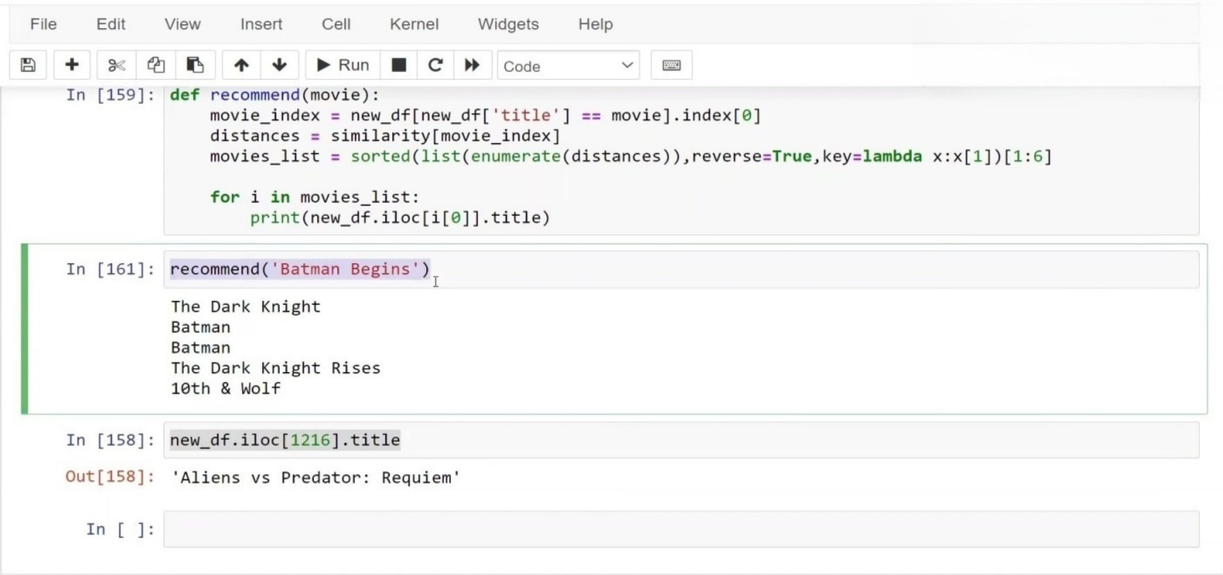
Fig.3 Similarity calculation

Define Similarity Metric The first step in similarity calculation is to define a similarity metric that can be used to compare movies. This can involve using techniques such as cosine similarity, Euclidean distance, or Jaccard similarity to calculate the similarity scores between the feature vectors. Calculate Similarity Scores: Once the similarity metric is defined, the next step is to calculate the similarity scores between the movies. This can involve calculating the similarity score between each pair of movies using the defined similarity metric. For example, cosine similarity can be used to calculate the similarity score between each pair of movies based on their feature vectors.

Sort Movies by Similarity Scores: After calculating the similarity scores, the next step is to sort the movies by their similarity scores. This can involve ranking the movies based on their similarity scores in descending order, so that the most similar movies appear at the top of the list. Filter Movies: The final step in similarity calculation is to filter the movies based on certain criteria, such as popularity or release date. This can involve selecting the top-rated movies from the similarity scores, or selecting the most recent movies based on their release date.

Rank the Movies: The first step in ranking and filtering is to rank the movies based on their similarity scores or other criteria such as user ratings, popularity, or release date. The ranking can be based on a single criterion or a combination of multiple criteria. Apply Filters: After ranking the movies, the next step is to apply filters to narrow down the recommendations. The filters can be based on factors such as genre, language, release date, or popularity. For example, the system may recommend only movies that belong to a certain genre, or that were released within a specific time frame. Generate the Final Recommendations: After applying the filters, the final step is to generate the recommendations. This can involve presenting the top-ranked movies that meet the filtering criteria to the user, along with additional information such as movie trailers, reviews, and ratings. The recommendations can be presented in the form of a list or grid, with each movie represented by an image or thumbnail.

5.4 Model Evaluation



```
In [159]: def recommend(movie):
              movie_index = new_df[new_df['title'] == movie].index[0]
              distances = similarity[movie_index]
              movies_list = sorted(list(enumerate(distances)),reverse=True,key=lambda x:x[1])[1:6]

              for i in movies_list:
                  print(new_df.iloc[i[0]].title)

In [161]: recommend('Batman Begins')

          The Dark Knight
          Batman
          Batman
          The Dark Knight Rises
          10th & Wolf

In [158]: new_df.iloc[1216].title

Out[158]: 'Aliens vs Predator: Requiem'

  In [ ]:
```

Fig.4 Model Evaluation

Define Evaluation Metrics: The first step in model evaluation is to define the evaluation metrics that will be used to measure the performance of the recommendation system. These metrics can include measures such as accuracy, precision, recall, F1-score, or mean average precision (MAP). The choice of evaluation metric will depend on the specific goals of the recommendation system and the type of data being used. Here is a step-wise explanation of how to use Google Teachable Machine to train a custom machine learning model. Split the Data: The next step is to split the data into training and testing sets. The training set is used to train the recommendation model, while the testing set is used to evaluate its performance. The split can be done randomly or using a pre-defined method such as k-fold cross-validation.

Train the Model: After splitting the data, the next step is to train the recommendation model using the training set. This can involve using techniques such as collaborative filtering, content-based filtering, or hybrid models to generate recommendations. Evaluate the Model Once the model is trained, the next step is to evaluate its performance using the testing set. This can involve measuring the model's accuracy, precision, recall, F1-score, or MAP using the evaluation metrics defined in the first step. Tune the Model: If the model's performance is not satisfactory, the next step is to tune the model by adjusting its parameters or using different techniques. This can involve experimenting with different similarity metrics, collaborative filtering algorithms, or regularization techniques to improve the model's performance. Repeat the Process: Once the model is tuned, the process can be repeated by splitting the data into new training and testing sets and training the model again. This can be done iteratively until the desired level of performance is achieved.

5.5 Deployment

Select Deployment Platform The first step in deployment is to select a deployment platform that meets the requirements of the recommendation system. This can involve choosing between cloud-based platforms such as Amazon Web Services (AWS) or Google Cloud Platform (GCP), or on-premises deployment options such as a dedicated server or a local network. Configure the Environment: Once the deployment platform is selected, the next step is to configure the environment for the recommendation system. This can involve setting up the necessary software packages, libraries, and dependencies to ensure that the system can run smoothly in the deployment environment. Test the System: After configuring the environment, the next step is to test the recommendation system in the deployment environment.

This can involve performing unit tests, integration tests, and load tests to ensure that the system can handle the expected traffic and user load. Deploy the System: Once the system is tested and ready, the next step is to deploy it in the production environment. This can involve configuring the system for optimal performance, setting up monitoring and logging tools to track system performance and user behaviour, and implementing security measures to protect against potential threats.

Deploy the System once the system is tested and ready, the next step is to deploy it in the production environment. This can involve configuring the system for optimal performance, setting up monitoring and logging tools to track system performance and user behaviour, and implementing security measures to protect against potential threats.

Monitor and Maintain the System After the system is deployed, the next step is to monitor and maintain it to ensure that it continues to function properly over time. This can involve monitoring system performance, user feedback, and usage patterns to identify potential issues or areas for improvement, and performing regular maintenance tasks such as updating software packages, fixing bugs, and adding new features.

Overall, the deployment step in movie recommendation systems involves selecting a deployment platform, configuring the environment, testing the system, deploying it in the production environment, and monitoring and maintaining the system over time. The specific techniques used may vary depending on the deployment platform and the requirements of the recommendation system.

Chapter 6

IMPLEMENTATION

6.1 Content-based movie recommendation systems

A movie recommendation system is a type of personalized recommender system that suggests movies to users based on their past viewing behaviour, ratings, and other relevant factors such as genre, cast, director, etc. These systems are designed to provide users with personalized and convenient ways of finding new movies to watch..

In this project, we will be using Python to build a movie recommendation system. We will train a train model using a dataset of movies dataset of different types of movies and then use this model to classify personalized recommender system that suggests movies to users.

- To get started, we will need to:
  1. Collect and pre-process the dataset of movies.
  2. Train a model using the dataset.
  3. Test the model on a set of validation movies data to check its accuracy.
  4. Use the trained model to suggests movies to users.
- To accomplish these steps, we will be using the following tools and libraries:
  1. Python - a general-purpose programming language.
  2. Streamlit - an open-source Python framework that allows data scientists and developers to quickly build and deploy web applications.

6.2 Creating model classsifier

1. **Gather Data:** Gather a dataset of movies along with their attributes. Some of the attributes can include genre, director, actors, keywords, rating, etc. This data can be collected from websites such as IMDb, Rotten Tomatoes, or TMDb.

2. **Feature Engineering:** Extract features from the movie data. For example, you can create a feature vector for each movie that contains information about its genre, actors, and director. One-hot encoding can be used to represent categorical variables such as genre.

3. **User Profile:** Create a user profile based on the movies the user has liked in the past. This profile can be created by aggregating the feature vectors of the movies the user has rated highly.,

4. **Similarity Score:** Calculate a similarity score between the user profile and the feature vectors of the other movies in the dataset. Cosine similarity is a commonly used metric for calculating similarity.

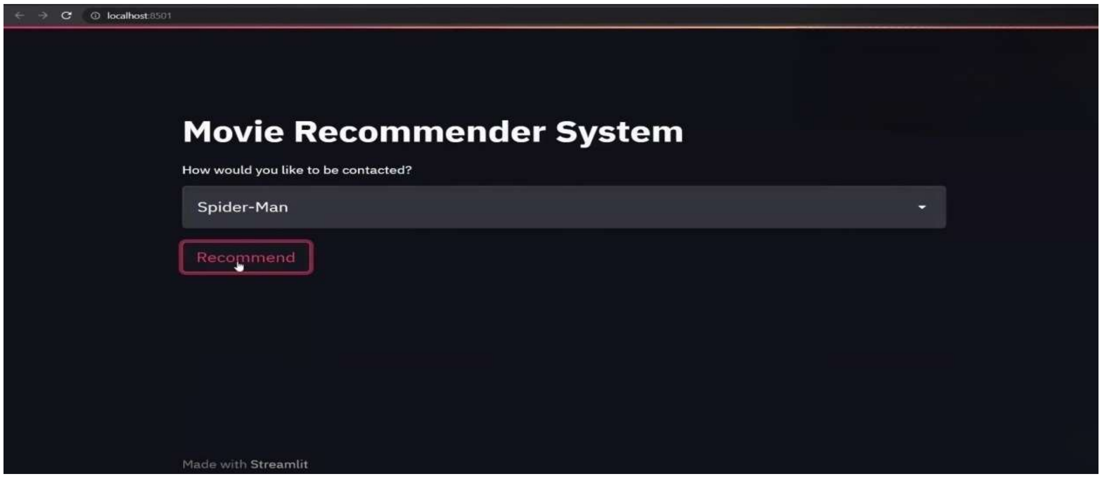
5. **Recommend Movies:** Recommend the top movies with the highest similarity score to the user.

6. **Evaluation:** Evaluate the performance of the recommendation system using metrics such as precision, recall, and F1-score.

7. **Refinement:** Refine the recommendation system based on the evaluation results. This can involve adding new features or adjusting the similarity score calculation.

8. **Deployment:** Deploy the recommendation system in a user-friendly interface such as a website or mobile app.

6.3 Results

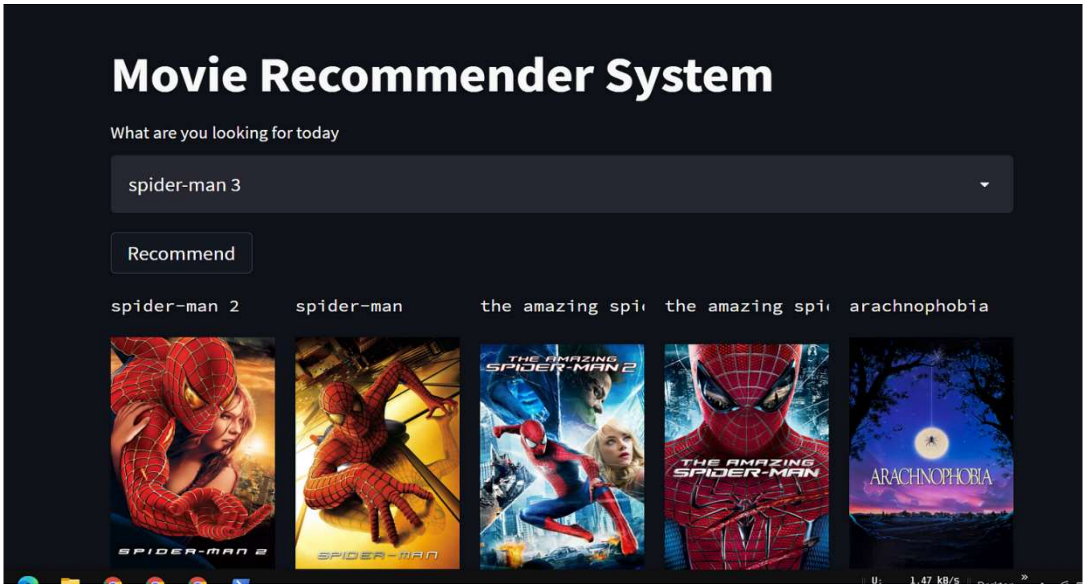


Movie Recommender System

How would you like to be contacted?

Spider-Man

Fig.5 Search bar to for movies recommend



Movie Recommender System

What are you looking for today

spider-man 3

```
spider-man 2    spider-man    the amazing spi    the amazing spi    arachnophobia
```

Fig.6 similar movies are displayed as result

Chapter 7

CONCLUSION

The internship aims to use Python programming language for Machine Learning so as to apply the theoretical knowledge to solve real-time and complex problems. The internship help integrate corporate experience in college life. The internship helped to find appropriate prediction model to the problems by applying suitable learning algorithm which can be used in future. The internship project assigned by the company helped to improve programming skills and to implement basic knowledge for solving real world problem like movie recommendation system which helped to use appropriate Machine Learning algorithm such as Multiple Linear Regression for building a model that can be used to predict the monetary value of a house in Boston area by using Python