

Ministry of Science and Higher Education of the Republic of Kazakhstan L.N. Gumilyov Eurasian National University

Faculty of Information Technology Department of Information Systems

COURSEWORK ON THE SUBJECT

"Mathematical Foundations of Intelligent Systems"

For third-year students of the specialty 6B06103 – Information Systems

Topic: Building a recommendation system for e-commerce based on user behavior data

- [Introduction](#)
- [Movie Recommendation System](#)
- [Dataset](#)
- [Model Implementation](#)
- [Machine Learning Model](#)
- [System development](#)
- [Interface](#)
- [Conclusion](#)
- [Movie Recommender System](#)

Completed by: Student of group IS-33, Zhumabek A.R

Coursework Supervisor: Prof, Zhukabayeva T. K.

Introduction

What is a Recommendation System?

A recommender system is an algorithmic tool used to provide users with items of interest such as products, music and other resources. The main task of such systems is to analyse user behaviour and, on this basis, provide products or content that best suit the interests of each particular user. There are three main approaches to recommender systems :

Collaborative filtering, which is based on the similarity of users' tastes.

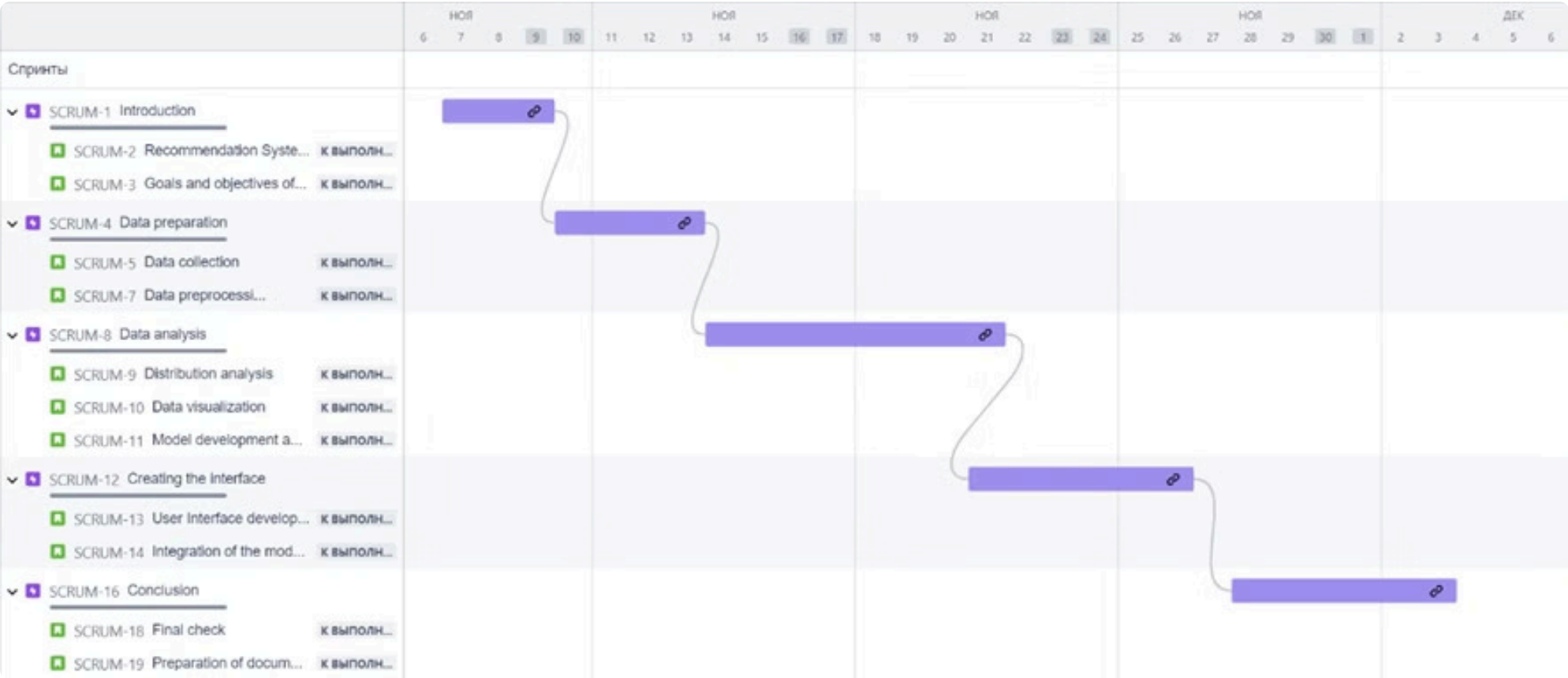
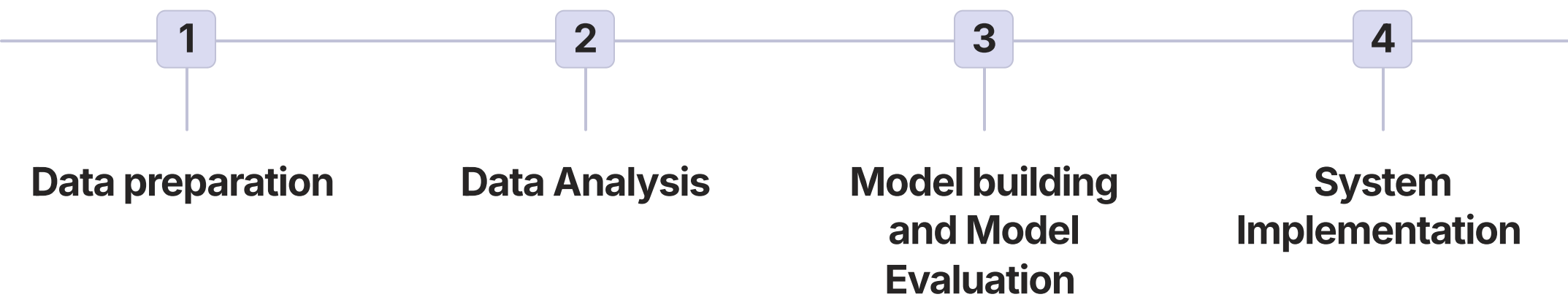
Content filtering, which uses characteristics of films such as genres or actors.

Hybrid models that combine the two approaches for greater accuracy.

Modern technologies, including machine learning and neural networks, help account for even hidden preferences and temporal changes in interests. For example, Netflix uses hybrid models to make recommendations more accurate and take seasonal interests into account.

Movie Recommendation System

To build the film recommendation system, the project was divided into several key steps to ensure the efficiency of data processing, model building, and application building.



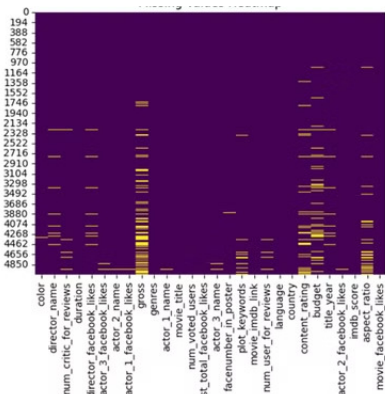
Dataset

In the course project, the basis for building the recommendation system was a dataset from the Kaggle platform containing information on more than 5,000 films. This stage includes data preparation and processing, which is a key factor in the success of the model. The main columns of the selected [dataset](#) are:

1. *Film title*: used to display recommendations.
2. *Genres*: allow user preferences to be taken into account.
3. *IMDb rating*: an important indicator of a film's popularity.
4. *Directors and actors*: influence viewers' choices.
5. *Budget and box office receipts*: reflect the financial success of films.
6. *Duration and year of release*: help to consider time preferences.

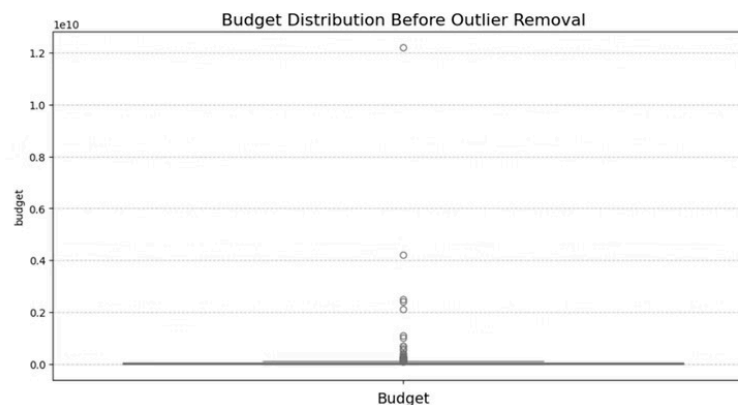
1

Data cleaning



2

Data transformation

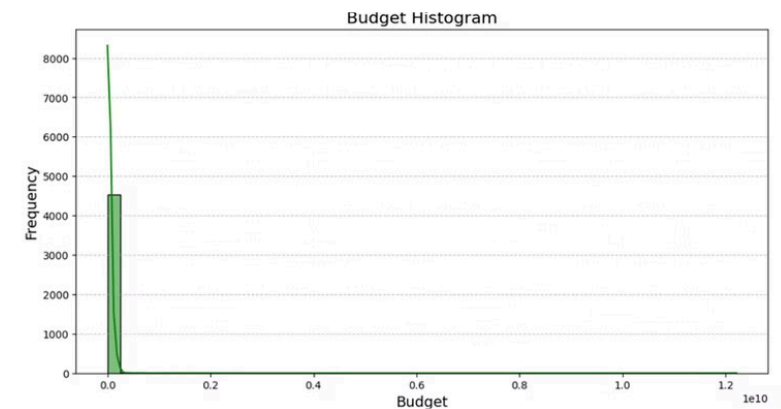


3

Data partitioning

4

Data saving



Model Implementation

The designed recommender system is based on three key methods:

Collaborative filtering

Content-based filtering

Random Forest & KNN

To implement the model, the dataset used is movie_metadata.csv, which is pre-processed: gaps are removed, new features are created, and the data is split into training and test samples. An example of data processing code:

```
import pandas as pd
import numpy as np

data = pd.read_csv('movie_metadata.csv')

data['success'] = np.where(data['gross'] > data['budget'], 1, 0)

data = data[['budget', 'duration', 'gross', 'title_year', 'genres', 'success']]

data = pd.get_dummies(data, columns=['genres'], drop_first=True)

data.dropna(inplace=True)

X = data.drop(columns=['success'])

y = data['success']
```

Machine Learning Model

Random forest is an ensemble method, meaning it combines multiple decision trees to improve prediction accuracy. Random Forest can be used to predict various outcomes such as box office success, user ratings, or overall popularity of a film. Random Forest model:

```
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = RandomForestClassifier(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

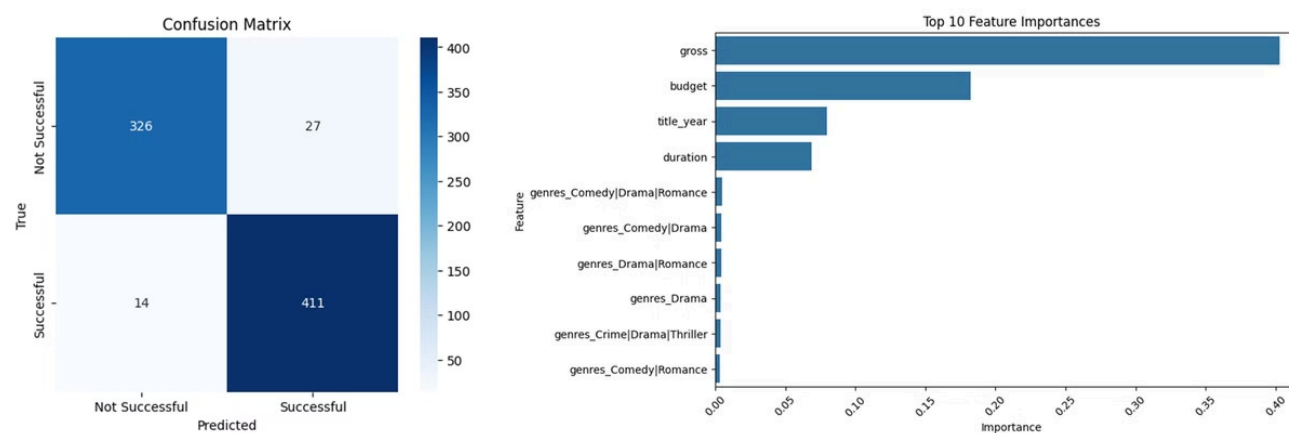
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy:.4f}')

print(confusion_matrix(y_test, y_pred))

print(classification_report(y_test, y_pred))
```



System development

The system used a JSON data format containing information about films: genres, ratings, posters and descriptions. The data is connected to the system as follows:

```
import json

with open('movie_data.json', 'r+', encoding='utf-8') as f:

    data = json.load(f)

    with open('movie_titles.json', 'r+', encoding='utf-8') as f:

        movie_titles = json.load(f)
```

The main stages of implementation:

Data preparation: calculation of the Euclidean distance between the characteristics of films.

Types of recommendations:

- By selected movie: search for similar ones by genre and rating.
- By genre: filters are user configurable

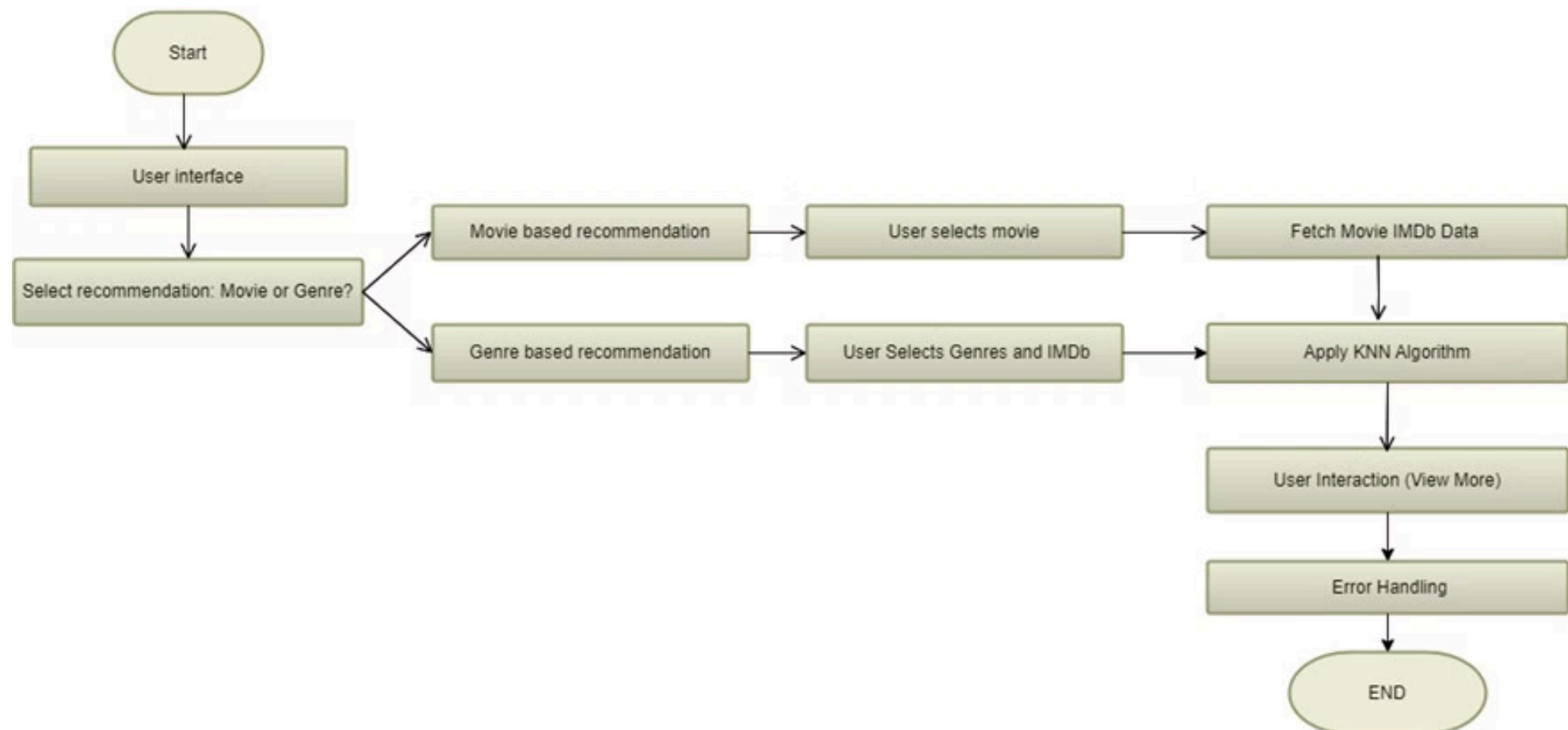
Integration with IMDb: Extracting information (posters, descriptions, directors, actors) using BeautifulSoup.

Interface

The system is designed using the Streamlit library. The main functions include:

- Drop-down lists for selecting movies/genres.
- Sliders for setting parameters (rating, number of movies).
- Dynamic display of results with posters, descriptions and links to IMDb.


Here we can see the flowchart of the system.




Conclusion


In this project, was developed a movie recommendation system using the K-nearest neighbors (KNN) algorithm and data analysis. The main goal was to create a simple yet effective tool that helps users find movies they're likely to enjoy. The system uses data on movie genres and ratings, along with analysis to understand the relationships between different movie features. It offers two ways to get recommendations: based on a selected movie or by genre. The results showed that the KNN algorithm and data analysis make it possible to generate accurate and personalized movie suggestions. Looking ahead, the system can be enhanced by integrating more advanced algorithms, like random forest, and incorporating additional data sources, such as user reviews. Overall, this project highlighted the value of machine learning in building practical applications and opened up opportunities for further development of the system.

akbayan5555/
COURSE




 1


Contributor

 0


Issues


 0


Stars

 0

Forks



 GitHub



GitHub - akbayan5555/COURSE

Contribute to akbayan5555/COURSE development by creating an account on GitHub.

Movie Recommender System

