King Saud University

College of Computer and Information Sciences

Information Technology department

IT 362: Principles of data science

TMDB Movies

Section#:	76320	
Group Members	Name	ID
	Reem Almusharraf	442200387
	Deema Alresheed	442200938
	Danah Alotaibi	442200854
	Latifah Allaboun	442200484

Introduction

Introduction

Why do we choose to connect to TMDB ?

TMDB : The Movie Database (TMDB) is a community built movie and TV database. Every piece of data has been added by an amazing community dating back to 2008. TMDB's strong international focus and breadth of data is largely unmatched and something they are incredibly proud of. Put simply, they live and breathe community and that's precisely what makes us different. Furthermore we try to explore TMDB to find answers to questions of our interest which are:

- Does the popularity of top rated movies affect the chance to be from a particular language?
- what are the most frequent languages from the top rated movies?
- What is the language that got the most number of Vote Count?
- is it true that movies with the highest vote count has the highest popularity?
- what is the language that got the heights rating average?

Data Collection

Data Collection

How do we collect data?

We collect our dataset Through TMDB API which refers to any software with a distinct function. An interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses [1], we connect to TMDB API and retrieved top-rated movies by users so we can study these Movies' features that made them appear in the top rated. We retrieved 980 movie data and fill them into a data frame and applied some analysis and visualization techniques.

Data Collection

challenges in data collection

This project aims to analyze TMDB movies database, the features of top rated Movies, and the relationships and correlations between these features. One of the problems we faced was that TDBM provided inconsistent features (not all movies have the same attributes), so we decided to use some attributes in order to collect as many records as possible. Also, since TDMB provides results for each page separately, we had to go through each page to get the full data, then we merged these results to make the calculations and visualizations easier and cleaner. Of course, working with data presents many challenges and difficulties, but the treasure
resulting from these difficulties is great and powerful.

Data Overview

Data Overview

Our dataset consists of a 979 of rows, each row represents a movie and has 7 features, which are:

- Id
- Original title
- Overview
- Original Language
- Vote Average
- Vote Count
- Popularity

Sample of dataset:

[3]:

	id	original_title	overview	original_language	vote_average	vote_count	popularity\r
0	502356	The Super Mario Bros. Movie	While working underground to fix a water main,...	en	7.8	3432	5203.400
1	385687	Fast X	Over many missions and against impossible odds...	en	6.9	449	4334.616
2	603692	John Wick: Chapter 4	With the price on his head ever increasing, Jo...	en	8.0	1861	4746.497
3	840326	Sisu	Deep in the wilderness of Lapland, Aatami Korp...	fi	7.5	391	2367.432
4	713704	Evil Dead Rise	Three siblings find an ancient vinyl that give...	en	7.1	1292	2301.350

Features Description:

Feature	Description
Id	The id of movie.
Original title	The name of movie.
Overview	The description of movie.
Original Language	The movie language.
Vote Average	Averaging users rating for a movie.
Vote Count	Counting the number of users rating for movie.
Popularity	Measuring the movie popularity.

Info() function shows each feature and its type.

```
4]: dataset_csv.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 979 entries, 0 to 978
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 979 non-null    int64
 1   original_title     979 non-null    object
 2   overview           974 non-null    object
 3   original_language  979 non-null    object
 4   vote_average       979 non-null    float64
 5   vote_count         979 non-null    int64
        979 non-null    float64
dtypes: float64(2), int64(2), object(3)
memory usage: 53.7+ KB
```

Static measure:

We previewed the static measure using describe() for numeric features. The first figure is for numeric features and we noticed the vote_arerage its mean is less than its median, so the data is skewed to the left. In addition, vote_arerage has a low standard deviation, which implies that its values are close to each other. For vote_count and vote_average, we noticed min is equal to zero, which implies there are movies that have not been rated by users. For popularity, its mean is higher than its median, which indicates the data is skewed to the right and has a very high standard deviation, which means we have a big range of values and the spread is high.

For Nominal Features, we noticed the original_title count is 979 and its unique value is 966,which means there are 13 movie titles duplicated. For overview,the count and unique are equal, which indicates there are no duplicates for movie overview. For the last feature which is original_language the unique values equal 25, which implies that there are 25 languages for movies.

[3]:
```
dataset_csv.describe()
```

[3]:

	id	vote_average	vote_count	popularity\r
count	9.790000e+02	979.000000	979.000000	979.000000
mean	4.458146e+05	6.702349	4710.054137	124.886610
std	3.641892e+05	1.387913	6199.295685	315.613431
min	1.100000e+01	0.000000	0.000000	30.232000
25%	5.190100e+04	6.300000	124.000000	51.137000
50%	4.276410e+05	6.900000	1525.000000	64.609000
75%	7.706295e+05	7.500000	7315.500000	95.662000
max	1.122493e+06	10.000000	33725.000000	5203.400000

[4]:
```
dataset_csv.describe(include=['O'])
```

[4]:

	original_title	overview	original_language
count	979	974	979
unique	966	974	25
top	The Little Mermaid	While working underground to fix a water main,...	en
freq	3	1	740

Data Cleaning

Data Cleaning

Null Values:

After TMDB data overview and understand the features and their descriptions. We checked for null values by using isnull().sum(). As you can see, the overview has five null values. We decided to drop it since there is no meaning to fill in the null values of overview by the original title and also because only a few null values exist.

```
dataset_csv.isnull().sum()
```

```
id                  0
original_title      0
overview            5
original_language   0
vote_average        0
vote_count          0
popularity\r        0
dtype: int64
```

Before

```
dataset.isnull().sum()
```

```
Unnamed: 0          0
original_title      0
overview            0
original_language   0
vote_average        0
vote_count          0
popularity\r        0
dtype: int64
```

After

Duplicates values:

As we mentioned in the data overview section, there are duplicates of the original_title. By using the drop_duplicates() function, we dropped the duplicates.

```
dataset_csv['original_title'].duplicated().sum()
```

```
13
```

```
dataset_csv.drop_duplicates(['original_title'],keep="first" ,inplace=True)
```

Drop column:

The last thing we did on cleaning we decided to drop id column since we did not need for classifier model.

```
dataset_csv= dataset_csv.drop(columns='id')
dataset_csv.head()
```

	original_title	overview	original_language	vote_average	vote_count	popularity\r
0	The Super Mario Bros. Movie	While working underground to fix a water main,...	en	7.8	3432	5203.400
1	Fast X	Over many missions and against impossible odds...	en	6.9	449	4334.616
2	John Wick: Chapter 4	With the price on his head ever increasing, Jo...	en	8.0	1861	4746.497
3	Sisu	Deep in the wilderness of Lapland, Aatami Korp...	fi	7.5	391	2367.432
4	Evil Dead Rise	Three siblings find an ancient vinyl that give...	en	7.1	1292	2301.350

Data Analysis

Data Analysis

First, we check for correlation between the target and all predictors for all categorical and numeric variables to take benefits from them in the classification model.
using ANOVA hypothesis test:

Null hypothesis(H0): The variables are not correlated with each other.
P-value: The probability of Null hypothesis being true.
Accept Null hypothesis if P-value>0.05. Means variables are NOT correlated.
Reject Null hypothesis if P-value<0.05. Means variables are correlated.

The categorical variables:

```
# 1
#Cross tabulation between original_language and original_title
CrosstabResult=pd.crosstab(index=data['original_language'],columns=data['original_title'])
#print(CrosstabResult)
# importing the required function
from scipy.stats import chi2_contingency

# Performing Chi-sq test
ChiSqResult = chi2_contingency(CrosstabResult)

# P-Value is the Probability of H0 being True
# If P-Value > 0.05 then only we Accept the assumption(H0)

print('The P-Value of the ChiSq Test is:', ChiSqResult[1])
```

```
The P-Value of the ChiSq Test is: 0.45427408653971896
```

The numeric variables:

```
# f_oneway() function takes the group data as input and
# returns F-statistic and P-value
from scipy.stats import f_oneway

# Running the one-way anova test between vote_count and original_language
# Assumption(H0) is that original_language and vote_count are NOT correlated

# Finds out the vote_count data for each original_language as a list
CategoryGroupLists=data.groupby('original_language')['vote_count'].apply(list)

# Performing the ANOVA test
# We accept the Assumption(H0) only when P-Value > 0.05
AnovaResults = f_oneway(*CategoryGroupLists)
print('P-Value for Anova is: ', AnovaResults[1])
```

```
P-Value for Anova is:  5.1988301975680486e-18
```

Data Analysis

Second, make the data more balanced by merging all languages that have a small count of values whose count is less than 30 into one type, "Others".
To get more balanced datasets to make better classification models.

```
print(data['original_language'].value_counts())
```

```
en    727
ja     70
ko     39
es     37
fr     18
it     13
zh     10
cn     10
ru      6
pl      6
de      5
no      4
nl      2
th      2
eu      2
uk      1
fi      1
ro      1
is      1
te      1
id      1
tl      1
da      1
tr      1
sv      1
Name: original_language, dtype: int64
```

Before

```
data['original_language'] = data['original_language'].replace(
['fr', 'it','zh','cn','ru','pl','de','no','nl','th','eu','eu','uk','fi','ro','is','te','id','tl','da','tr','sv']
    ,'Others')
```

```
print(data['original_language'].value_counts())
```

```
en          727
Others       88
ja           70
ko           39
es           37
Name: original_language, dtype: int64
```

After

Data Analysis:

For Classifier model we use both **Logistic Regression** and **SVM:**

1. Choose “original_language” as a class label (response):

```
X = df1.drop('original_language',axis=1)  #Predictors
y = df1['original_language']              #Response
```

2. Split the data into a training set and testing set and fit a model using sklearn:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y)
#X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.2) #80% training, 20% testing

print (X_train.shape,X_test.shape,y_train.shape, y_test.shape)

#split 75% 25%
```

3. Apply the model on the test data make a prediction

SVM

```
from sklearn.svm import SVC
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

SVM = SVC(gamma='auto')
SVM.fit(X_train, y_train)
print('Accuracy of SVM classifier on test set: {:.2f}'.format(SVM.score(X_test, y_test)))
```

```
Accuracy of SVM classifier on test set: 0.73
```

```
#Make prediction for the test data
y_pred= SVM.predict(X_test)
#print(y_pred)
```

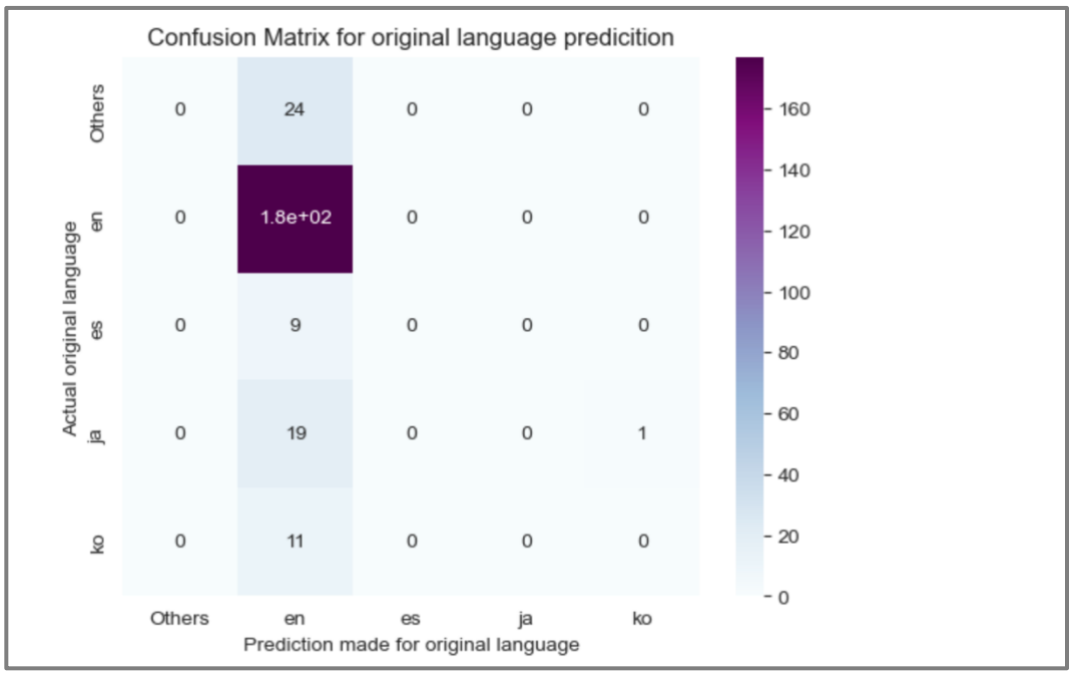
Logistic Regression

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score



logreg = LogisticRegression()
logreg.fit(X_train, y_train)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))
```

```
#Make prediction for the test data
y_pred= SVM.predict(X_test)
#print(y_pred)
```

4. Evaluate the model accuracy using the confusion matrix.

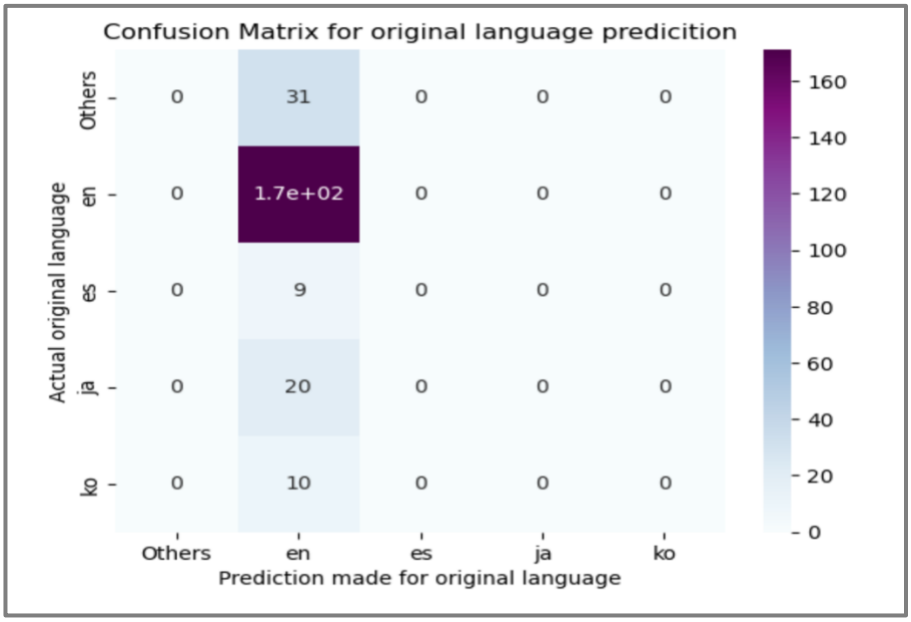
SVM




```
print(classification_report(y_test, SVM_prediction))
```

	precision	recall	f1-score	support
Others	0.00	0.00	0.00	24
en	0.74	1.00	0.85	177
es	0.00	0.00	0.00	9
ja	0.00	0.00	0.00	20
ko	0.00	0.00	0.00	11
accuracy			0.73	241
macro avg	0.15	0.20	0.17	241
weighted avg	0.54	0.73	0.62	241

Logistic Regression



	precision	recall	f1-score	support
Others	0.00	0.00	0.00	31
en	0.71	1.00	0.83	171
es	0.00	0.00	0.00	9
ja	0.00	0.00	0.00	20
ko	0.00	0.00	0.00	10
accuracy			0.71	241
macro avg	0.14	0.20	0.17	241
weighted avg	0.50	0.71	0.59	241

Findings

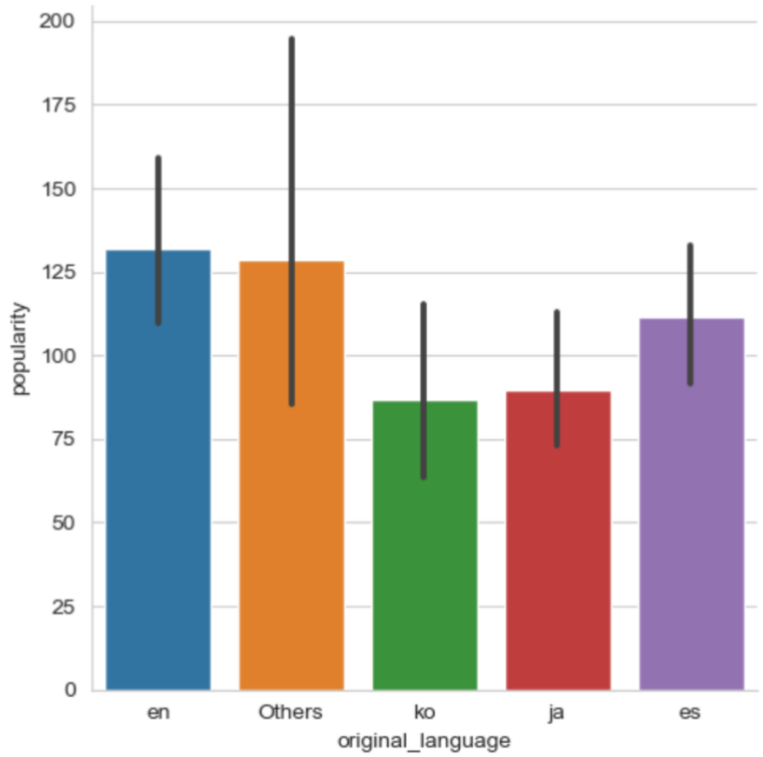
The accuracy is 73% in confusion matrix and classifier model.

Since "others", "es”, “ja" and "ok" have 0 values for the f1-score, recall, and precision, and the English language has almost 1, then the prediction of it is the accurate one.

Questions

Q1:Does the popularity of top rated movies affect the chance to be from a particular language?

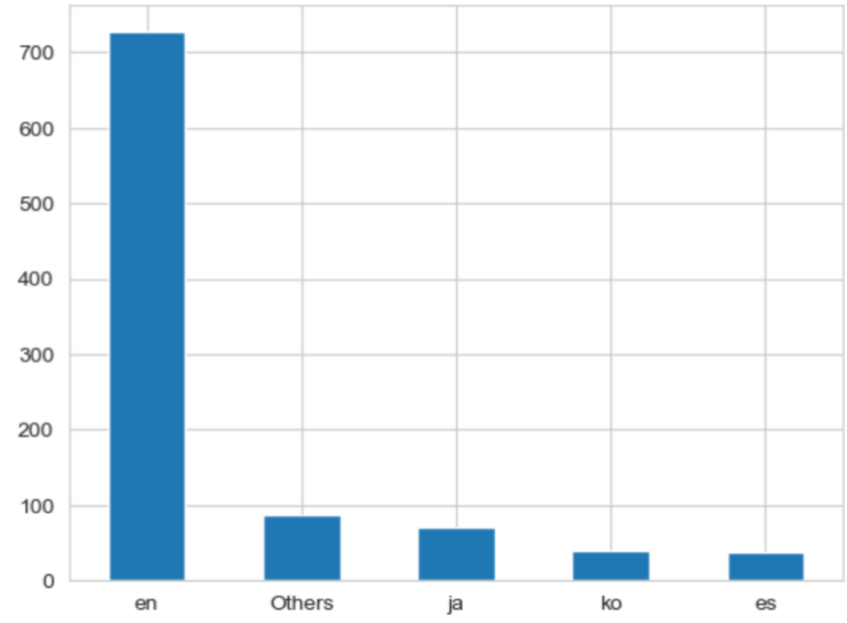
```
In [35]: #df1.plot.bar(x = 'original_language', y = 'popularity', rot = 40)
         sns.catplot(x="original_language", y="popularity", kind="bar", data=df1);
```



From the par plot we conduct that the English movies have the heights popularity, so we can say that the language effect the popularity.

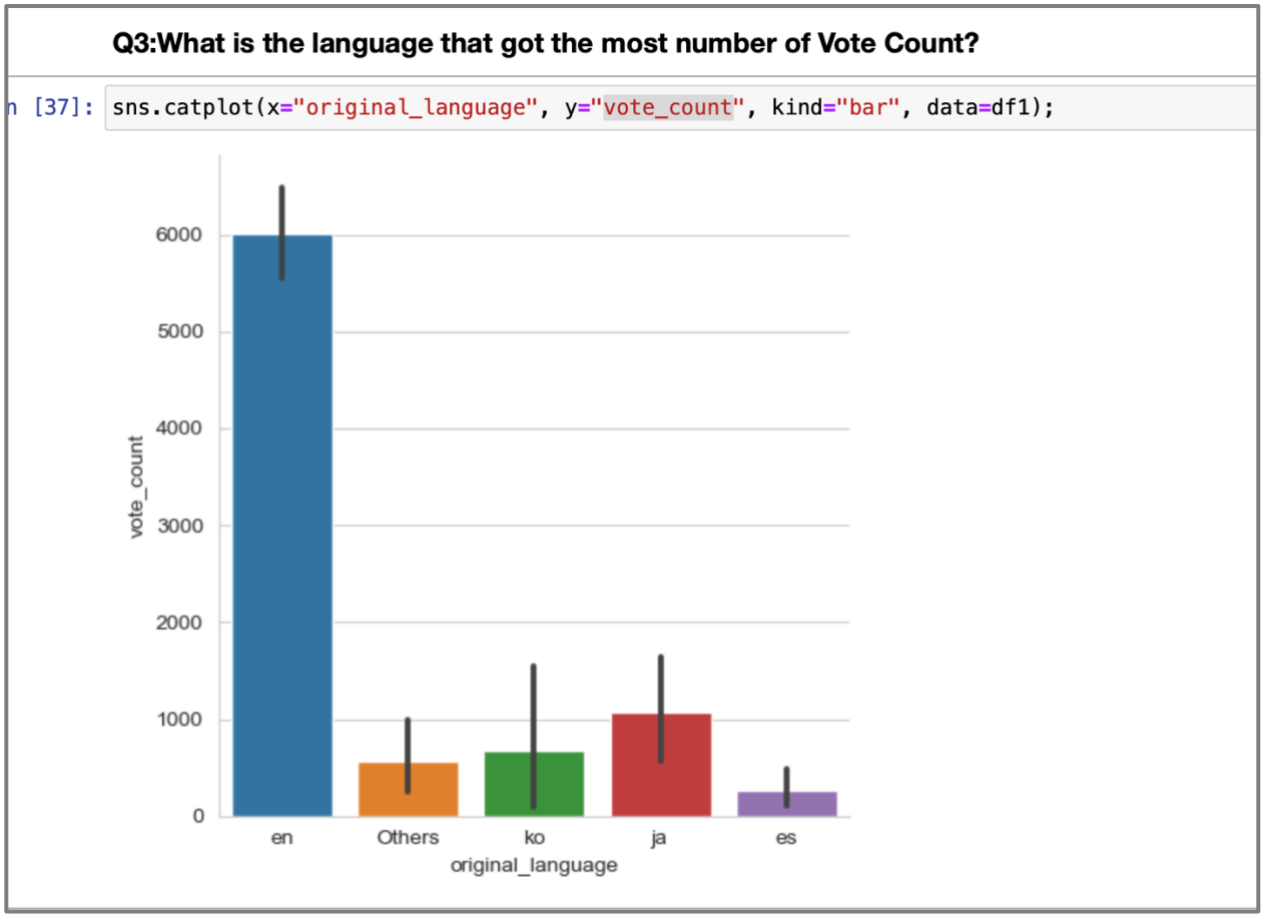
Q2:what are the most frequent language from the top rated movies?

```
In [36]: df1.original_language.value_counts().plot(kind='bar')
         plt.xticks(rotation=0);
```



From bar plot above we concluded that most top rated movies are in English.

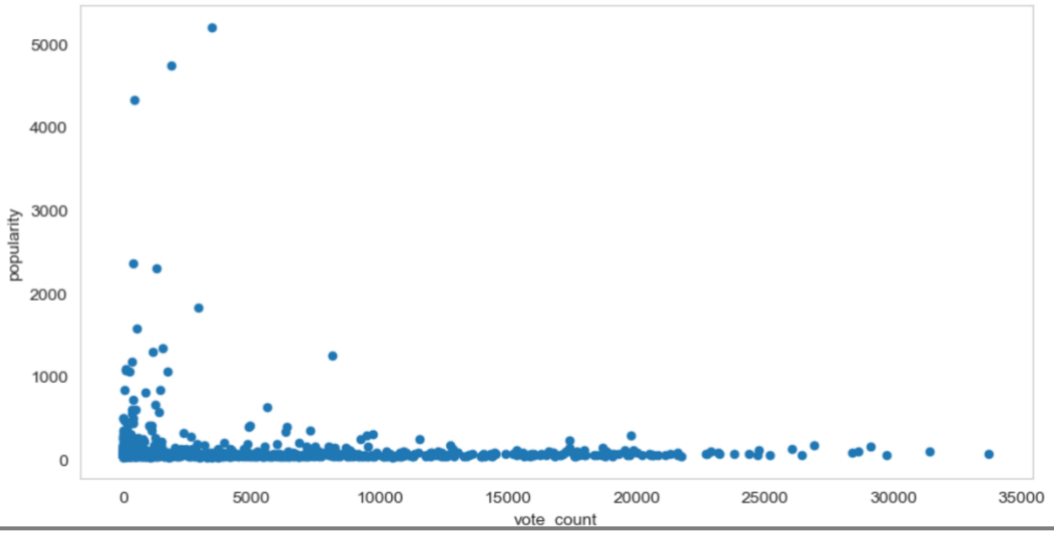
Q3:What is the language that got the most number of Vote Count?



From the figure above we can say that English movies have the heights vote count.

Q4:is it true that movies with the highest vote count has the highest popularity?

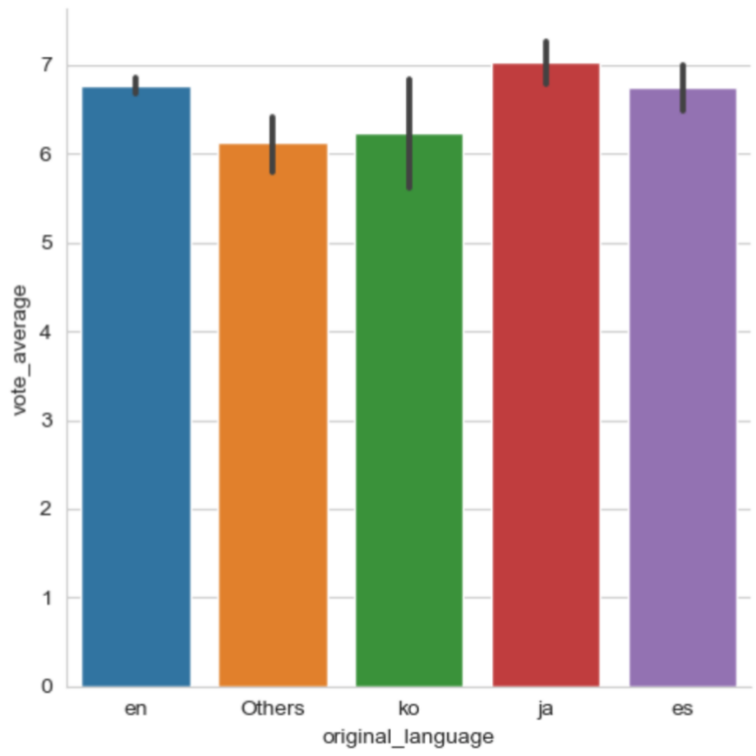
```
In [64]: df1.plot.scatter(x='vote_count', y='popularity', figsize=(10,5), grid=False)
Out[64]: <Axes: xlabel='vote_count', ylabel='popularity'>
```



From the scatter plot above we notice that there is a negative relationship between them and conducted that the high vote doesn’t imply high popularity.

Q5:what is the language that got the heights rating average?

```
In [88]: #df1.plot.scatter(x='vote_average', y='popularity', figsize=(10,5), grid=False)
         sns.catplot(x="original_language", y="vote_average", kind="bar", data=df1);
```



From the bar plot above can say that Japanese moves have the heights rating

Conclusion

Conclusion

At the end, our dataset contains knowledge about the most popular movies based on the original language. Many other interesting possibilities can be explored using this dataset. Briefly, our insights about data analysis and visualization that we applied to the dataset can be shown as follows:

The website we studied has good prediction results, and we visualize them using different chart types. We notice that the English language was the most common type that predicted more successfully than the other languages, followed by the Japanese language, so we conclude that if you want to make your film popular and have a high rating, make it in the English language.

Finally, just as the ways in which questions can be asked vary, so does the way in which a problem is tackled. We have tried as much as possible to have our analysis of the data be minutely observed, tested, and provide results that are trustworthy.

References

Bibliography

[1] "What Is An API (Application Programming Interface)?," https://aws.amazon.com/what-is/api/#:~:text=API%20stands%20for%20Application%20Programming,other%20using%20requests%20and%20responses.. [Online]. [Accessed 30 5 2023].

[2] F. Hashmi, "how-to-measure-the-correlation-between-two-categorical-variables-in-python," Thinking Neuron, 10 Marach 2022. [Online]. Available: https://thinkingneuron.com/how-to-measure-the-correlation-between-two-categorical-variables-in-python/. [Accessed 30 May 2023].

[3] F. Hashmi, "https://thinkingneuron.com/how-to-measure-the-correlation-between-a-numeric-and-a-categorical-variable-in-python/," Thinking Neuron, 1 April 2022. [Online]. Available: https://thinkingneuron.com/how-to-measure-the-correlation-between-a-numeric-and-a-categorical-variable-in-python/. [Accessed 29 May 2023].

[4] A. Nagori, "logistic-regression-using-python-and-excel," Analytics Vidhya, 8 February 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/02/logistic-regression-using-python-and-excel/. [Accessed 28 May 2023].

[5] TMDB, [Online]. Available: https://www.themoviedb.org. [Accessed 26 May 2023].