

```

mean_fit_time    0.01162
environment_comes_with_many_useful_analytics 0.01194
0.010739        0.011338
0.011609        0.009943
0.010204        0.010253
0.010347        0.01093
0.011097        0.010624
0.010688        0.010668
0.062692        0.066256
0.062618        0.063578
0.06313         0.062632

# This run will take many minutes to run
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 2000 to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session

/kaggle/input/popular-video-games-1980-2023/games.csv

```

Loading the data

```
[7]: df = pd.read_csv("/kaggle/input/popular-video-games-1980-2023/games.csv")
```

Data Preprocessing

```
[8]: df.head(10)
```

0	0	Elden Ring	Feb 25, 2022	[Bandai Namco Entertainment, FromSoftware]	4.5	3.9K	3.9K	[Adventure, RPG]	Elden Ring is a fantasy, action and open world...	["The first playthrough of elden ring is one o...	17K	3.8K	4.6K	4.8K
1	1	Hades	Dec 10, 2019	[\$uperlant Games]	4.3	2.9K	2.9K	[Adventure, Brower, Indie, RPG]	A rogue-like hack and slash dungeon crawler in...	["convinced this is a roguelike for people who...	21K	3.2K	6.3K	3.6K
2	2	The Legend of Zelda: Breath of the Wild	Mar 03, 2017	[Nintendo, Nintendo ESD Production Group No...]	4.4	4.3K	4.3K	[Adventure, RPG]	The Legend of Zelda: Breath of the Wild is the...	["This game is the game (that is not CS:GO) th...	30K	2.5K	5K	2.6K
3	3	Undertale	Sep 15, 2015	[tobyfox, '8-4']	4.2	3.5K	3.5K	[Adventure, Indie, RPG, Turn Based Stra...	A small child falls into the Underground, wher...	["soundtrack is tied for #1 with niet automata...	28K	679	4.9K	1.8K
4	4	Hollow Knight	Feb 24, 2017	['Team Cherry']	4.4	3K	3K	[Adventure, Indie, Platform]	A 2D metroidvania with an emphasis on close...	["this games worldbuilding is incredible, with...	21K	2.4K	8.3K	2.3K
5	5	Minecraft	Nov 18, 2011	['Mojang Studios']	4.3	2.3K	2.3K	[Adventure, Simulator]	Minecraft focuses on allowing the player to ex...	["Minecraft is what you make of it. Unfortunat...	33K	1.8K	1.1K	230
6	6	Omori	Dec 25, 2020	['OMOCAT', PLAYISM]	4.2	1.6K	1.6K	[Adventure, Indie, RPG, Turn Based Stra...	A turn-based surreal horror RPG in which a chi...	["The best game I've played in my life", "omori...	7.2K	1.1K	4.5K	3.8K
7	7	Metroid Dread	Oct 07, 2021	[Nintendo, MercurySteam]	4.3	2.1K	2.1K	[Adventure, Platform]	Join intergalactic bounty hunter Samus Aran i...	["Have only been a Metroid fan for couple of ...	9.2K	759	3.4K	3.3K
8	8	Among Us	Jun 15, 2018	['InnerSloth']	3.0	867	867	[Indie, Strategy]	Join your crew-mates in a multiplayer game of ...	["It's a solid party game. I'm bad at lying th...	25K	470	776	126
9	9	Nier: Automata	Feb 23, 2017	[PlatinumGames, Square Enix]	4.3	2.9K	2.9K	[Brower, RPG]	Nier: Automata tells the story of androids 28...	["Holy shit", "I'm carrying the weight of the w...	18K	1.1K	6.2K	3.6K

```
[9]: df.shape
```

```
(1512, 14)
```

```
[10]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1512 entries, 0 to 1511
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype  
2   Release Date    1512 non-null   object 
3   Team            1512 non-null   object 
4   Rating          1509 non-null   float64 
5   Times Listed    1512 non-null   object 
6   Number of Reviews 1512 non-null   object 
7   Genres          1512 non-null   object 
8   Summary          1512 non-null   object 
9   Reviews          1512 non-null   object 
10  Plays            1512 non-null   object 
11  Playing          1512 non-null   object 
12  Backlogs         1512 non-null   object 
13  Wishlist         1512 non-null   object 
dtypes: float64(1), int64(1), object(12)
memory usage: 165.5+ KB
```

```
[11]: df.describe()
```

	Unnamed: 0	Rating
count	1512.000000	1499.000000
mean	755.500000	3.719346
std	436.621117	0.532608
min	0.000000	0.700000
25%	377.750000	3.400000
50%	755.500000	3.800000
75%	1133.250000	4.100000
max	1511.000000	4.800000

```
[222]: df.isnull().sum()
```

	Release Date	Team	Rating	Times Listed	Number of Reviews	Genres	Summary	Reviews	Plays	Playing	Backlogs	Wishlist
count	8	1	13	0	0	0	0	0	0	0	0	0
mean	0	0	0	0	0	0	0	0	0	0	0	0
std	0	0	0	0	0	0	0	0	0	0	0	0
min	0	0	0	0	0	0	0	0	0	0	0	0
25%	0	0	0	0	0	0	0	0	0	0	0	0
50%	0	0	0	0	0	0	0	0	0	0	0	0
75%	0	0	0	0	0	0	0	0	0	0	0	0
max	0	0	0	0	0	0	0	0	0	0	0	0
dtype	int64	object	float64	int64	int64	object	object	int64	int64	int64	int64	int64

```
[223]: total_null = df.isnull().sum().sort_values(ascending = False)
percent = ((df.isnull().sum()/df.isnull().count())*100).sort_values(ascending = False)
print("Total records = ", df.shape[0])

missing_data = pd.concat([total_null,percent.round(2)],axis=1,keys=['Total Missing','In Percent'])

missing_data
```

	Total Missing	In Percent
Rating	13	0.86
Team	1	0.07
Summary	1	0.07
Unnamed: 0	0	0.00
Title	0	0.00
Release Date	0	0.00
Times Listed	0	0.00
Number of Reviews	0	0.00
Genres	0	0.00
Reviews	0	0.00

Playing	0	0.00
Backlogs	0	0.00
Wishlist	0	0.00

```
[224]: df.head()
```

	Unnamed: 0	Title	Release Date	Team	Rating	Times Listed	Number of Reviews	Genres	Summary	Reviews	Plays	Playing	Backlogs	Wishlist
0	0	Elden Ring	Feb 25, 2022	[Bandai Namco Entertainment, FromSoftware]	4.5	3.9K	3.9K	[Adventure, RPG]	Elden Ring is a fantasy, action and open world...	[The first playthrough of elden ring is one o...	17K	3.8K	4.6K	4.8K
1	1	Hades	Dec 10, 2019	[Supergiant Games]	4.3	2.9K	2.9K	[Adventure, Brawler, Indie, RPG]	A rogue-lite hack and slash dungeon crawler in...	[convinced this is a roguelike for people who...	21K	3.2K	6.3K	3.6K
2	2	The Legend of Zelda: Breath of the Wild	Mar 03, 2017	[Nintendo, Nintendo EPD Production Group No...]	4.4	4.3K	4.3K	[Adventure, RPG]	The Legend of Zelda: Breath of the Wild is the...	[This game is the game that is not CSGO though]	30K	2.5K	5K	2.6K
3	3	Undertale	Sep 15, 2015	[tobyfox, 8-4]	4.2	3.5K	3.5K	[Adventure, Indie, RPG, Turn-Based Strateg...	A small child falls into the Underground, where...	[soundtrack is tied for #1 with nier automata...	28K	679	4.9K	1.8K
4	4	Hollow Knight	Feb 24, 2017	[Team Cherry]	4.4	3K	3K	[Adventure, Indie, Platform]	A 2D metroidvania with an emphasis on close co...	[this games worldbuilding is incredible, with...	21K	2.4K	8.3K	2.3K

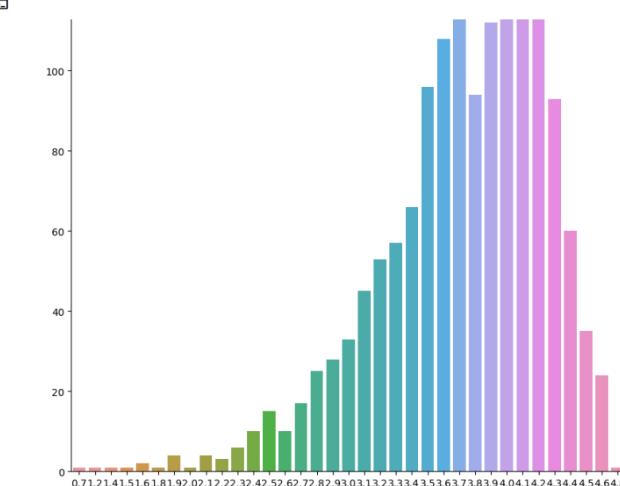
```
[225]: df['Rating'].value_counts()
```

```
4.1    131
4.0    124
3.9    113
3.8    112
3.6    108
3.5    96
3.4    94
4.3    93
3.4    66
```

```
3.1    45
4.5    35
3.0    33
2.9    28
2.8    25
4.6    24
2.7    17
2.5    15
2.6    10
2.4    10
2.3    6
2.1    4
1.5    4
2.2    3
1.6    2
2.0    1
0.7    1
1.4    1
1.2    1
4.8    1
1.5    1
1.8    1
Name: Rating, dtype: int64
```

```
[226]: import matplotlib.pyplot as plt
import seaborn as sns
```

```
[227]: plt.figure(figsize=(10, 10))
sns.barplot(x=df['Rating'].value_counts().index,y=df['Rating'].value_counts().values)
plt.show()
```



[12]:

```
Rating      Rating
Rating    1.0
Rating   -0.3
```

[228]:

```
high_rating={0:'0',1:'1',2:'2',3:'3',4:'4',5:'5'}
```

As the percentage of null values is not very high we will replace the null values

[229]:

```
df['Rating'].fillna(value = df['Rating'].mean().round(0), inplace=True)
```

[230]:

```
df.isnull().sum()
```

```
Unnamed: 0      0
Title          0
Release Date   0
Team           1
Rating          0
Times Listed   0
Number of Reviews  0
Genres          0
Summary         1
Reviews         0
Plays           0
Playing         0
Backlogs        0
Wishlist        0
dtype: int64
```

[231]:

```
df = df.dropna()
```

[232]:

```
df.isnull().sum()
```

```
Unnamed: 0      0
Title          0
Release Date   0
Team           0
Rating          0
Times Listed   0
Number of Reviews  0
Genres          0
Plays           0
Playing         0
Backlogs        0
Wishlist        0
dtype: int64
```

[233]:

```
df.shape
```

```
(1510, 14)
```

[234]:

```
df.columns
```

```
Index(['Unnamed: 0', 'Title', 'Release Date', 'Team', 'Rating', 'Times Listed',
       'Number of Reviews', 'Genres', 'Summary', 'Reviews', 'Plays', 'Playing',
       'Backlogs', 'Wishlist'],
      dtype='object')
```

Libraries

[235]:

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn import tree
```

[236]:

```
from sklearn.metrics import accuracy_score
import string
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
```

[237]:

```
df.drop(['Genres'], axis=1, inplace=True)
df.drop(['Unnamed: 0'], axis=1, inplace=True)
df.drop(['Title'], axis=1, inplace=True)
df.drop(['Release Date'], axis=1, inplace=True)
df.drop(['Team'], axis=1, inplace=True)
df.drop(['Times Listed'], axis=1, inplace=True)
df.drop(['Number of Reviews'], axis=1, inplace=True)
df.drop(['Summary'], axis=1, inplace=True)
df.drop(['Plays'], axis=1, inplace=True)
df.drop(['Playing'], axis=1, inplace=True)
df.drop(['Backlogs'], axis=1, inplace=True)
```

```
[236]: df.drop(['Wishlist'], axis=1, inplace=True)
```

```
[237]: df.head(10)
```

	Rating	Reviews
0	4.5	["The first playthrough of elden ring is one o...
1	4.3	["convinced this is a rogue-like for people who...
2	4.4	["This game is the game (that is not CSGO) tha...
3	4.2	["soundtrack is tied for #1 with nier automata...
4	4.4	["This game's worldbuilding is incredible, with...
5	4.3	["Minecraft is what you make of it. Unfortunat...
6		
7	4.3	["Have only been a Metroid fan for couple of y...
8	3.0	["It's a solid party game. I'm bad at lying th...
9	4.3	["Holy shit; I'm carrying the weight of the w...

Removing hashtags

```
[238]: df['Reviews'].replace( { r"#(\w+)" : '' }, inplace=True, regex=True)
```

Removing Mentions

```
[239]: df['Reviews'].replace( { r"@(\w+)" : '' }, inplace=True, regex=True)
```

Removing URLs

```
[240]: df['Reviews'].astype(str).replace( { r"http\S+" : '' }, inplace=True, regex=True)
```

Removing LowerCase

```
[241]: df['Reviews']=df['Reviews'].str.lower()
```

Removing StopWords

```
[242]: from nltk.corpus import stopwords  
stop = stopwords.words('english')  
df['Reviews'] = df['Reviews'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
```

Randomization

```
[243]: df = df.sample(frac = 1)
```

EDA for final Dataset

```
[244]: df.head(10)
```

	Rating	Reviews
700	2.8	["the custom dice make characters clearly bett...
323	3.9	["I loved game, time spent loved it, pretty da...
666	3.7	["this game funnnn made mitsuru even hot", "te...
170	4.3	["this game absolutely amazing.\n\ngameplay --
135	3.1	["recomendado, pero tener en cuenta que es la...
1225	4.2	["I enjoyed 8.11 much 4. really see appeal ser...
752	4.1	["there's genuinely nothing funny could say th...
695	3.1	["the story really short think racing game rea...
1307	3.4	["I think game mediocre best, controls battle ...
654	4.0	["played on: classic ps4 completed", "ergen...

```
[245]: df.shape
```

```
[246]: df.describe()
```

mean	3.722252
std	0.530596

```
min    0.700000
25%   3.400000
50%   3.800000
75%   4.100000
max    4.800000
```

[247]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1510 entries, 700 to 899
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   Rating   1510 non-null   float64
 1   Reviews  1510 non-null   object  
dtypes: float64(1), object(1)
memory usage: 35.4+ KB
```

[248]: df.columns

```
[248]: Index(['Rating', 'Reviews'], dtype='object')
```

[249]: df.isnull().sum()

Vectorization

Text Vectorization

[250]: from sklearn.feature_extraction.text import TfidfVectorizer

[251]: y = round(df['Rating'])

x = df['Reviews']

Creating a word corpus for vectorization

corpus = []

for i in range(x.shape[0]):

corpus.append(x.iloc[i])

vectorizer1 = TfidfVectorizer(max_features=1000)

X1 = vectorizer1.fit_transform(x)

feature_names1 = vectorizer1.get_feature_names()

denseslist1 = X1.todense().tolist()

train = pd.DataFrame(denseslist1, columns=feature_names1)

/opt/conda/lib/python3.7/site-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.

warnings.warn(msg, category=FutureWarning)

BOW Vectorization

[252]: from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

CountVec = CountVectorizer(ngram_range=(1,1), # to use bigrams ngram_range=(2,2)

stop_words='english')

#transform

Count_data = CountVec.fit_transform(x)

#create dataframe

cv_dataframe=pd.DataFrame(Count_data.toarray(),columns=CountVec.get_feature_names())

print(cv_dataframe)

```
    000 007 008 01 03 00 09 05 10 100 ... 400j 3ij 精神分裂病 \
0    0    0    0    0    0    0    0    0    0 ...    0    0    0
1    0    0    0    0    0    0    0    0    0 ...    0    0    0
2    0    0    0    0    0    0    0    0    0 ...    0    0    0
3    0    0    0    0    0    0    0    0    0 ...    0    0    0
4    0    0    0    0    0    0    0    0    0 ...    0    0    0
...
1585 0    0    0    0    0    0    0    0    0 ...    0    0    0
1586 0    0    0    0    0    0    0    0    0 ...    0    0    0
1587 0    0    0    0    0    0    0    0    0 ...    0    0    0
1588 0    0    0    0    0    0    0    0    1 ...    0    0    0
1589 0    0    0    0    0    0    0    0    0 ...    0    0    0
精神分裂症 茶番鬼 SHIT Decepc Gamer Major Minha
0    0    0    0    0    0    0    0    0
1    0    0    0    0    0    0    0    0
2    0    0    0    0    0    0    0    0
3    0    0    0    0    0    0    0    0
4    0    0    0    0    0    0    0    0
...
1585 0    0    0    0    0    0    0    0
1586 0    0    0    0    0    0    0    0
1587 0    0    0    0    0    0    0    0
1588 0    0    0    0    0    0    0    0
1589 0    0    0    0    0    0    0    0

```

[1510 rows x 18860 columns]

[253]: cv_dataframe

Rating Reviews

700 2.8 [the custom dice make characters clearly bett...

323 3.9 [I loved game, time spent loved it, pretty da...

666 3.7 [this game funnnn made mitsuru even hot', 'te...

170 4.3 [this game absolutely amazing.\n\ngameplay...

135 3.1 [recomendado, pero tener en cuenta que es la...

1225 4.2 [I enjoyed 8.11 much 4. really see appeal ser...

752 4.1 [there's genuinely nothing funny could say th...

695 3.1 [the story really short think racing game rea...

1307 3.4 [I think game mediocre best, controls battle ...

654 4.0 [played on classic ps4 | completed/, ergen ...

[254]: cv_dataframe.head

[254]: <bound method NDFrame.head of 000 007 008 01 03 00 09 05 10 100 ... 400j 3ij 精神分裂病 \
0 0 0 0 0 0 0 0 0 0 ... 0 0 0
1 0 0 0 0 0 0 0 0 0 ... 0 0 0

```

2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...   ...
1505 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
1506 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
1507 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
1508 0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0
1509 0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
[1510 rows x 18860 columns]

```

Model Application

Applying the modals

```
[255]:
from sklearn.linear_model import LogisticRegression
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LinearRegression, LogisticRegression
```

```
[256]:
import string
from sklearn.model_selection import train_test_split
```

```
# accuracy = {'TF-IDF':[]}

# #Linear Regression
# linearrr = LinearRegression()
# linearrr.fit(X_temp,y_temp)
# predict = linearrr.predict(X_test)
# a = linearrr.score(X_test,y_test)
# lrscore = a
# accuracy['TF-IDF'].append(a)

# # Logistic Regression
# regressor_LR_tf = LogisticRegression(C=1.0,penalty='l2',solver='newton-cg')
# regressor_LR_tf.fit(X_temp, y_temp)
# y_predict_LR_tf = regressor_LR_tf.predict(X_test)
# a=(regressor_LR_tf.score(X_test, y_test))
# lrscore = a
# accuracy['TF-IDF'].append(a)

# # Decision Tree

# model_DT_tf = DecisionTreeClassifier(criterion='gini', max_depth=2)
# model_DT_tf.fit(X_temp, y_temp)
# y_predict_DT_tf = model_DT_tf.predict(X_test)
# accuracyscoretree = a
# accuracy['TF-IDF'].append(a) # Append accuracy to the list

# ## Gradient Boosting
# model_GB_tf = GradientBoostingClassifier(learning_rate= 0.1, max_depth=3, n_estimators= 100,subsample=1.0)
# model_GB_tf.fit(X_temp, y_temp)
# y_predict_GB_tf = model_GB_tf.predict(X_test)
# a=(model_GB_tf.score(X_test, y_test))
# accuracy['TF-IDF'].append(a)

# # Random Forest
# model_RF_tf = RandomForestClassifier(n_estimators= 100, max_features = 'log2')
# model_RF_tf.fit(X_temp, y_temp)
# y_predict_RF_tf = model_RF_tf.predict(X_test)

# accuracy['TF-IDF'].append(a)

# # K-Neighbors Classifier
# model_KN_tf = KNeighborsClassifier(metric= 'manhattan', n_neighbors= 5, weights= 'distance')
# model_KN_tf.fit(X_temp, y_temp)
# y_predict_KN_tf = model_KN_tf.predict(X_test)
# a=(model_KN_tf.score(X_test, y_test))
# knnaccuracy = a
# accuracy['TF-IDF'].append(a)

# # Evaluation
# model = ['LR','DT','RF','KN']
# data = {'model':model,'accuracy':accuracy['TF-IDF']}
# compare_models = pd.DataFrame(data)
# compare_models

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier

# splitting the training and testing part from the data
X_temp, X_test, y_temp, y_test = train_test_split(train, y, test_size=0.2, random_state=0)

# For creating a table of the accuracies in the end
accuracy = {'TF-IDF':[]}

# Logistic Regression
regressor_LR_tf = LogisticRegression(C=1.0,penalty='l2',solver='newton-cg')
regressor_LR_tf.fit(X_temp, y_temp)
y_predict_LR_tf = regressor_LR_tf.predict(X_test)
a=(regressor_LR_tf.score(X_test, y_test))
lrscore = a
accuracy['TF-IDF'].append(a) # Append accuracy to the list
```

```
# Decision Tree

model_DT_tf = DecisionTreeClassifier(criterion='gini', max_depth=2)
model_DT_tf.fit(X_temp, y_temp)
y_predict_DT_tf = model_DT_tf.predict(X_test)
accuracyscoretree = a
accuracy['TF-IDF'].append(a) # Append accuracy to the list

# Gradient Boosting
model_GB_tf = GradientBoostingClassifier(learning_rate= 0.1, max_depth=3, n_estimators= 100,subsample=1.0)
```

```

model_LR_tf = LogisticRegression().fit(X_temp, y_temp)
y_predict_LR_tf = model_LR_tf.predict(X_test)
a=(model_LR_tf.score(X_test,y_test))
gradient = a
accuracy['TF-IDF'].append(a)

# Random Forest
model_RF_tf = RandomForestClassifier(n_estimators= 100, max_features = 'log2')
model_RF_tf.fit(X_temp, y_temp)
y_predict_RF_tf = model_RF_tf.predict(X_test)
a=(model_RF_tf.score(X_test,y_test))
accuracy['RF'].append(a)

# K-Neighbors Classifier
model_KN_tf = KNeighborsClassifier(metric= 'manhattan', n_neighbors= 5, weights= 'distance')
model_KN_tf.fit(X_temp, y_temp)
y_predict_KN_tf = model_KN_tf.predict(X_test)
a=(model_KN_tf.score(X_test,y_test))
knnaccuracy = a
accuracy['KN'].append(a)

# Evaluation
model = [ 'LR', 'DT', 'GB', 'RF', 'KN' ]
data = {'model':model,'accuracy':accuracy['TF-IDF']}
compare_models = pd.DataFrame(data)
compare models

```

	model	accuracy
0	LR	0.764901
1	DT	0.764901
2	GB	0.764768
3	RF	0.804636
4	KN	0.804636

[258]: train

	10	100	20	2023	2d	3d	3d	64	abilities	ability	...	yakuza	yeah	year	years	yes	yet	you	zelda	zero	dy
0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.154254	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...
1505	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.089588	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1506	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1507	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1508	0.000000	0.111382	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1509	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

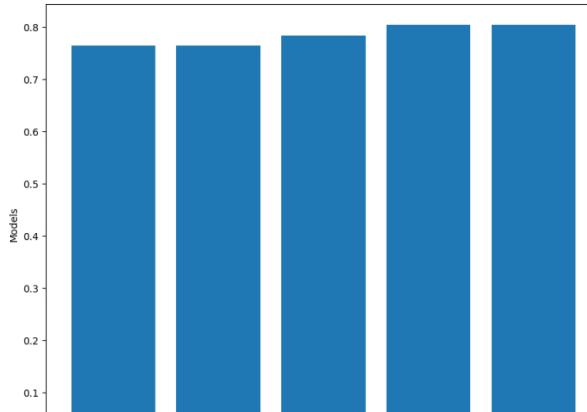
1510 rows × 1000 columns

```

acc=[lrcscore,accuracyscoretree,gradient,accuracyscoreforest,knnaccuracy]
plt.figure(figsize=(10,8))
graph = plt.bar(names,acc)
plt.xlabel('Accuracy')
plt.ylabel('Models')

```

[25]: Text(0, 0.5, 'Models')



... Logistic Regression Decision Tree Classification Gradient Boosting Random Forest Classifier KNN
Accuracy

```

from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import GridSearchCV

```

```

grid={
    'n_neighbors':[3,5,7],
    'weights':['uniform','distance'],
    'algorithm':['auto','ball_tree','kd_tree'],
    'metric':['manhattan','euclidean','minkowski']
}

```

```

gr = GridSearchCV(estimator=KNeighborsClassifier(),param_grid=grid).fit(X_temp,y_temp)
gr = pd.DataFrame(gr.cv_results_)
gr.T

```

```

/opt/conda/lib/python3.7/site-packages/sklearn/model_selection/_split.py:680: UserWarning: The least populated class in y has only 3 members, which is less than n_splits=5.
UserWarning: 
/opt/conda/lib/python3.7/site-packages/sklearn/model_selection/_validation.py:372: FitFailedWarning:
90 fits failed out of a total of 270.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score='raise'.
Below are more details about the failures:
-----
```

```
30 fits failed with the following error:  
Traceback (most recent call last):  
  File "/opt/conda/lib/python3.7/site-packages/sklearn/model_selection/_validation.py", line 680, in _fit_and_score  
    estimator.fit(X_train, y_train, **fit_params),
```

```
self._check_algorithm_metric)
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_base.py", line 378, in _check_algorithm_metric
    "Metric can also be a callable function." % self.metric, alg_check)
ValueError: Metric 'euclidean' not valid. Use metric='euclidean', 'MANHATTAN', 'MINKOWSKI' or 'brute' to get valid options. Metric can also be a callable function.
```

```
30 fits failed with the following error:  
Traceback (most recent call last):  
File "/opt/conda/lib/python3.7/site-packages/sklearn/model_selection/_validation.py", line 688, in _fit_and_score  
    estimator = clone(estimator)  
File "/opt/conda/lib/python3.7/site-packages/sklearn/base.py", line 159, in clone  
    return self._safe_fitt(x)  
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_classification.py", line 198, in fit  
    return self._fit(x)  
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_base.py", line 437, in _fit  
    check_algorithm_metric  
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_base.py", line 378, in _check_algorithm_metric  
    "Metric can also be a callable function." % (self.metric, alg_check)  
ValueError: Metric 'euclidean' not valid. Use sorted(sklearn.neighbors.VALID_METRICS['ball_tree']) to get valid options. Metric can also be a callable function.
```

```
30 fits failed with the following error:
Traceback (most recent call last):
File "/opt/conda/lib/python3.7/site-packages/sklearn/model_selection/_validation.py", line 680, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_classification.py", line 198, in fit
    self._check_algorithm_metric()
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_base.py", line 437, in _fit
    self._check_algorithm_metric()
File "/opt/conda/lib/python3.7/site-packages/sklearn/neighbors/_base.py", line 378, in _check_algorithm_metric
    metric = check_metric(metric)
ValueError: Metric must also be a callable function. % (self.metric, self._check)
Metric can also be a callable function. Use scores(skllearn.neighbors.VALID_METRICS['kd_tree']) to get valid options. Metric can also be a callable function.

warnings.warn(some_fits_failed.message, FitFailedWarning)
nan nan nan nan nan nan nan
0.73778508 0.7798065 0.72350742 0.77897534 0.7268235 0.78229142
0.73675457 0.78311786 0.72350742 0.77897534 0.7268235 0.78311786
0.73778508 0.7798065 0.72350742 0.77897534 0.7268235 0.78229142
0.73675457 0.78311786 0.72764994 0.72764994 0.78311786
0.73778508 0.7798065 0.72350742 0.77897534 0.7268235 0.78229142
0.73675457 0.78311786 0.72764994 0.72764994 0.78311786
0.73778508 0.7798065 0.72350742 0.77897534 0.7268235 0.78229142
0.73675457 0.78311786 0.72764994 0.72764994 0.78311786
```

17 rows × 54 columns

+ Code + Markdown