Final Report

Group 3 - TED Talks

Members:
Hou Qinhan
Wang Yibo
Wang Zhihang
Sui Mingben

1. Initialization Step

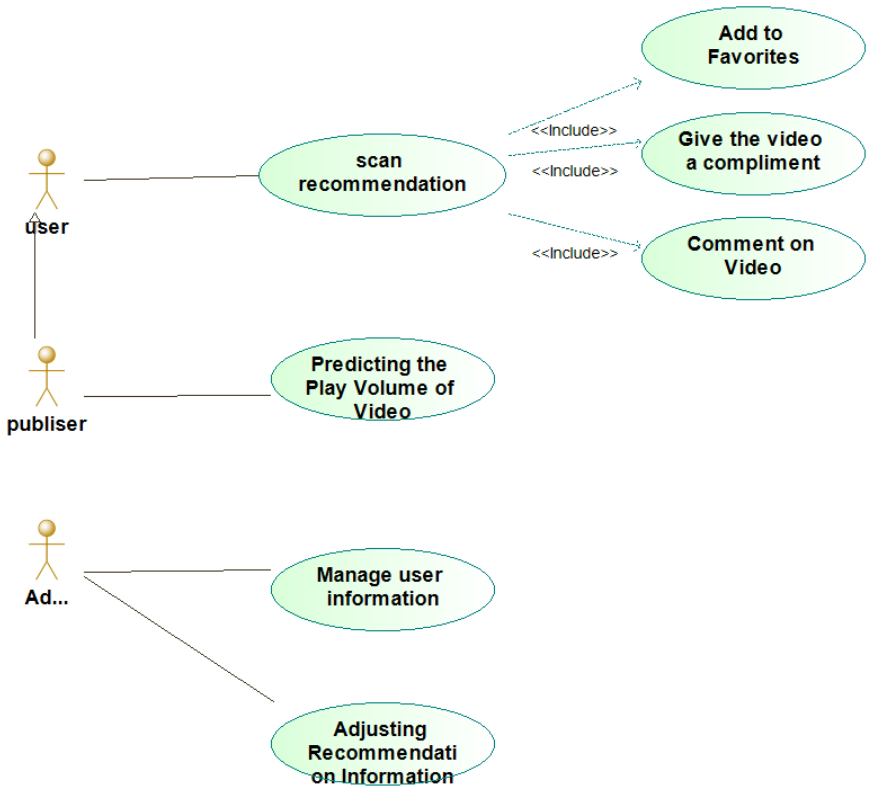
1.1 Background

Our team had selected a data set containing the play volume, volume of comments, tags, and comments of TED presentations in recent years, and conducted data analysis and machine learning.

There are many dimensions of data in this dataset, so we do machine learning and data mining, and hope to find more connections.

We hope to analyze the TED talk volume, the amount of comments and the time period, then analyze the possible clicks and comments from the newly uploaded TED talks. Through machine learning and data mining for this dataset, we hope to find the topic of the TED talk, the relationship between the content and the amount of play, the amount of comments, and predict the analysis of the amount and amount of comments on the newly uploaded TED speech video. At the same time, we also hope that we can recommend videos to users based on the tags that users are interested in and the preferences of users to watch videos.

1.2 Use case Diagram



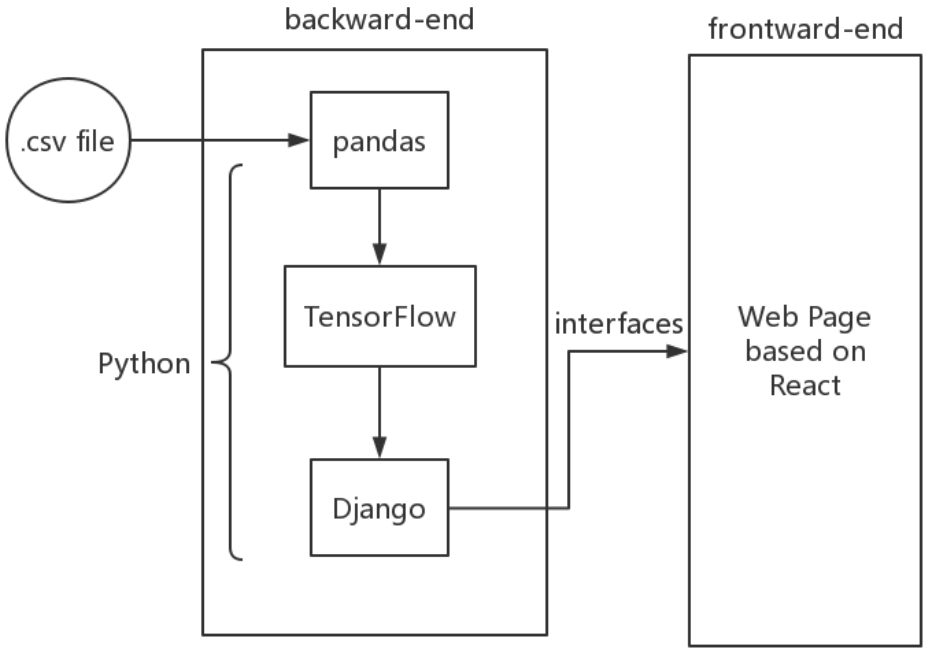
In the web application we developed, the user needs to log in first. After logging in, the user selects the tag of interest, and then enters the main interface to view the recommended video.

After watching the video, the user can choose whether to like this video, like this video, and can comment on the video. At the same time, users can also modify the tags they are interested in.

As the publisher of the video, after the video is released and the relevant information is input, relevant prediction information of the video playback amount and the comment amount can be obtained.

The administrator of the web application can manage the corresponding user information, and can also adjust the information recommended to the user.

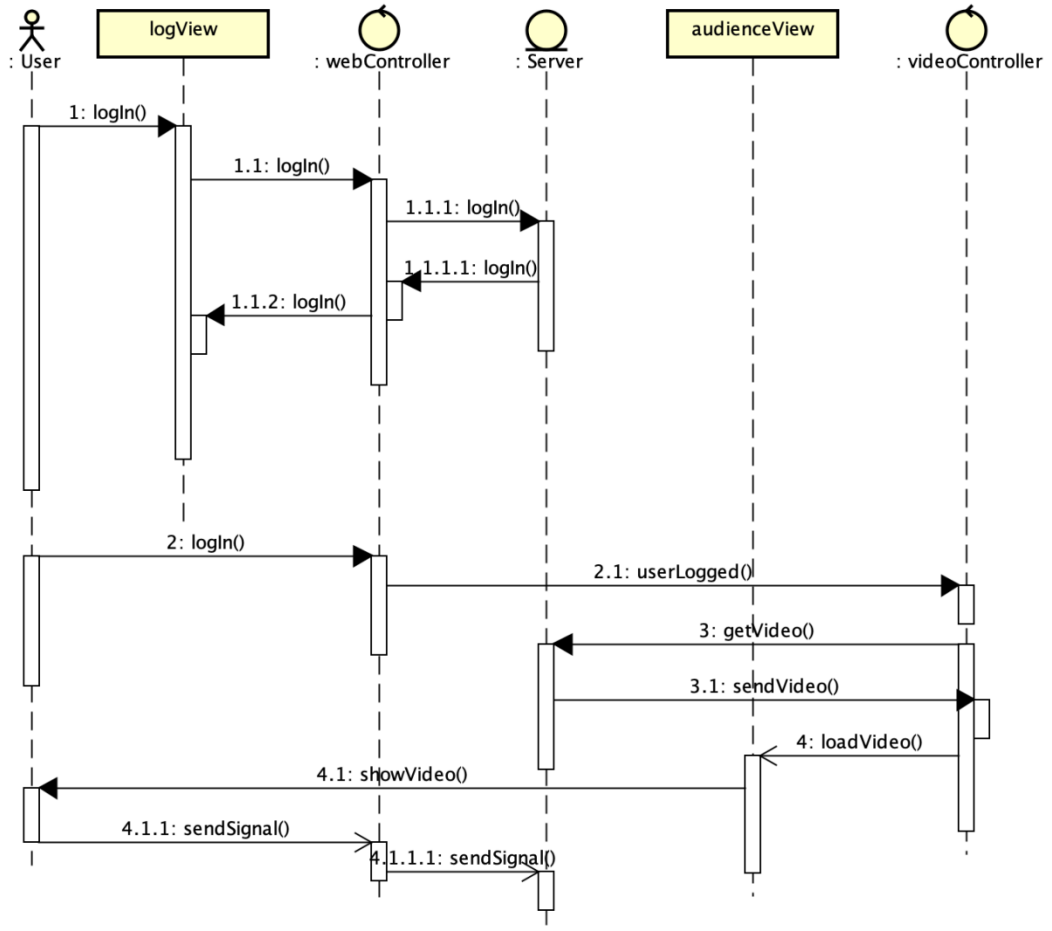
1.3 Global Architecture of the Project



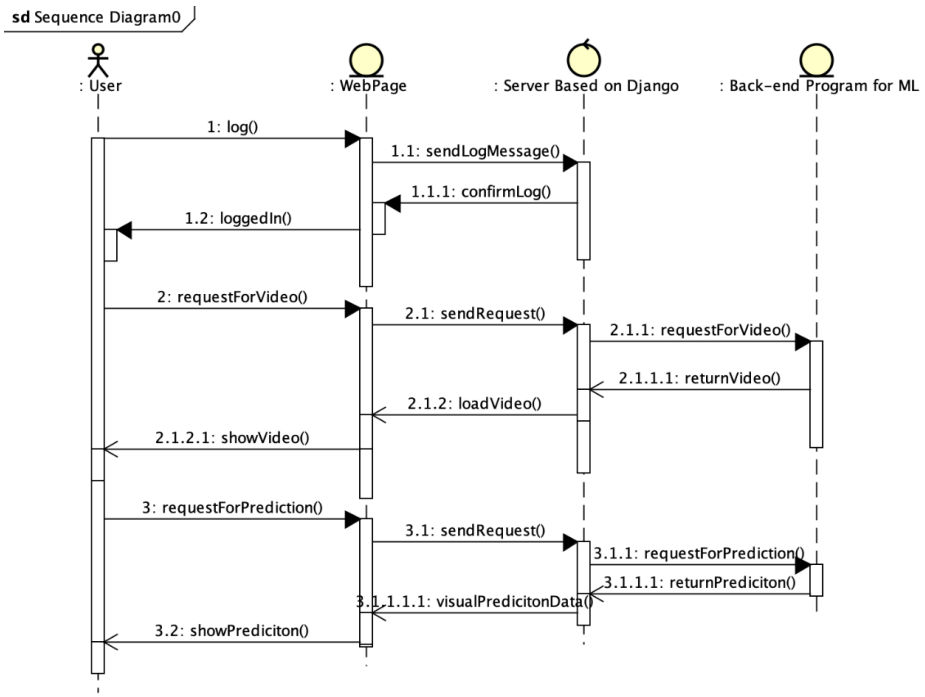
First of all, we use a Python data processing framework called Pandas to extract the data from .csv file to a specific data format for TensorFlow.

After getting the data, we use TensorFlow to train our model and get a prediction from our data.

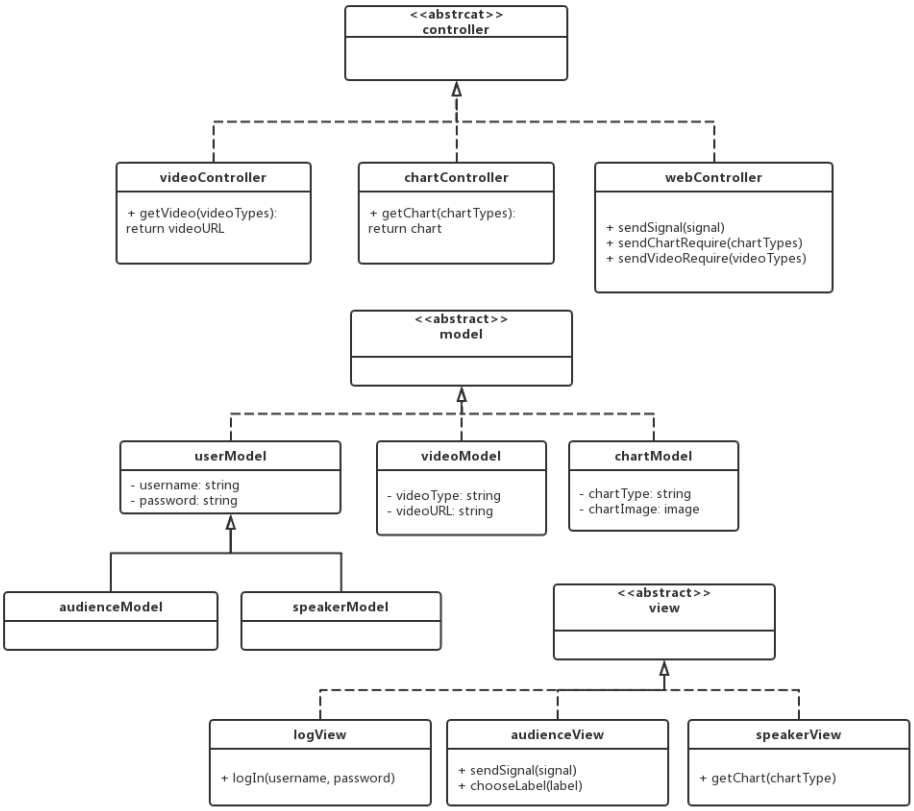
The result of TensorFlow will be organized by using Django and then will be sent to the web page by prescribed interfaces.



This timing diagram shows the dynamic collaboration between multiple objects by describing the chronological order in which messages are sent between objects. It can represent the order of behavior of the use cases. When a use case behavior is executed, each of the messages corresponds to a class operation or a trigger event in the state machine that causes the conversion.



This diagram shows a high-level invocation in all the project. There are three parts of our project, including front-end web page, back-end server based on Django and Python program processing all the data.



The overall design follows the MVC design pattern.

We abstract users, videos, and charts into models to facilitate the exchange and delivery of data.

The front end and back end will request and pass data through their respective controllers.

2. Elaboration Step

2.1 Scraping and collect the data

We can download the data set we need directly from kaggle's official website. (https://www.kaggle.com/rounakbanik/ted-talks). The main dataset contains metadata about every TED Talk hosted on the TED.com website until September 21, 2017. And this data set is a matrix of 2550*17.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	comments	description	duration	event	film_date	languages	main_spea	name	num_speal	published_	ratings	related_tall	speaker_oc	tags	title	url	views
2	4553	Sir Ken Rol	1164	TED2006	1.14E+09	60	Ken Robin	Ken Robins	1	1.15E+09	[{'id': 7, 'na	[{'id': 865,	Author/edi	['children',	Do school	https://ww	47227110
3	265	With the sa	977	TED2006	1.14E+09	43	Al Gore	Al Gore: A	1	1.15E+09	[{'id': 7, 'na	[{'id': 243,	Climate ad	['alternativ	Averting th	https://ww	3200520
4	124	New York T	1286	TED2006	1.14E+09	26	David Pog	David Pogi	1	1.15E+09	[{'id': 7, 'na	[{'id': 1725	Technolog	['computer	Simplicity s	https://ww	1636292
5	200	In an emot	1116	TED2006	1.14E+09	35	Majora Car	Majora Car	1	1.15E+09	[{'id': 3, 'na	[{'id': 1041	Activist for	['MacArthu	Greening t	https://ww	1697550
6	593	You've nev	1190	TED2006	1.14E+09	48	Hans Rosli	Hans Rosli	1	1.15E+09	[{'id': 9, 'na	[{'id': 2056	Global hea	['Africa', 'A	The best st	https://ww	12005869
7	672	Tony Robb	1305	TED2006	1.14E+09	36	Tony Robb	Tony Robb	1	1.15E+09	[{'id': 7, 'na	[{'id': 229,	Life coach;	['business',	Why we do	https://ww	20685401
8	919	When two	992	TED2006	1.14E+09	31	Julia Sweer	Julia Sweer	1	1.15E+09	[{'id': 3, 'na	[{'id': 22, 'n	Actor, com	['Christiani	Letting go	https://ww	3769987
9	46	Architect Ja	1198	TED2006	1.14E+09	19	Joshua Prir	Joshua Prir	1	1.15E+09	[{'id': 9, 'na	[{'id': 750,	Architect	['architectu	Behind the	https://ww	967741
10	852	Philosophe	1485	TED2006	1.14E+09	32	Dan Denne	Dan Denne	1	1.15E+09	[{'id': 3, 'na	[{'id': 71, 'n	Philosophe	['God', 'TE	Let's teach	https://ww	2567958

2.1.1 Data cleaning and transformation

First we use dropna() to discard the uncomplete rows.

```
pd.set_option('display.max_columns', None)
data=pd.read_csv("C:/Users/w/a1/ted_main.csv")
df = data.dropna() #clean
```

Remove the useless columns and reorder the columns

2.2 Analysis of the dataset

2.2.1 Overall analysis

```
print(df.head(5))
print(df.describe())
```

Several useful columns such as title main speaker, views, comments, event are retained.

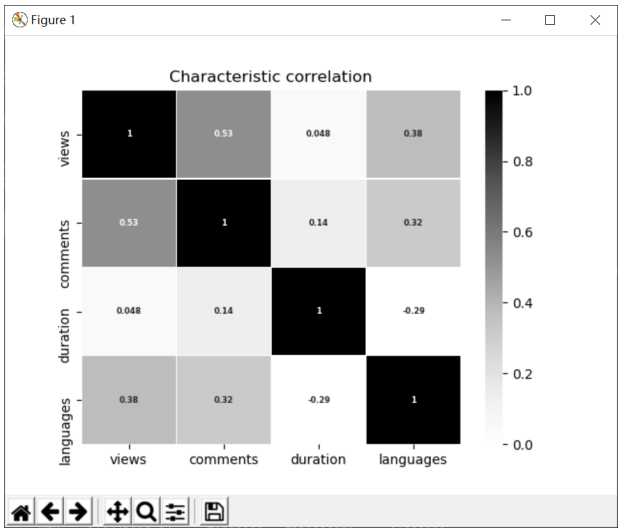
title	main_speaker	views	comments	event	duration	film_date	published_date	languages
Do schools kill creativity?	Ken Robinson	47227110	4553	TED2006	1164	25-02-2006	27-06-2006	60
Averting the climate crisis	Al Gore	3200520	265	TED2006	977	25-02-2006	27-06-2006	43
Simplicity sells	David Pogue	1636292	124	TED2006	1286	24-02-2006	27-06-2006	26
Greening the ghetto	Majora Carter	1697550	200	TED2006	1116	26-02-2006	27-06-2006	35
The best stats you've ever seen	Hans Rosling	12005869	593	TED2006	1190	22-02-2006	27-06-2006	48

We can see the overall situation of digital data such as views, comments, duration, languages, etc.

	views	comments	duration	languages
count	2.544000e+03	2544.000000	2544.000000	2544.000000
mean	1.699779e+06	191.706761	827.316431	27.319969
std	2.501043e+06	282.613719	373.828955	9.563529
min	5.044300e+04	2.000000	135.000000	0.000000
25%	7.565802e+05	63.000000	578.750000	23.000000
50%	1.123870e+06	118.000000	848.500000	28.000000
75%	1.702149e+06	222.000000	1047.000000	33.000000
max	4.722711e+07	6404.000000	5256.000000	72.000000

2.2.2 Analyze the correlation diagram between these data

```
correlations = df.corr()
# plot correlation matrix
fig = plt.figure()
ax = fig.add_subplot(figsize=(20,20)) #size 20*20
ax = sns.heatmap(correlations,cmap=plt.cm.Greys,
linewidths=0.05,vmax=1,
vmin=0 ,annot=True,annot_kws={'size':6,'weight':'bold'})
ax.set_title('Characteristic correlation')#set the title
plt.show()
```



无	−0.09 to 0.0	0.0 to 0.09
弱	−0.3 to −0.1	0.1 to 0.3
中	−0.5 to −0.3	0.3 to 0.5
强	−1.0 to −0.5	0.5 to 1.0

According to the Pearson product-moment correlation coefficient, Greater than 0.5 has a strong association, so comments and views have a strong association, the number of comments and the number of languages are moderately related, and the number of languages and pageviews are also moderately correlated.

2.2.3 Concrete analysis of view

```
a = df.sort_values("views",inplace=False,ascending=False)
a = a[['title', 'main_speaker', 'views',
'comments','published_date']]
print("Most viewed videos")
print(a.head())
```

These are the most viewed videos among so many videos.

```
Most viewed videos
                                  title  main_speaker     views  comments published_date
0                 Do schools kill creativity?  Ken Robinson  47227110      4553     27-06-2006
1346  Your body language may shape who you are     Amy Cuddy  43155405      2290     01-10-2012
677           How great leaders inspire action   Simon Sinek  34309432      1930     04-05-2010
837               The power of vulnerability   Brené Brown  31168150      1927     23-12-2010
452     10 things you didn't know about orgasm    Mary Roach  22270883       354     20-05-2009
```

The most viewed is the Do schools kill creativity? The pageview reached an astonishing 47.2billion, equivalent to a total population of Spain.

2.2.4 Analyze pageview digital features

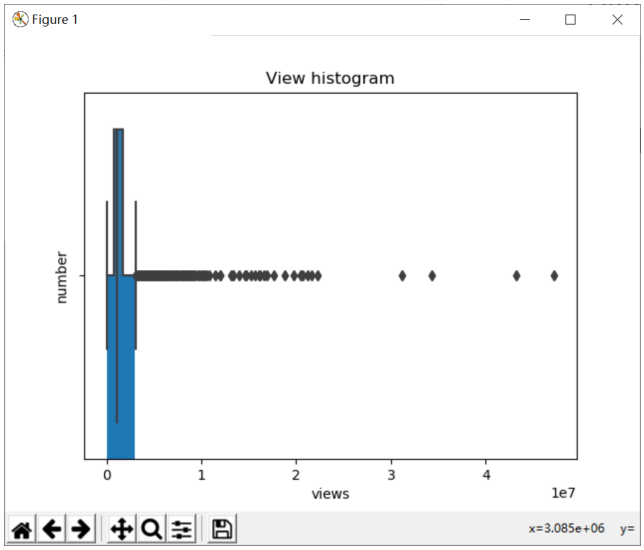
```
print(df['views'].describe())
```

```
count    2.544000e+03
mean     1.699779e+06
std      2.501043e+06
min      5.044300e+04
25%      7.565802e+05
50%      1.123870e+06
75%      1.702149e+06
max      4.722711e+07
Name: views, dtype: float64
```

The average number of views on the TED talks was 1.6 million. The median was 1.12 million. This shows that the average popularity of the TED talks is very high.

The boxplot of it

```
sns.boxplot(df['views'])
```

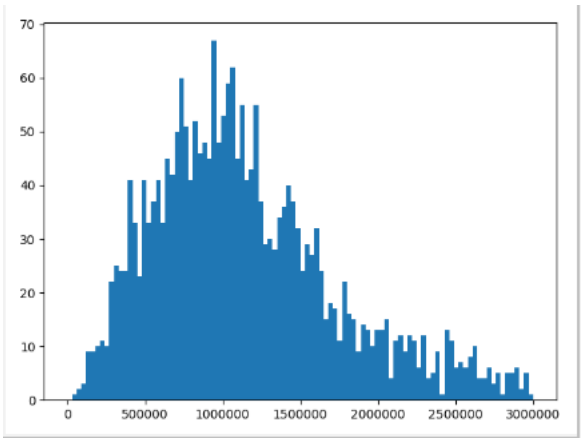


A large number of video playback is concentrated around 3008500.

```
plt.hist(df.views, range=(0,3000000), bins=100, rwidth=1)
plt.xlabel(u"views")# plots an axis lable
plt.ylabel(u"number")
plt.title(u"View histogram")
```

Hist：

```
plt.hist(df.views, range=(0,3000000), bins=100, rwidth=1)
plt.show()
```



The data distribution is a bit like a normal distribution

2.2.5 Concrete analysis of comments

```
a =
```

```
df.sort_values("comments",inplace=False,ascending=False)

a = a[['title', 'main_speaker',

'views','comments','published_date']

print("Most commented videos")

print(a.head(10))
```

Most commented videos

	title	main_speaker	views	comments	published_date
96	Militant atheism	Richard Dawkins	4374792	6404	16-04-2007
0	Do schools kill creativity?	Ken Robinson	47227110	4553	27-06-2006
644	Science can answer moral questions	Sam Harris	3433437	3356	22-03-2010
201	My stroke of insight	Jill Bolte Taylor	21190883	2877	12-03-2008
1787	How do you explain consciousness?	David Chalmers	2162764	2673	14-07-2014
954	Taking imagination seriously	Janet Echelman	1832930	2492	08-06-2011
840	On reading the Koran	Lesley Hazleton	1847256	2374	04-01-2011
1346	Your body language may shape who you are	Amy Cuddy	43155405	2290	01-10-2012
661	The danger of science denial	Michael Specter	1838628	2272	12-04-2010
677	How great leaders inspire action	Simon Sinek	34309432	1930	04-05-2010

The most commented is Militant atheism, not the most viewed Do schools kill creativity?

Analyze the numerical characteristics of the commentary:

```
print(df['comments'].describe())
```

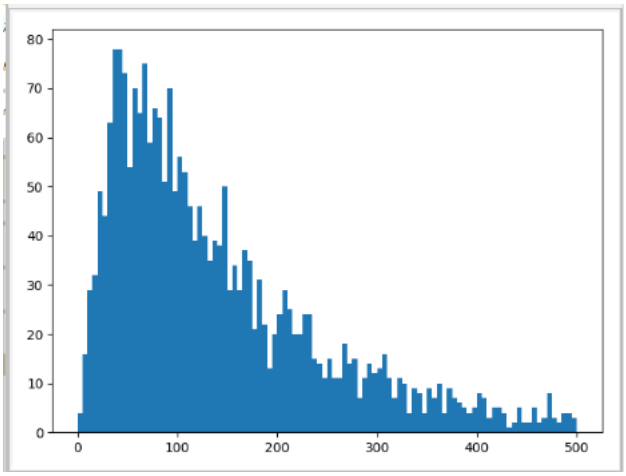
```
count    2544.000000
mean      191.706761
std       282.613719
min         2.000000
25%        63.000000
50%       118.000000
75%       222.000000
max      6404.000000
Name: comments, dtype: float64
```

Visualize with boxplot

```
sns.boxplot(df['comments'])
```

Hist：

```
plt.hist(df.views, range=(0,500), bins=100, rwidth=1)
```



Rising around 0-70, then falling steadily

2.3.5 The relationship between comments and page views.

```
a = sns.jointplot(x = 'views', y = 'comments', data = df)

plt.title("relationship between views and comments")



b = df[['views', 'comments']].corr()

print(b)
```

2.3.6 Analysis of the number of languages

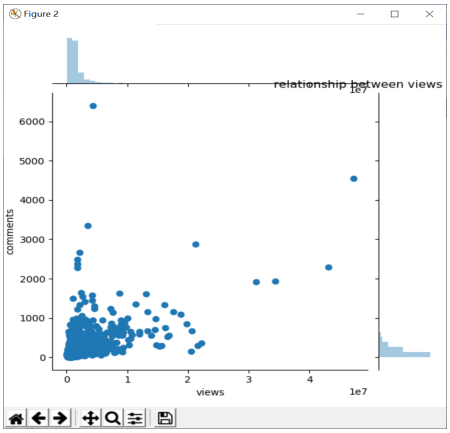
```
a = df['languages'].describe()

print("language discrible")

print(a)
```

```
language discrible
count    2544.000000
mean       27.319969
std         9.563529
min         0.000000
25%        23.000000
50%        28.000000
75%        33.000000
max        72.000000
```

A video can have up to 72 different languages, 75% in 33, and the median is 27.

2.3.7 Analyze the relationship between page views and the number of languages

```
a = sns.jointplot(x = 'views', y = 'languages', data = df)
b = df[['views', 'languages']].corr()
print("correlation between views and languages")
print(b)
```




```
correlation between views and languages
                views  languages
views        1.000000   0.378027
languages    0.378027   1.000000
```

Weak connection.

2.3.8 Analysis of duration data below

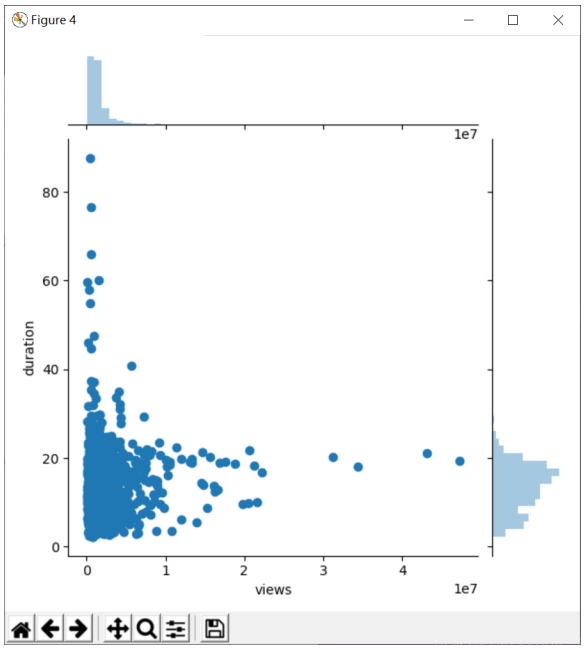
```
df['duration'] = df['duration']/60
a = df['duration'].describe()
```

```
print("duration describe")
print(a)
```

```
duration describe
count    2544.000000
mean       13.788607
std         6.230483
min         2.250000
25%         9.645833
50%        14.141667
75%        17.450000
max        87.600000
Name: duration, dtype: float64
```

The duration is in minutes. The longest one is ted video 87.6 minutes, the shortest is only 2.25 minutes, and 75% is at 17.45, so most of them are mainly short and medium video.

```
a = sns.jointplot(x = 'views', y = 'duration', data = df)
b = df[['views', 'duration']].corr()
print("correlation between views and duration")
print(b)
```



```
correlation between views and duration
              views  duration
views      1.000000  0.048489
duration   0.048489  1.000000
```

2.3.9 Analyze which speakers are the most talked

```
speaker_df =
df.groupby('main_speaker').count().reset_index()[['main_sp
eaker', 'comments']]
speaker_df.columns = ['main_speaker', 'count']
speaker_df = speaker_df.sort_values('count',
ascending=False)
showdata = speaker_df.head(10)
print("speakers who publised most videos")
print(speaker_df.head(10))
```

```
              main_speaker  count
767            Hans Rosling      9
1063         Juan Enriquez      7
1275         Marco Tempest      6
1689                 Rives      6
422            Dan Ariely      5
395           Clay Shirky      5
1484   Nicholas Negroponte      5
248             Bill Gates      5
1072       Julian Treasure      5
847   Jacqueline Novogratz      5
```

It can be seen that the most is Hans Rosling, the number of times is 9.

2.3.10 Analyze the occasion of its release

Each ted video has its own source, let's analyze the occasion of its release.

```
events_df = df[['title',
'event']].groupby('event').count().reset_index()
events_df.columns = ['event', 'talks']
events_df = events_df.sort_values('talks', ascending=False)
s = events_df.head(10)
print("ted events describtion")
print(s)
```

event	talks
TED2014	84
TED2009	83
TED2013	77
TED2016	77
TED2015	75
TEDGlobal 2012	70
TED2011	70
TED2010	68
TED2007	68
TED2017	67

The most is Ted2014, which released 84 speeches.

2.3.11 concrete analysis of tags

Finally, the analysis of the label Because each video corresponds to multiple tags, it needs to be saved with a list, then split and counted.

```
# transform string to list
df2['tags'] = df2['tags'].apply(lambda x: ast.literal_eval(x))
# divide the tags
```

```
s = df2.apply(lambda x:
pd.Series(x['tags']),axis=1).stack().reset_index(level=1,
drop=True)
s.name = 'theme'
# add the tags into the dataframe
theme_df = df2.drop('tags', axis = 1).join(s)
print("the number of tag ：
{}".format(len(theme_df['theme'].value_counts())))
#most popular tags
```

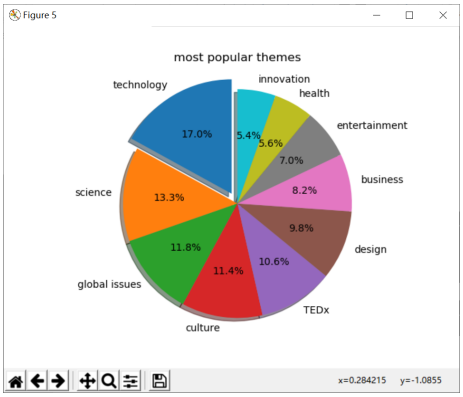
Count the total number of tags and find the 10 most popular tags

```
pop_themes =
pd.DataFrame(theme_df['theme'].value_counts()).reset_ind
ex()
pop_themes.columns = ['theme', 'talks']
print("most popular themes")
print(pop_themes.head(10))
```

Visualize with pie map.

```
#pie diagram
labels = pop_themes.head(10)['theme']
sizes = pop_themes.head(10)['talks']
```

```
explode = (0.1, 0, 0, 0,0,0,0,0,0,0)    # only "explode" the 2nd
slice (i.e. 'Hogs')


fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels,
autopct='%1.1f%%',
                shadow=True, startangle=90)
ax1.axis('equal')    # Equal aspect ratio ensures that pie is
drawn
as a circle.
plt.title("most popular themes")
plt.show()
```



Technology, science, global issues are the most popular themes.

3. Construction Step

3.1 Introduction

Based on the above data analysis, we will further machine learning the data to achieve the purpose of exploring deeper laws and practical applications. Machine learning is divided into supervised learning, unsupervised learning and intensive learning. This project only involves supervised learning and unsupervised learning.

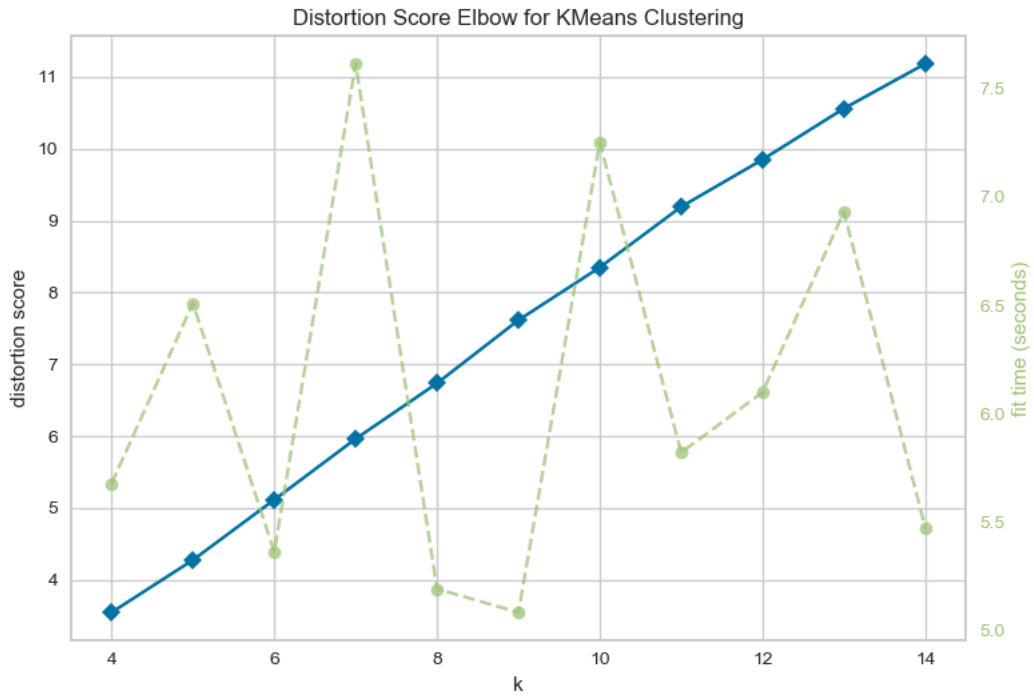
In unsupervised learning, we use KMeans, which is a clustering algorithm, for cluster analysis. In supervised learning, we use a variety of typical classification algorithms to classify content and analyze the pros and cons between them. In addition, we Linear regression, ridge regression and random forest algorithms are used to predict the data.

3.2 UNSUPERVISED LEARNING – CLUSTERING

3.2.1 Find the Most Suitable K Value

In order to cluster using the KMeans method, we need to know a suitable K value in advance to decide how many clusters to divide the data into. We use the Elbow Method to calculate the K values in the range 4 - 15 to get the Distortion Score and Fit Time for each K

value. For the same K value, we need the Distortion Score to be as small as possible and the Fit Time to be as low as possible, so we can choose 9 as the number of clusters.



3.2.2 Do the cluster

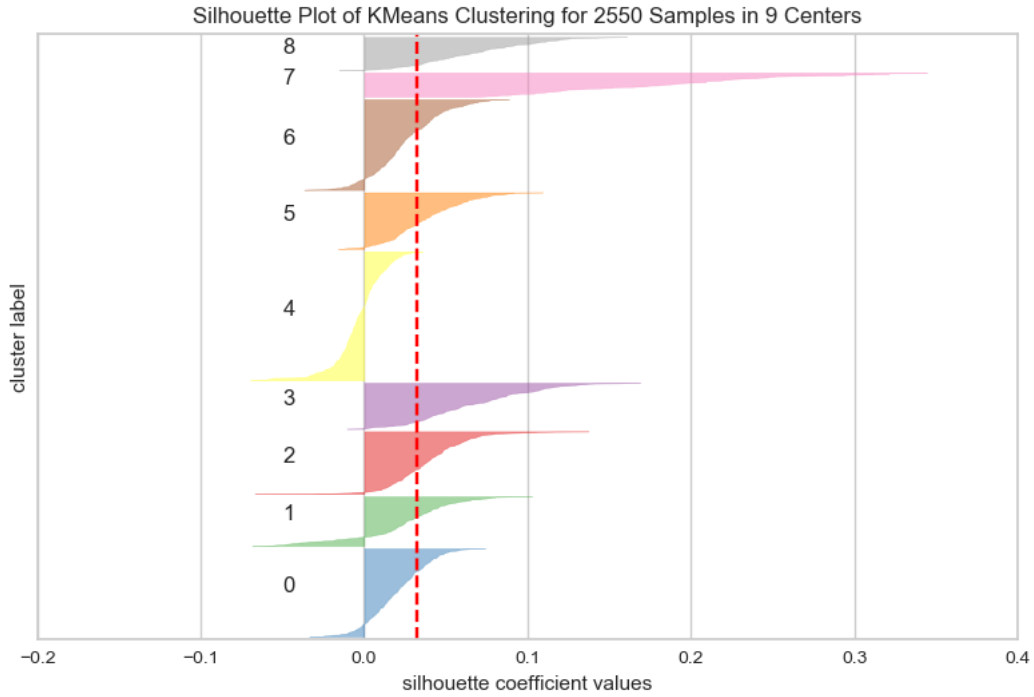
We pass the TF-IDF matrix as a training basis to the KMeans method, and thus get a model. We store the trained models so that we can call the model directly instead of spending time again.

Code like this:

```
tfidf = TfidfVectorizer(max_df=0.8, stop_words='english')
tfidf_matrix = tfidf.fit_transform(replaceList)
km = KMeans(n_clusters=7).fit(tfidf_matrix)

joblib.dump(km, modelURL)
```

3.2.3 Calculate the Sihouette Value

Sihouette Value is a factor used to detect how well KMeans are classified. We can see through visualization that among the data items classified into 7 categories, the data classified into the 7th item is more accurate, and the value of the 4th type data has a larger negative number, and the result is not good. Overall, the clustering results are acceptable.



3.3 SUPERVISED LEARNING - CLASSIFICATION

3.3.1 Find out input and output

We use the result of clustering in the previous step as a new parameter called "topic category", which represents the topic category to which the video belongs. In addition, we divide the three attributes of comment quantity, view quantity and duration

according to the histogram, and change them from continuous variables to discrete variables, so that they can be used in classification. We decided to use the subject, comment volume, and duration as input to get a categorization of views.

3.3.2 Select classification algorithm

We used 500 of the more than 2,000 data as a test set. Through the training set, we used different algorithms to obtain the classification model results and scored the results. The results are as follows:

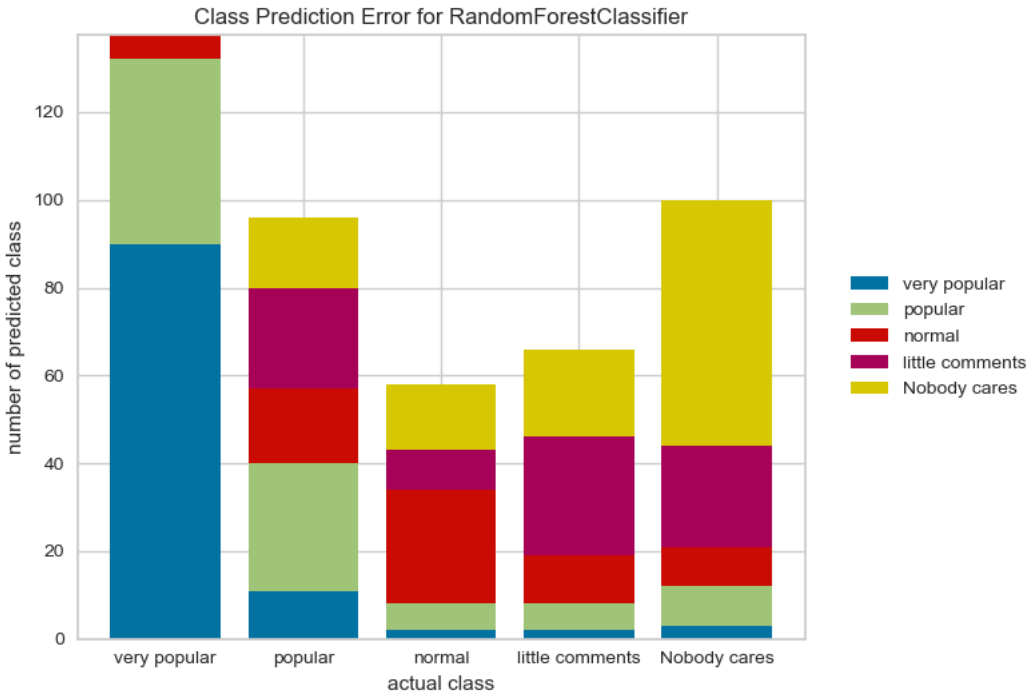
```
the classifier is : svm
the score is : 0.436
the classifier is : decision_tree
the score is : 0.452
the classifier is : naive_gaussian
the score is : 0.4
the classifier is : naive_mul
the score is : 0.324
the classifier is : K_neighbor
the score is : 0.398
the classifier is : bagging_knn
the score is : 0.28
the classifier is : bagging_tree
the score is : 0.39
the classifier is : random_forest
the score is : 0.474
the classifier is : adaboost
the score is : 0.38
the classifier is : gradient_boost
the score is : 0.398
```

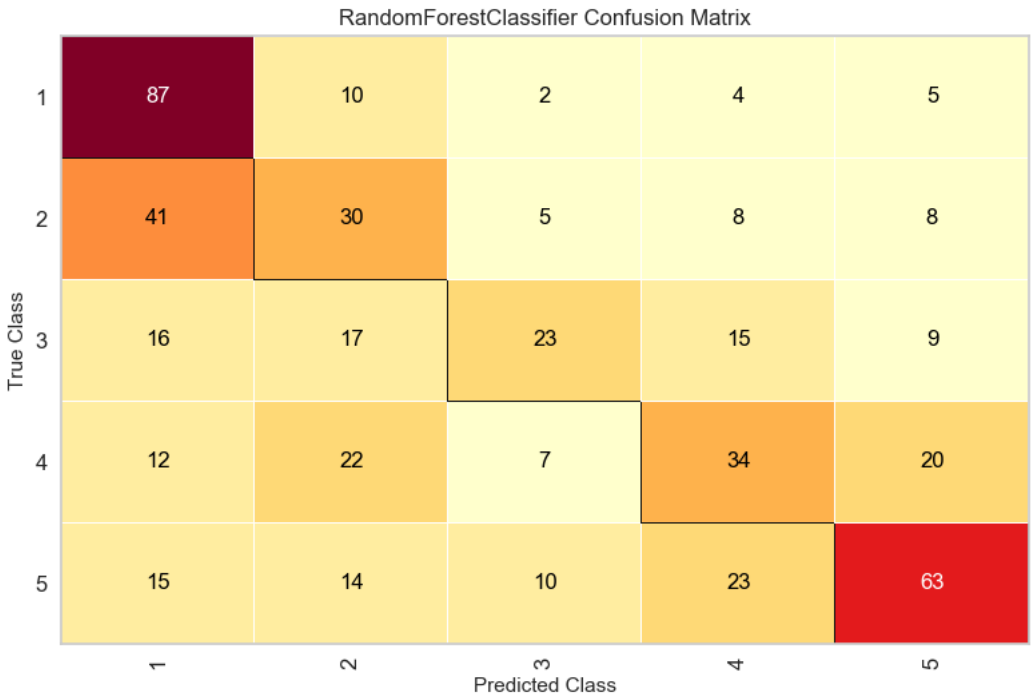
From the scoring we can see that the RandomForest score is the highest for the integration method, and the BaggingKnn algorithm, which is also the integration method, has the lowest score.

Therefore, our classification decided to use the RandomForest algorithm.

3.3.3 Test results

We use the RandomForest algorithm for classification, and then for the result test, we get the Prediction Error Figure and Confusion Matrix obtained by this algorithm classification.





As a result, it can be seen that the classification result is not perfect, but the classification task is basically completed, and the test set data can be roughly distinguished.

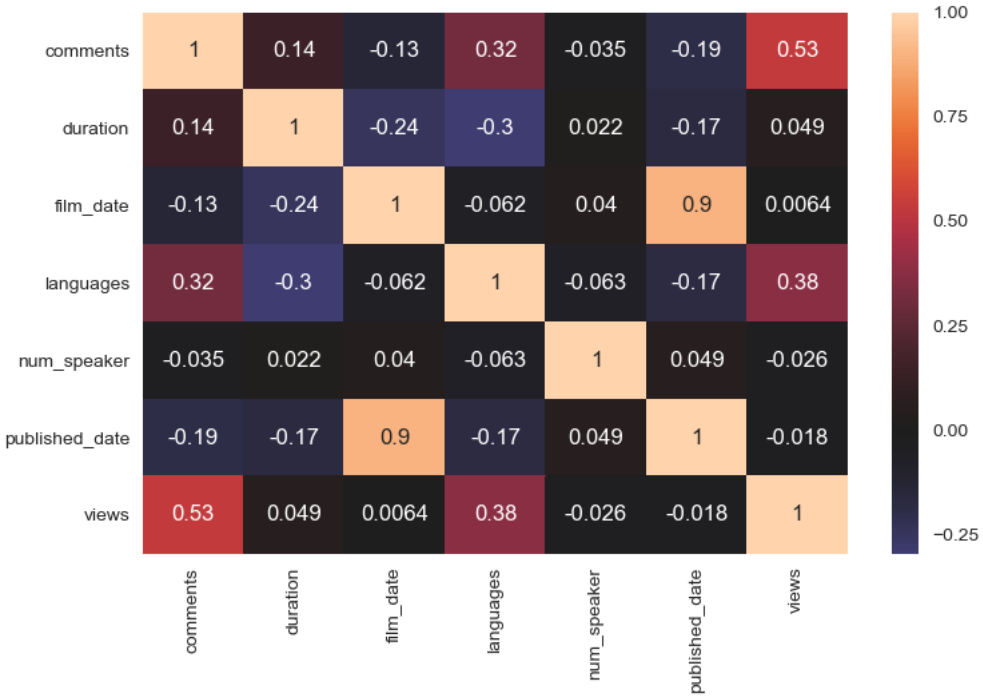
We suspect that the problem is not relevant. When we discussed Regression, we saw that there is not a strong correlation between the data in this dataset, so the prediction results are not as perfect as the datasets with strong correlation.

3.4 SUPERVISED LEARNING – REGRESSION

3.4.1 Looking for relevance

We used Heatmap to represent the relationship between the various numeric attributes.

The following results were obtained.



We can see that there is not much strong correlation between other attributes except for the obvious positive correlation (0.53) between views and comments. This also proves that we may not be able to produce perfect predictions, but we can try to go back. We ended up using duration, language and views to make regression predictions on comments.

3.4.2 Try different regression algorithms

In order to get relatively good regression results, we tried three algorithms: linear regression, ridge regression and random forest algorithm. Their results are as follows:

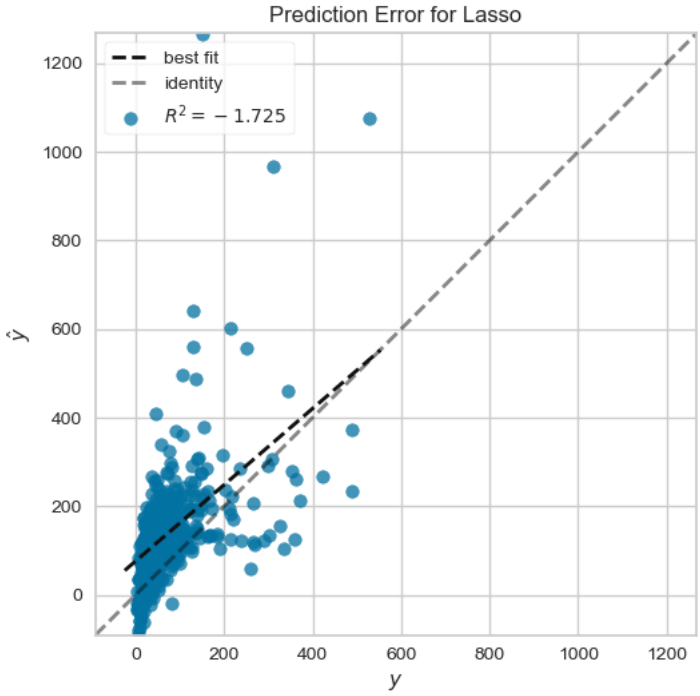
```
MSE: 14752.617541618034
MSE(Calculate MSE according to the formula): 14752.617541618047
RMSE: 121.46035378516743


MSE: 6954.128185800001
MSE(Calculate MSE according to the formula): 6954.128185799998
RMSE: 83.39141554021013


MSE: 14752.619583607015
MSE(Calculate MSE according to the formula): 14752.61958360701
RMSE: 121.46036219115689
```

We can see that the random forest algorithm in the middle obtains the smallest RMSE deviation, while the results of linear regression and ridge regression are not much different, and the results from the random forest algorithm have large errors. Therefore, we decided to use the random forest algorithm for regression prediction in the project.

However, the RMSE is as high as 83 compared to the average of 186.17, which is not a good result. So we consider trying other algorithmic analysis.

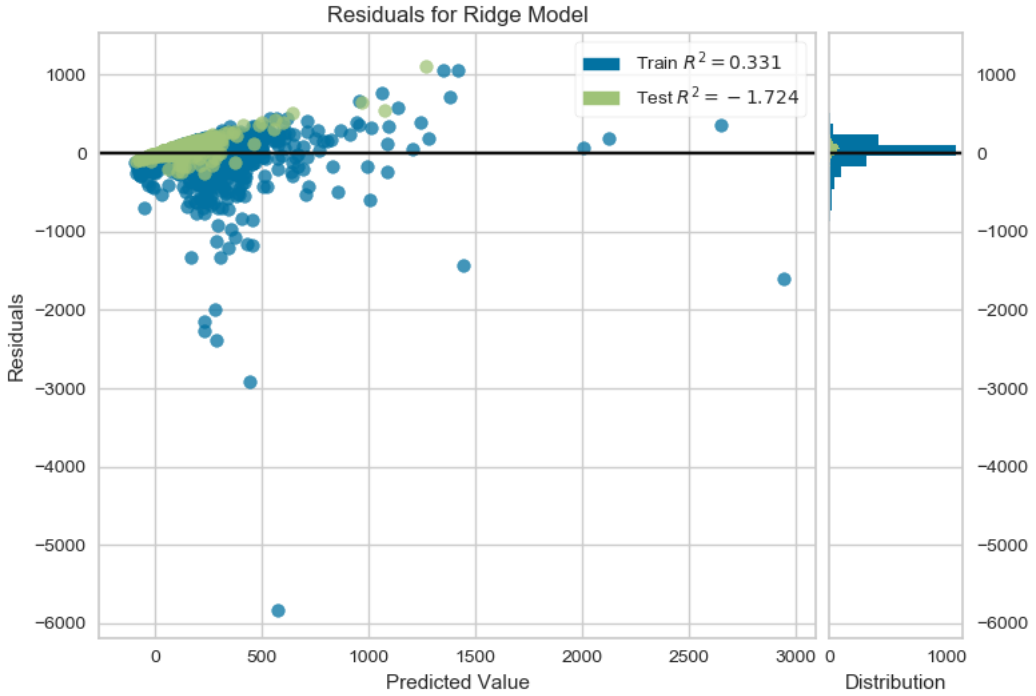
3.4.3 Deviation analysis

We use the visual component Yellowbricks for another way of error analysis. We chose the Lasso and Ridge methods for error analysis. Lasso is a biased estimate of sparse data that minimizes the value of MSE under sparse data.



We obtained the above image by regression analysis performed by the Lasso algorithm. Our data is almost concentrated near a straight line at a 45-degree angle, but the predicted values are a little larger than the true value. As you can see from the figure, the Lasso algorithm is more suitable for our sparse, irrelevant data, but because it is a biased estimate, we can't use it in real prediction.

We also used the Ridge algorithm for error analysis.



Our training data is a blue point, and the test data is a green point. This figure shows the residuals between the test data and the training data. We see that the residuals of most of the test data are clustered around 0, indicating that our training model is more realistic.

4. Application

4.1 Overall

4.1.1 Introduction

Because our goal is to provide user-defined task flow, we need a way of interaction. Here we choose B/S architecture. Users use browsers to interact with our services. So we choose the Django architecture.

4.1.2 Architecture

This is a variation of the MVC pattern as you can see in the acronym itself the Template keyword replaces the Controller. Although, the Template is not exactly functioning as the controller and has some different properties than the controller.
The definitions of Model still remain the same that is, the Model contains the logical file structure of the project and is the middleware & data handler between database and view. The Model provides a definition of how the data formats as coming from the view so, it stores in the database and vice-versa, i.e., the retrieving information from the database transfers to the view in the displayable format.
The View in MTV architecture can look like the controller, but it's not. The View in this MTV architecture is formatting the data via

the model. In turn, it communicates to the database and that data which transfer to the template for viewing.

4.1.3 Workflow:

1. Users request a page through a browser

2. The request arrives at Request Middlewares, and the middleware does some pre-processing or direct response requests to the request.

3. URLConf finds the corresponding View through the urls.py file and the requested URL

4. View Middlewares is accessed, and it can also do some processing of requests or return responses directly.

5. Call the functions in View

6. The method in View can selectively access the underlying data through Models

7. All Model-to-DB interactions are done through Manager

8. Views can use a special Context if needed

9.

A. Context is passed to Template to generate pages

Template uses Filters and Tags to render output

B. The output is returned to View

C. HTTPResponse is sent to Response Middlewares

D. Any Response Middleware can enrich response or return a completely different response

E.Response returns to the browser and presents to the user

4.1.4 Database

There is a need to save the user's information and movies' attributes. So constructing two tables to save them.

This application chooses SQLite 3 because SQLite is a very light weighted database and it is easy to use. Reading and writing operations are very fast for SQLite database. It is almost 35% faster than File system.

User has fundamental attributes such as name, password, nickname. Preference stands for the appetite of this user and isSuper means whether this user is a super user because this application has a backstage to adjust the data.

4.1.5 User log in & log out(Hard)

First of all, the login page and data submission are written in this method, but the request mode of the login page is GET, and the request mode of the login data submission is POST. So request. method is used to judge the request method. Different request methods have different processing codes. For the login page

request processing method is simple, just return to the corresponding template page, but in order to log back to the original request page after login, the parameter context ['previous_page'] = request. GET. get ('from_page') is used to record the URL of the request login page. Form form is used to submit login data. Bootstrap is used as the CSS framework in the form style, but there may be errors in login parameters, so JS pop-up window method is used to realize the error prompt. When the form data is submitted to the corresponding processing method, the corresponding user name and password are obtained by request. POST ['key']. Then the login is realized by user = authenticate (request, username = username, password = password) and auth. login (request, user).

4.1.6 Recommendation part

Term frequency–inverse document frequency(tf-idef) is a commonly used weighting technology for information retrieval and text mining. TF-IDF is a statistical method to evaluate the importance of a word to a document set or one of the documents in a corpus. The importance of words increases with the number of times they appear in documents, but decreases inversely with the frequency they appear in corpus. TF-IDF weighted forms are often

used by search engines as a measure or rating of the degree of correlation between files and user queries.

TfIdfVectorizer in sklearn library is directly used to calculate TF-IDF matrix.

Code like this:

```
    replaceList.append(input_tag)
      tfidf_matrix = tfidf.fit_transform(replaceList)
      cosine_sim = linear_kernel(tfidf_matrix, tfidf_matrix)
      sim_scores = list(enumerate(cosine_sim[-1]))
      sim_scores = sorted(sim_scores, key=lambda x: x[1],
reverse=True)
      a = random.randint(2,10)
      sim_scores = sim_scores[a:a+1]
      movie_indices = [i[0]for i in sim_scores]


      result = movies['url'].iloc[movie_indices].tolist()
      result2 = movies['tags'].iloc[movie_indices].tolist()
```

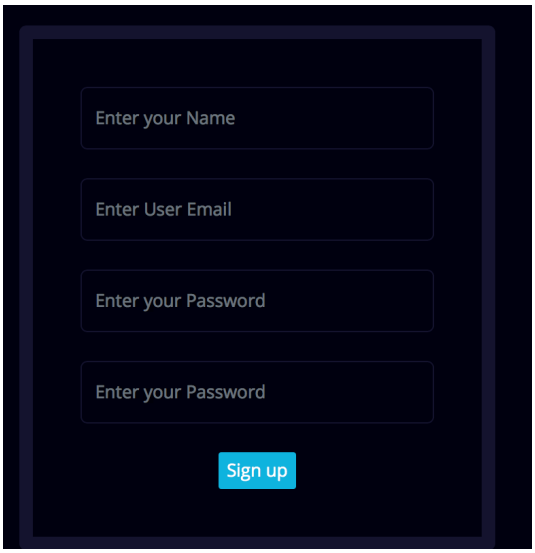
Cosine distance is used to calculate the correlation between the two films. The cosine distance can be obtained by calculating the dot product with linear_kernel(), and the similarity can be calculated according to the distance.

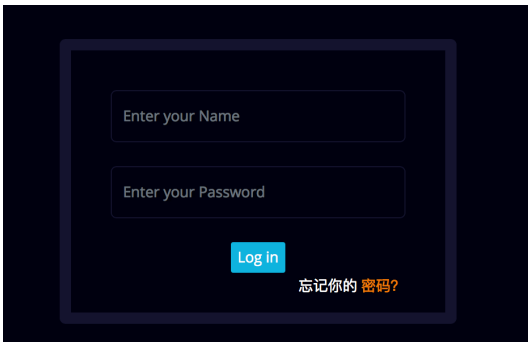
4.2 Features Introduction

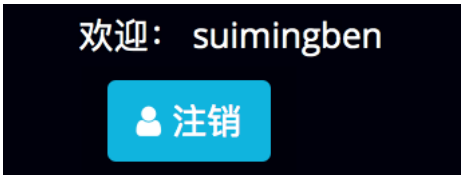
Users can select videos recommended by the web according to their own preferences. The user's personal web page can display attributes and other videos that the user likes the video. The data page shows machine-related pictures and text as well as detailed information about the most popular TED videos in the Statistics Office. Users can also enter the video's current, supported language, play volume, and date to predict the amount of comments. Our website displays our basic information and users can contact us according to the website guidelines. The website supports Chinese, English and French.

The language of our project uses HTML, CSS, JavaScript, and Python.

When the user enters the website, he can select the desired function according to the different drop-down menus of the navigation bar.

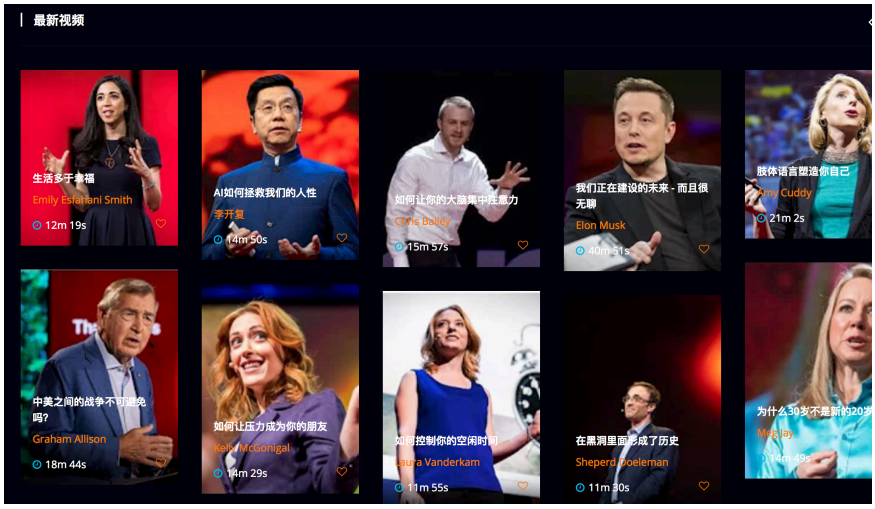






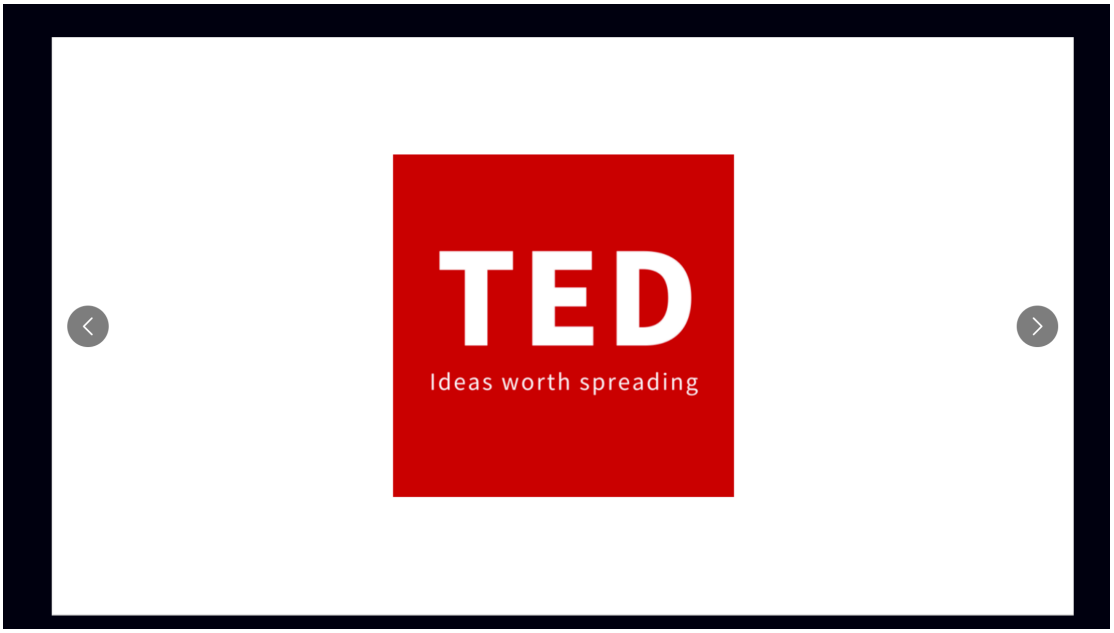
After the user logs in successfully, you can see the information shown in the upper right corner of the web page and you can choose to log out.

After logging in, users can choose TED videos that they like or dislike according to their preferences.

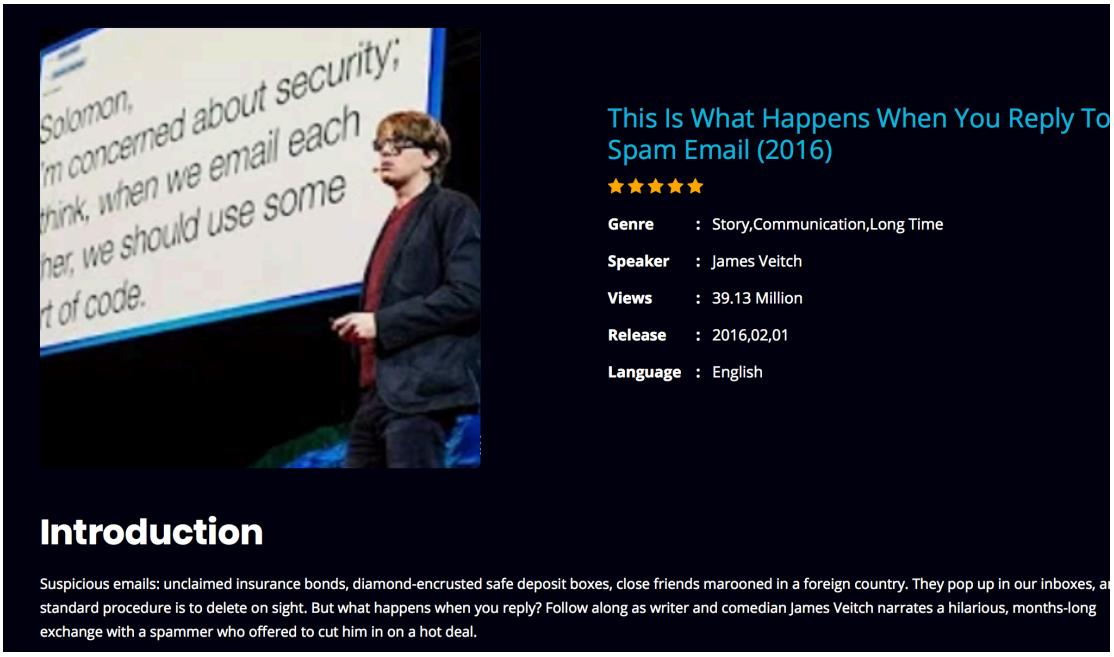


Some of the sites below the homepage display some of the latest and most popular videos that users can click to watch.

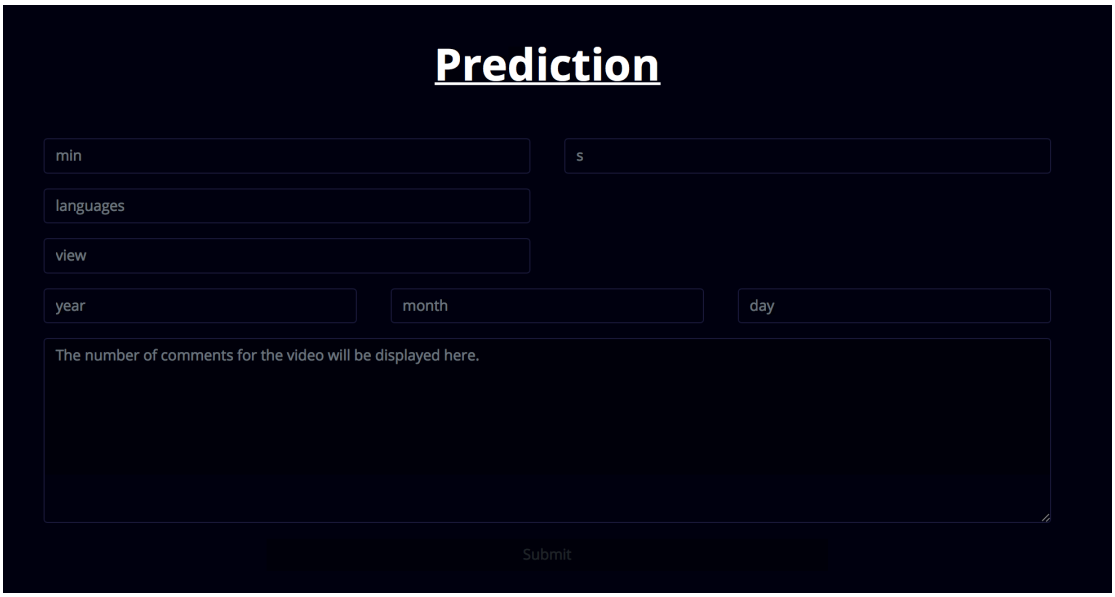
When the user enters the personal homepage, the website displays the personal information and the type of video the user likes, which makes it easier for the user to find the desired video again.



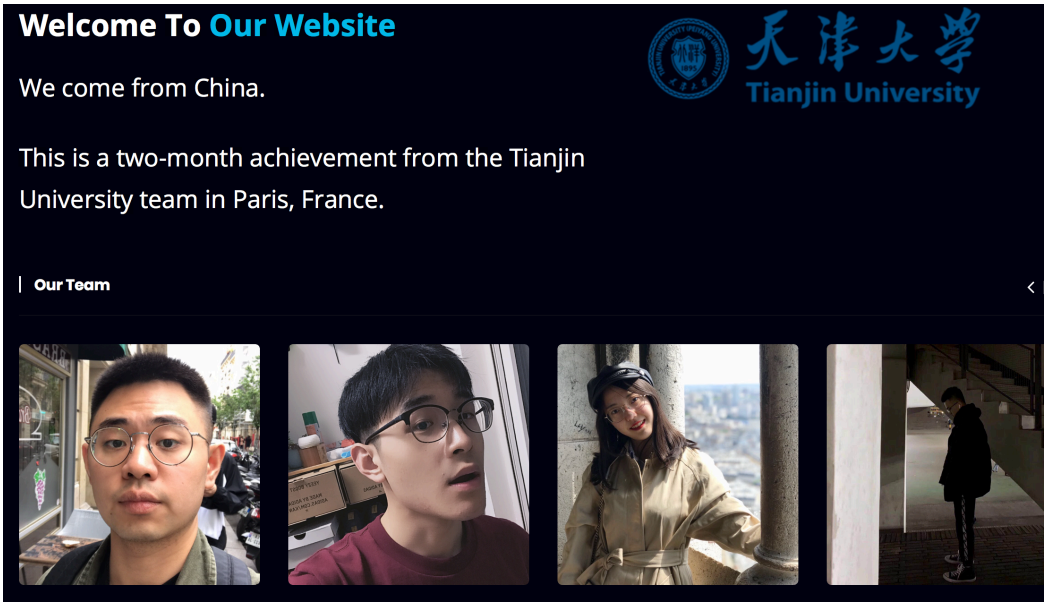
On the Data page, the website will display some data graphs.



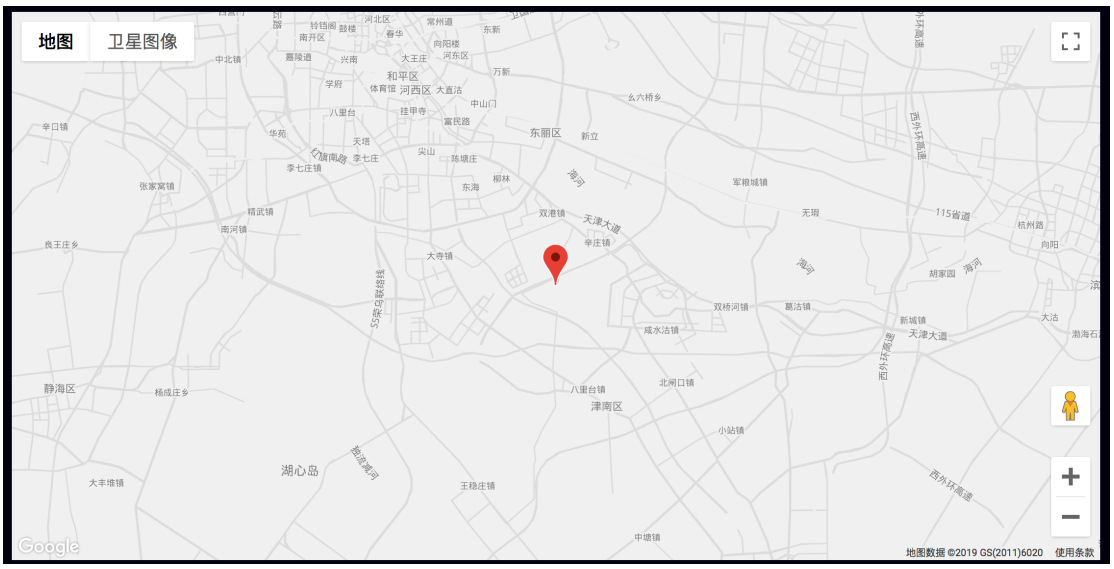
In the next section, you will see the details of the current TED video with the most statistics.

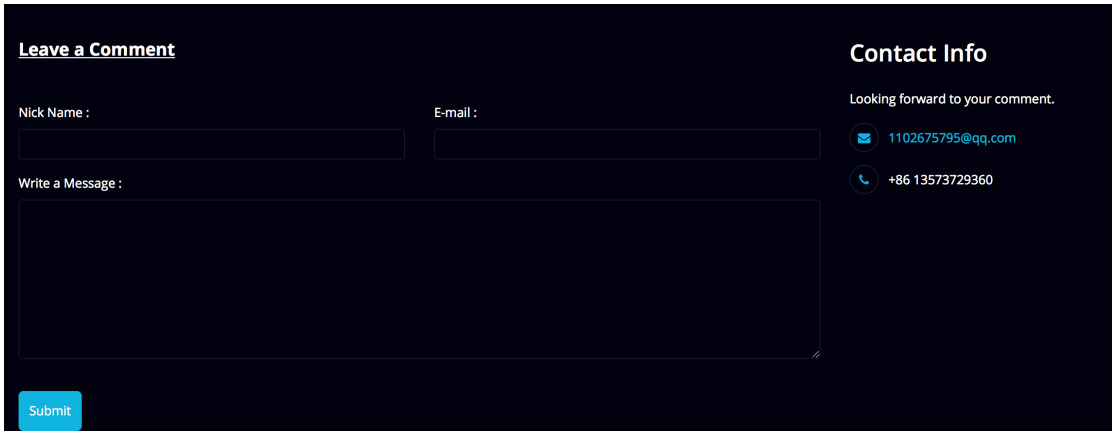


At the very bottom of the Data page, users can enter relevant data to predict the amount of comments on the video.

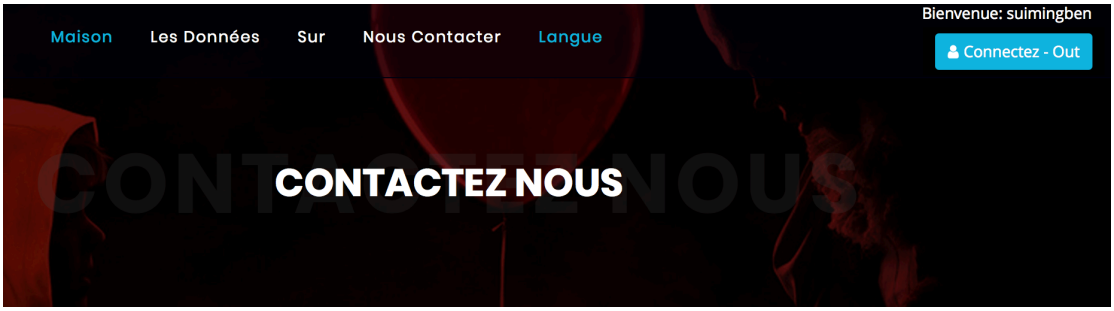


When the user enters the About Us page, they will see basic information about our team.





When the user enters the Contact Us page, they will see the specific location of our Chinese school. You can leave a message to us, you can learn more about us and can communicate with us.



Users can choose the language they want according to their needs.

Support: Chinese, English, French.

4.3 Something else about front end

Web design uses simple html, css, and js to construct the overall framework and structure. For example, simple styles:

```
.custom-btn {
background-color: #0fb5de; color: #fff; border:none; border-radius:2px;
transition: all 0.8s;

}
.custom-btn:hover {
   background-color: #fd7e14;

}
.datadiv{
width:80%;
margin:0 auto;
}
.banner{
    position:relative;
```

```
.arrow_left:before,
.arrow_left:after {
  position: absolute;
  content: "";
  left: 50px;
  top: 0;
  width: 50px;
  height: 80px;
  background: darkred;
  -moz-border-radius: 50px 50px 0 0;
  border-radius: 50px 50px 0 0;
  -webkit-transform: rotate(-45deg);
  -moz-transform: rotate(-45deg);
  -ms-transform: rotate(-45deg);
  -o-transform: rotate(-45deg);
```

The page translation part uses translater.js for remarks translation.

```
$('#toZH').click((el) => {
    var tran = new Translater({
        lang:"zh"
    });
});
$('#toFR').click((el) => {
    var tran = new Translater({
        lang:"fr"
    });
});
$('#toEN').click((el) => {
    var tran = new Translater({
        lang:"en"
    });
});
```

Call the Google Maps API to show the dynamic location of our Chinese schools.

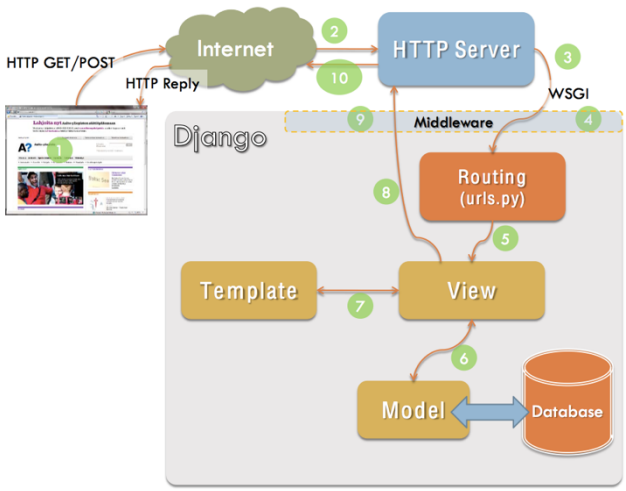
```
google.maps.event.addDomListener(window, 'load', init);
function init() {
    var mapOptions = {
        zoom: 11,
        center: new google.maps.LatLng(39.003408,117.321618),
        styles: [{
```

```
var mapElement = document.getElementById('map');
var map = new google.maps.Map(mapElement, mapOptions);
var marker = new google.maps.Marker({
    position: new google.maps.LatLng(39.003408,117.321618),
    map: map,
    title: 'Snazzy!'
});
```

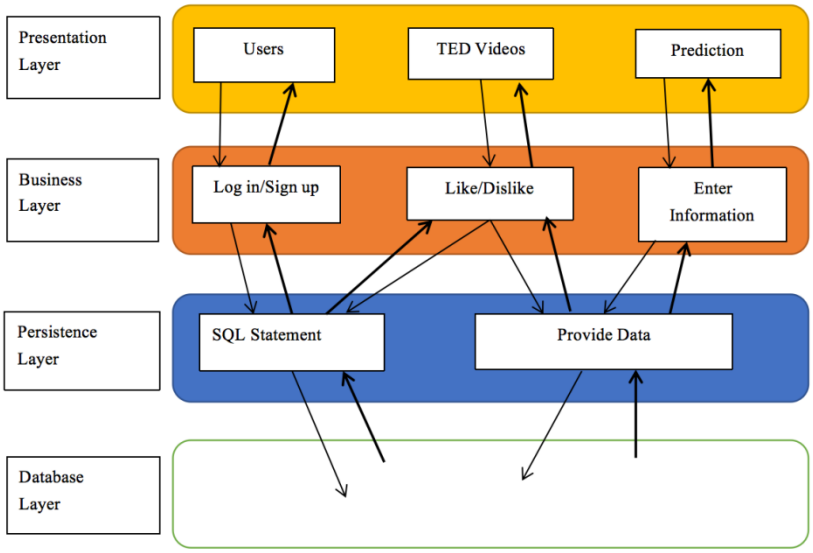
```
<script src="http://maps.google.com/maps/api/js?key=AIzaSyDBVF1z5zP0UpwGRPEpLWdY3eo8YnPy1-E"></script>
```

In addition, we put multiple data graphs on the web page to scroll.

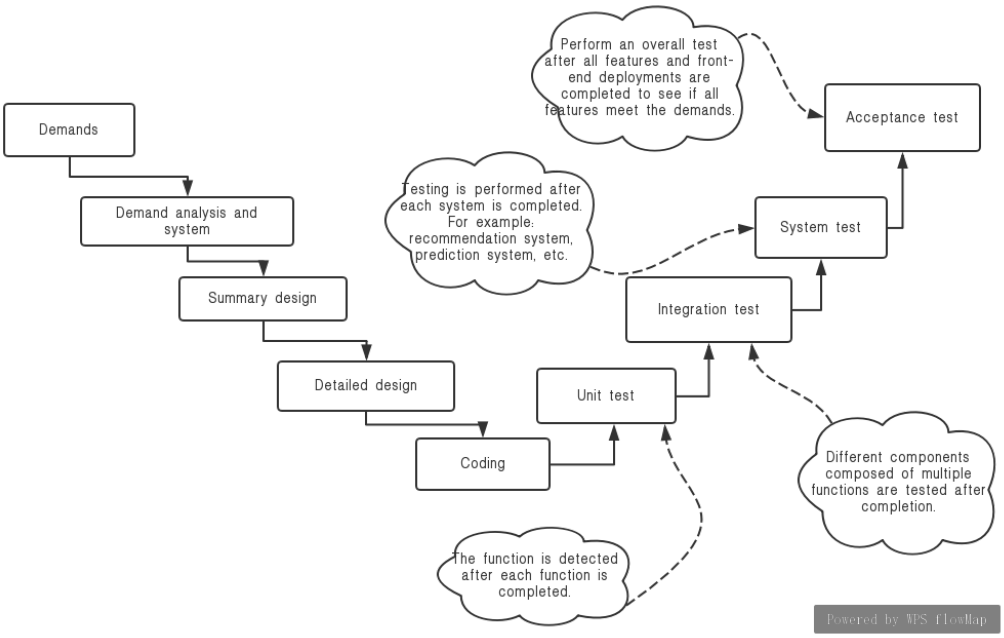
```
<div id="app-7" class="datadiv" style="height:558px" >
                <vue-slideshow :data="images" :config="config"></vue-slideshow>
            </div>
<br/>
```

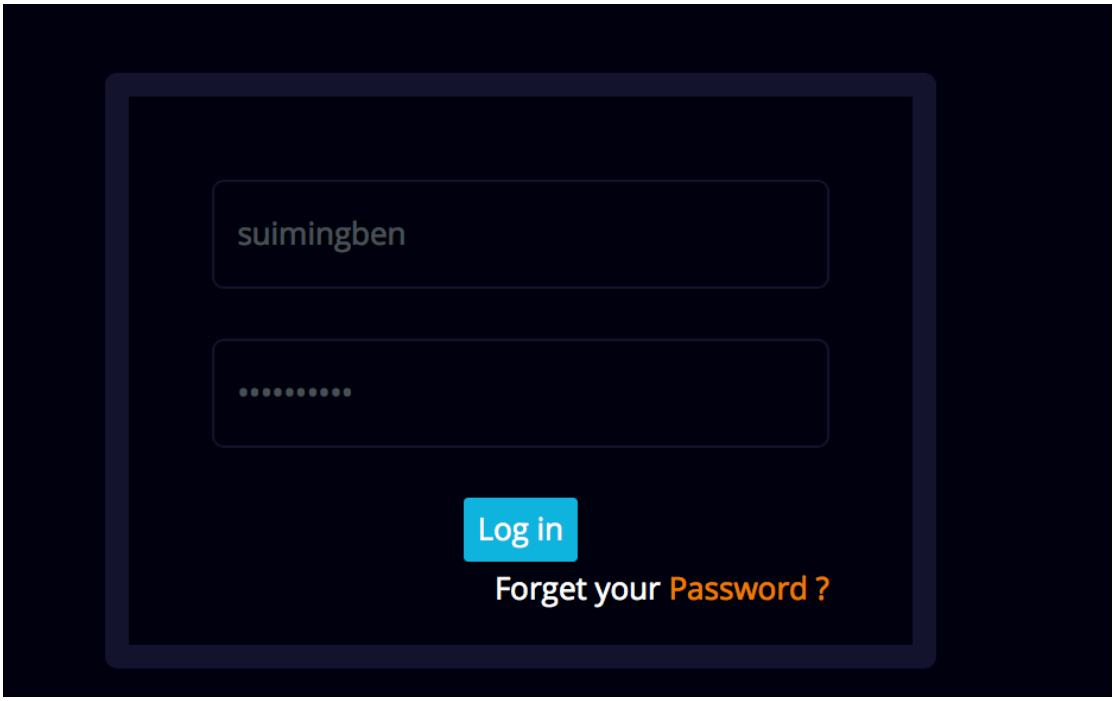


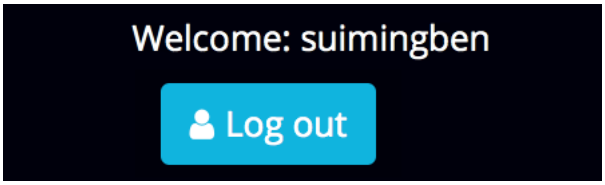
This is a project flow chart mentioned earlier, which briefly introduces the overall framework and front-end interaction.



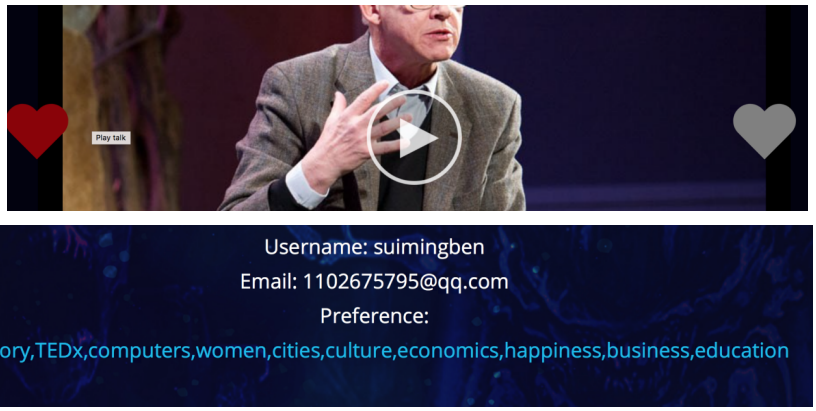
The hierarchical architecture of our project is shown in the figure. The whole is divided into four layers: presentation, business, business and database. When the user enters our website, they need to click the button to register and log in. The front end will pass the operation to the backend to activate the corresponding function and SQL statement to access the database for data storage and return the data to the front end for user interaction. For the user to use the prediction function, after inputting the information, the information is transmitted to the corresponding algorithm to call the corresponding algorithm, and the calculated result feedback is displayed in the webpage information box. The process for selecting a TED video for the user according to his or her preference is similar to the above two.


Demands
Demand analysis and system
Summary design
Detailed design
Coding
Unit test
Integration test
System test
Acceptance test
Perform an overall test after all features and front-end deployments are completed to see if all features meet the demands.
Testing is performed after each system is completed. For example: recommendation system, prediction system, etc.
Different components composed of multiple functions are tested after completion.
The function is detected after each function is completed.
Powered by WPS flowMap

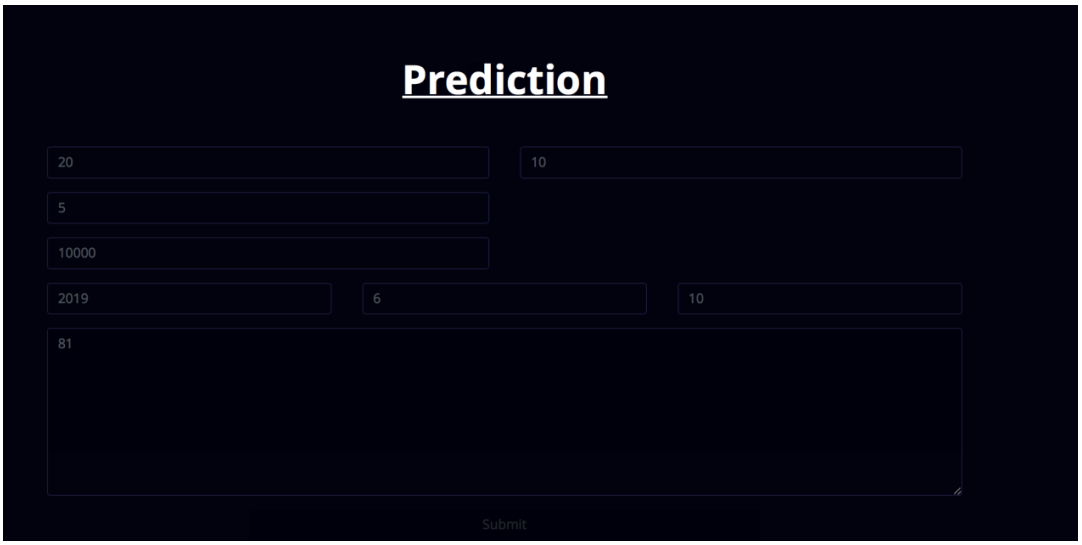

suimingben
Log in
Forget your Password ?



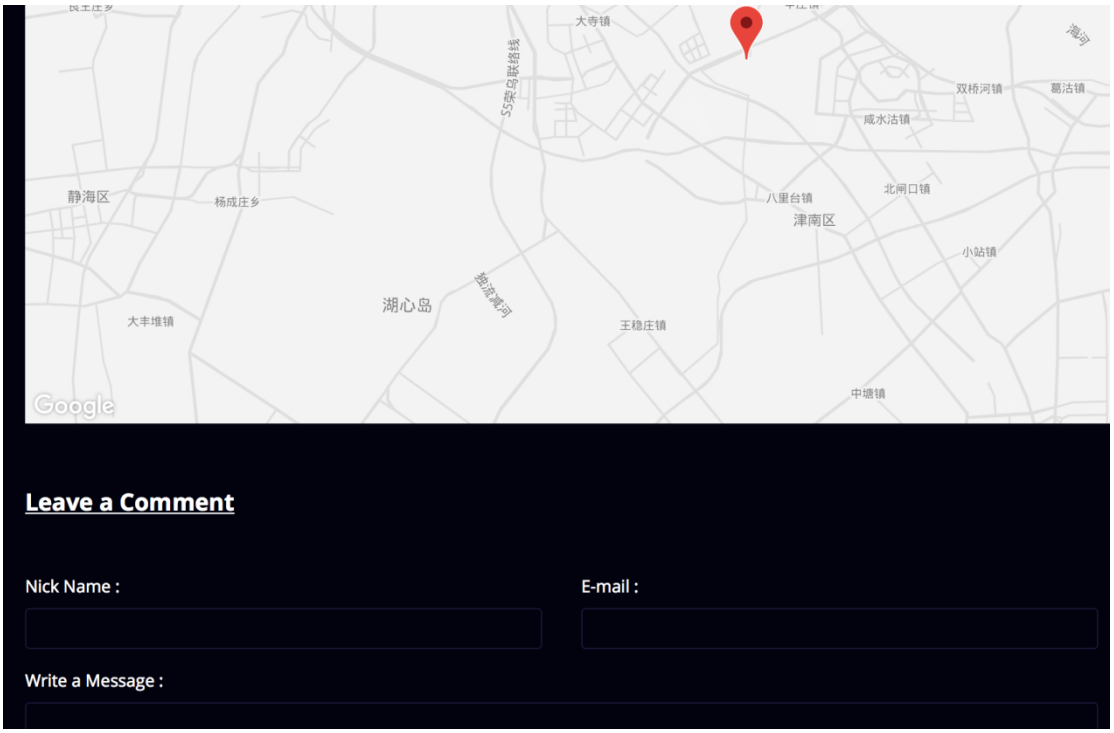
After we have registered the account, enter the account password and display the text in the upper right corner of the web and you can access the personal homepage to prove that the registration and log in function is normal.



When the user clicks the like button, the video can be replaced normally, and the profile of the user who likes the TED video is displayed on the personal homepage, which proves that the recommendation system is normal.



When the user tries to predict the function, enter different data and click the submit button, the website can display the corresponding different data, which proves that the prediction system is normal.



This feature is normal when the user can get a dynamic Google map and can enter a message to our team and the prompt is sent successfully.

If all the functions are working properly, the test is successful.