Data Analysis

Presented by Ziad Hosny

Overview

01 Project Description

02 Introduction to the Dataset

03 Data Manipulation with Python

04 Power Bi Reports

05 KPIs

06 Conclusion

Presented by Ziad Hosny

Project Description

01

Getting familiar with the TMDB Dataset

02

Establishing the goals of this presentation

03

Data Analysis using Python

04

Power Bi Reports

Introduction

The TMDB (The Movie Database) is a comprehensive movie database that provides information about movies, including details like titles, ratings, release dates, revenue, genres, and much more which we will be exploring together. This dataset contains a collection of 1,000,000 movies from the TMDB database.

Here is the Data Card:
https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies/data

Columns

- id: Movie’s Identification Number (Numerical)
- title: Movie’s Title (Categorical)
- vote_average: Average votes of the Movie (Numerical)
- vote_count: Vote Count of the Movie (Numerical)
- status: Is the Movie Released or not (Categorical)
- release_date: Movie’s release data (Datetime)
- revenue: Revenue generated by each Movie (Numerical)
- runtime: Movie’s period of time (Numerical)
- adult: a +18 Movie or not (Categorical)
- budget: The budget taken by each movie to be produced (Numerical)
- homepage: Movie’s homepage link (Categorical)
- imdb_id: Another Identification Number for each Movie (Numerical)
- original_language: Movie’s original language before publication (Categorical)

Columns

- original_title: Movie’s intended title before publication (Categorical)
- overview: A brief description about the Movie (Categorical)
- popularity: Movie’s popularity score (Categorical)
- poster_path: Movie’s poster link (Categorical)
- tagline: Movie’s tagline (Categorical)
- genres: Genres involved in the Movie (Categorical)
- production_companies: Production Companies involved in the making of each Movie (Categorical)
- production_countries: Countries which the Movie was filmed at (Categorical)
- spoken_languages: Languages spoken during the Movie (Categorical)
- keywords: Keywords that distinguish each Movie (Categorical)

Here is a closer look at the dataset

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	title	vote_avera	vote_count	status	release_da	revenue	runtime	adult	backdrop_	budget	homepage	imdb_id	original_la	original_titl	overview	popularity	poster_pat	tagline	genres	production	production	spoken_la	keywords
2	Inception	8.364	34495	Released	########	8.26E+08	148	FALSE	/8ZTVqvKD	1.6E+08	https://ww	tt1375666	en	Inception	Cobb, a ski	83.952	/oYuLEt3z	Your mind	Action, Sci	Legendary	United King	English, Fr	rescue, miss
3	Interstellar	8.417	32571	Released	########	7.02E+08	169	FALSE	/pbrkL804	1.65E+08	http://www	tt0816692	en	Interstellar	The advent	140.241	/gEU2QniE	Mankind w	Adventure,	Legendary	United King	English	rescue, futur
4	The Dark K	8.512	30619	Released	########	1E+09	152	FALSE	/nMKdUUe	1.85E+08	https://ww	tt0468569	en	The Dark K	Batman rai	130.643	/qJ2tW6WI	Welcome t	Drama, Ac	DC Comics	United King	English, Ma	joker, sadism
5	Avatar	7.573	29815	Released	########	2.92E+09	162	FALSE	/vL5LR6W	2.37E+08	https://ww	tt0499549	en	Avatar	In the 22nd	79.932	/kyeqWdyU	Enter the w	Action, Adv	Dune Enter	United Stat	English, Sp	future, socie
6	The Avenge	7.71	29166	Released	########	1.52E+09	143	FALSE	/9BBTo63A	2.2E+08	https://ww	tt0848228	en	The Avenge	When an u	98.082	/RYMX2wc	Some asse	Science Fi	Marvel Stu	United Stat	English, Hi	new york city
7	Deadpool	7.606	28894	Released	2/9/2016	7.83E+08	108	FALSE	/en971ME	58000000	https://ww	tt1431045	en	Deadpool	The origin s	72.735	/zq8Cl3PN	Witness th	Action, Adv	20th Centu	United Stat	English	superhero, a
8	Avengers: I	8.255	27713	Released	########	2.05E+09	149	FALSE	/mDfJG3L	3E+08	https://ww	tt4154756	en	Avengers: I	As the Aven	154.34	/7WsyChQ	An entire u	Adventure,	Marvel Stu	United Stat	English, Xh	sacrifice, ma
9	Fight Club	8.438	27238	Released	########	1.01E+08	139	FALSE	/hZkgoQYu	63000000	http://www	tt0137523	en	Fight Club	A ticking-ti	69.498	/pB8BM7p	Mischief. M	Drama	Regency E	United Stat	English	dual identity,
10	Guardians	7.906	26638	Released	########	7.73E+08	121	FALSE	/uLtVbjvS1	1.7E+08	http://mar	tt2015381	en	Guardians	Light years	33.255	/r7vmZjiyZ	All heroes	Action, Sci	Marvel Stu	United Stat	English	spacecraft, b
11	Pulp Fictio	8.488	25893	Released	########	2.14E+08	154	FALSE	/suaEOtk1	8500000	https://ww	tt0110912	en	Pulp Fictio	A burger-lo	74.862	/d5iIlFn5s	Just becau	Thriller, Cr	Miramax, A	United Stat	English, Sp	drug dealer,
12	Forrest Gu	8.477	25409	Released	########	6.77E+08	142	FALSE	/qdIMHd4s	55000000	https://ww	tt0109830	en	Forrest Gu	A man with	92.693	/arw2vcBv	The world w	Comedy, D	Paramoun	United Stat	English	vietnam war,
13	Harry Potte	7.916	25379	Released	########	9.76E+08	152	FALSE	/hziiv14Op	1.25E+08	https://ww	tt0241527	en	Harry Potte	Harry Potte	185.482	/wuMc08IP	Let the ma	Adventure,	Warner Br	United King	English	witch, schoo
14	Iron Man	7.64	24874	Released	########	5.85E+08	126	FALSE	/cyecB7go	1.4E+08	https://ww	tt0371746	en	Iron Man	After being	72.897	/78lPtwv72	Heroes are	Action, Sci	Marvel Stu	United Stat	English, Pe	middle east,
15	Django Unc	8.171	24672	Released	########	4.25E+08	165	FALSE	/5Lbm0gpI	1E+08	http://www	tt1853728	en	Django Unc	With the he	54.224	/7oWY8VD	Life, liberty	Drama, We	The Weins	United Stat	English, Fr	rescue, frien
16	The Shaws	8.702	24649	Released	########	28341469	142	FALSE	/kXfqcdQK	25000000		tt0111161	en	The Shaws	Framed in	122.61	/lyQBXzOQ	Fear can h	Drama, Cri	Castle Roc	United Stat	English	prison, friend
17	Avengers: I	8.263	23857	Released	########	2.8E+09	181	FALSE	/7RyHsO4y	3.56E+08	https://ww	tt4154796	en	Avengers: I	After the de	91.756	/or06FN3D	Avenge the	Adventure,	Marvel Stu	United Stat	English, Ja	superhero, ti
18	The Matrix	8.206	23815	Released	########	4.64E+08	136	FALSE	/oMsxZEvz	63000000	http://www	tt0133093	en	The Matrix	Set in the 2	78.564	/f89U3ADr	Welcome t	Action, Sci	Village Roa	United Stat	English	man vs mach
19	Titanic	7.9	23637	Released	########	2.26E+09	194	FALSE	/rzdPqYx7U	2E+08	https://ww	tt0120338	en	Titanic	101-year-o	102.348	/9xjZS2rlV	Nothing on	Drama, Ro	Paramoun	United Stat	English, Fr	epic, ship, dr
20	Joker	8.168	23425	Released	########	1.07E+09	122	FALSE	/hO7KbdvG	55000000	http://www	tt7286456	en	Joker	During the	54.522	/udDclJoHJ	Put on a ha	Crime, Thr	Warner Br	Canada, U	English	dream, stree
21	The Lord of	8.402	23323	Released	########	8.71E+08	179	FALSE	/x2RS3uTc	93000000	http://www	tt0120737	en	The Lord of	Young hobl	87.037	/6oom5QY	One ring to	Adventure,	New Line C	New Zeala	English	based on nov
22	The Lord of	8.474	22334	Released	########	1.12E+09	201	FALSE	/2u7zbn8E	94000000	http://www	tt0167260	en	The Lord of	Aragorn is	99.276	/rCzpDGLb	The eye of	Adventure,	New Line C	New Zeala	English	based on nov
23	Shutter Isla	8.2	22318	Released	########	2.95E+08	138	FALSE	/2nqsOT2A	80000000	http://www	tt1130884	en	Shutter Isla	World War	56.595	/4GDy0PH	Some plac	Drama, Th	Phoenix Pi	United Stat	English, Ge	island, based
24	The Wolf of	8.035	22222	Released	########	3.92E+08	180	FALSE	/63y4XSVT	1E+08	http://www	tt0993846	en	The Wolf of	A New York	97.444	/34m2tygA	Earn. Spen	Crime, Dra	EMJAG Pro	United Stat	English, Fr	corruption, d
25	Avengers: A	7.276	21754	Released	########	1.41E+09	141	FALSE	/6YwkGolw	3.65E+08	http://mar	tt2395427	en	Avengers: A	When Tony	96.565	/4ssDuvED	A new age	Action, Adv	Marvel Stu	United Stat	English	artificial inte
26	Captain Am	7.4	21541	Released	########	1.16E+09	147	FALSE	/wdwcOBM	2.5E+08	https://ww	tt3498820	en	Captain Am	Following t	70.741	/rAGiXaUfI	United we	Adventure,	Marvel Stu	United Stat	Romanian,	civil war, sup

Goals

01
Establish detailed insights and analyzing key factors that contribute to a Movie’s success

02
Power Bi Visualizations

03
Movie Recommendation System using NLP (TF-IDF score & Cosine Similarity)

Note

From my perspective, after examining the Dataset, I reached the conclusion that we can derive the factors that contributes to a movie’s success from the tactics taken by the top Movie Production Companies and using the correct mix of genres to achieve international recognition status. Therfore, that is what we will be exploring.

Data Analysis with Python

The approached tactics is as following:

- Clean The Dataset
- Split the Dataset into 3 Datasets for later usage (CleanedDataset, GenresDataset, ComapniesDataset)
- Movie Recommendation System

Cleaning data

```python
# CLEANING DATA

import pandas as pd
import warnings

warnings.filterwarnings('ignore')

df = pd.read_csv( filepath_or_buffer: "C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/TMDB_movie_dataset.csv", parse_dates=['release_date'])
print(f"before anything : {df.shape}")

df = df[df['vote_average'] != 0]
print(f"after dropping (vote average = 0) movies : {df.shape}")
```

```
before anything : (1121725, 24)
after dropping (vote average = 0) movies : (349832, 24)
```

I have dropped the movies with 0 for a vote average, because a movie with no votes is the furthest thing from a successful movie. It has reduced the size of the dataset drastically as it went from over a million rows to 350k rows.

Cleaning data

I dropped some unnecessary columns for our analysis because these features contain links and unnecessary data.

You may notice that I dropped both identification number columns (‘id’, ‘imdb_id’) as I will be using ‘title’ column as a unique identifier for each Movie.

```
df.drop(['backdrop_path',
         'homepage',
         'poster_path',
         'original_title',
         'id',
         'imdb_id',
         'vote_count',
         'original_language',
         'tagline',
         'spoken_languages',
         'production_countries',
         'status'], inplace=True, axis=1)
```

Cleaning data

```
df.drop_duplicates(subset='title', inplace=True)
print(f"after dropping title duplicates : {df.shape}")

df.dropna(inplace=True)
print(f"after dropping null values: {df.shape}")
```

```
after dropping title duplicates : (309880, 12)
after dropping null values: (101258, 12)
```

The shape of the dataset reduced drastically after dropping some of the columns and duplicated & null rows to 100k rows and 12 columns

Cleaning data

```
# COMPANIES
df['production_companies'] = df['production_companies'].apply(lambda x: [company.strip() for company in x.split(',')])
production_companies = df.explode('production_companies')
company_stats = production_companies.groupby('production_companies').agg({
    'revenue': 'sum',
    'popularity': 'sum',
    'vote_average' : 'sum'
})
company_stats['combined_score'] = company_stats['revenue'] + company_stats['popularity'] + company_stats['vote_average']
top_20_companies = company_stats.nlargest(20, 'combined_score')
top_20_company_names = top_20_companies.index.tolist()
for company in top_20_company_names:
    df[company] = df['production_companies'].apply(lambda x: 1 if company in x else 0)
```

I filtered for the top 20 Movie production companies based upon a score made up of the sum of the (popularity, revenue and average votes) for their movies.

Then i applied Dummy Encoding manually for the top 20 movie production companies.

Cleaning data

```
# GENRES
df['genres'] = df['genres'].apply(lambda x: [genre.strip() for genre in x.split(',')])
genres = df.explode('genres')
uniqueGenres = genres['genres'].unique()
for genre in uniqueGenres:
    df[genre] = df['genres'].apply(lambda x: 1 if genre in x else 0)
```

I did the same (applied Dummy Encoding manually) for every unique genre, bringing the columns tally to 51 (12 original columns, The top 20 Movie Production Companies columns, 19 Genres columns).

Cleaning data

```
df.to_csv("C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/CleanedData.csv")
```

Finally, I saved the cleaned data into ‘CleanedData.csv’. And by that I derived the 1st dataset which we will use to derive 2 more datasets which will be used for the Power BI reports.

Deriving the Genres Dataset

```python
import numpy as np
import pandas as pd
import warnings

warnings.filterwarnings('ignore')

df = pd.read_csv( filepath_or_buffer: "C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/CleanedData.csv", parse_dates=['release_date'])
df.drop( labels: ['Unnamed: 0'], inplace=True, axis=1)

start = df.columns.get_loc('Action')
end = df.columns.get_loc('Documentary') + 1

genres = df.iloc[:, start:end]
genreNames = genres.columns.tolist()
genreNames.sort()
```

I imported the ‘CleanedDataset’, located the genres columns names’ and stored them in the ‘genresNames’ variable.

Deriving the Genres Dataset

```
meanOfRevenue = np.mean(df['revenue'])

df['release_date'] = pd.to_datetime(df['release_date'], format='%m/%d/%Y')
df['year'] = df['release_date'].dt.year

profitableGenres = []
meanRatios = []
for name in genreNames:
    genresMean = np.mean(df['revenue'][(df[name] == 1) & ((df['year'] >= 2018) & (df['year'] <= 2022))])
    if genresMean / meanOfRevenue > 1:
        profitableGenres.append(name)
    meanRatios.append((genresMean / meanOfRevenue).round(2))

print(f'The profitable genres are: {profitableGenres}')
```

```
The profitable genres are: ['Action', 'Adventure', 'Animation', 'Comedy', 'Crime', 'Family', 'Fantasy', 'History', 'Mystery', 'Science Fiction', 'Thriller', 'War']
```

I calculated the Profitability Ratio for every Genre from the year 2018 up to 2022.

- Profitability Ration of Genre X = (Revenue Mean of Genre X) / (Revenue Mean of all the Genres)

Deriving the Genres Dataset

```
pro = []
for x in genreNames:
    if x in profitableGenres:
        pro.append(True)
    else:
        pro.append(False)
```

A genre is considered Profitable if it’s Profitability Ratio is >= 1.
So, I added a column called ‘pro’ that contains True if a certain genre is Profitable.

Deriving the Genres Dataset

```
top5Movies = []
for genreName in genreNames:
    movies = df[['title', 'revenue']][df[genreName] == 1].nlargest(5, 'revenue')
    for item in movies['title'].tolist():
        top5Movies.append(item)
```

I filtered for the top 5 Movies for each Genre, according to the revenue generated by that Movie.

Deriving the Genres Dataset

```
rev = []
for name in genreNames:
    rev.append(df['revenue'][(df[name] == 1) & (df['year'] == 2018)].sum())

for name in genreNames:
    rev.append(df['revenue'][(df[name] == 1) & (df['year'] == 2019)].sum())

for name in genreNames:
    rev.append(df['revenue'][(df[name] == 1) & (df['year'] == 2020)].sum())

for name in genreNames:
    rev.append(df['revenue'][(df[name] == 1) & (df['year'] == 2021)].sum())

for name in genreNames:
    rev.append(df['revenue'][(df[name] == 1) & (df['year'] == 2022)].sum())
```

I calculated the sum of revenue for each Genre over the past 5 five years.

Deriving the Genres Dataset

```
data = {
    'names': genreNames * 5,
    'meanRatios': meanRatios * 5,
    'profitable': pro * 5,
    'years' : [2018, 2019, 2020, 2021, 2022] * 19,
    'revenue' : rev
}

genresDataset = pd.DataFrame(data)

genresDataset.sort_values( by: 'names', inplace=True)
genresDataset['top 5 movies'] = top5Movies
genresDataset.to_csv("C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/GenresDataset.csv")
```

And finally, I created the 'GenresDatset' by evening the rows of the lists we created in the past slides and saving the dataset for the Power BI Report.

Power BI Report for the Genres Dataset

Open ‘GenresDatasetReport’.

Deriving the Companies Dataset

```python
import pandas as pd
import warnings

warnings.filterwarnings('ignore')


df = pd.read_csv( filepath_or_buffer: "C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/CleanedData.csv", parse_dates=['release_date'])
df.drop( labels: ['Unnamed: 0'], inplace=True, axis=1)

start = df.columns.get_loc('Warner Bros. Pictures')
end = df.columns.get_loc('Lionsgate') + 1

companyNames = df.iloc[:, start:end].columns.tolist()
```

I imported the ‘CleanedDataset’, located the companies columns names’ and stored them in the ‘companyNames’ variable.

Deriving the Companies Dataset

I filtered for the top 5 Movies for each Company, according to the revenue generated by that Movie, Calculated the mean of the Vote Rates & Run Time for each movie and lastly the percentage of Adult Movies produced by the top 20 Companies.

```
top5Movies = []
for companyName in companyNames:
    movies = df[['title', 'revenue']][df[companyName] == 1].nlargest(5, 'revenue')
    top5Movies.append(movies['title'].tolist())

voteRate = []
for companyName in companyNames:
    voteRate.append(df['vote_average'][df[companyName] == 1].mean().round(2))

avgRuntime = []
for companyName in companyNames:
    avgRuntime.append(df['runtime'][df[companyName] == 1].mean().round(2))

adultPercentage = []
for companyName in companyNames:
    counts = df['adult'][df[companyName] == 1].value_counts()
    percentage = counts.iloc[0] / counts.values.sum() * 100
    adultPercentage.append(percentage)
```

Deriving the Companies Dataset

I located the Genres features to calculate the percentage of each used genre per Company for later visualization.

```python
start = df.columns.get_loc('Action')
end = df.columns.get_loc('Documentary') + 1
genres = df.iloc[:, start:end].columns.tolist()



genresPercentages = []
for companyName in companyNames:
    tempDf = df[df[companyName] == 1]
    onesOfAllGenres = 0
    for x in genres:
        if len(tempDf[x].value_counts().tolist()) == 1:
            continue
        onesOfAllGenres += tempDf[x].value_counts().tolist()[1]
    for genre in genres:
        if len(tempDf[genre].value_counts().tolist()) == 1:
            onesPercentage = 0
            genresPercentages.append(onesPercentage)
            continue
        onesCount = tempDf[genre].value_counts().tolist()[1]
        onesPercentage = (onesCount / onesOfAllGenres) * 100
        genresPercentages.append(onesPercentage)
```

Deriving the Companies Dataset

And finally, I created the ‘CompaniesDataset’ by evening the rows of the lists we created in the past slides and saving for the Power BI Report.

```
data = {
    'names': companyNames * 19,
    'top 5 movies': top5Movies * 19,
    'average vote rate': voteRate * 19,
    'average run time': avgRuntime * 19,
    'adult movies percentage': adultPercentage * 19,
    'genres names': genres * 20,
    'used genres percentages': genresPercentages
}

df = pd.DataFrame(data)
df.sort_values( by: 'names', inplace=True)

df.to_csv("C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/CompaniesDataset.csv")
```

Power BI Report for the Companies Dataset

Open ‘companies’.

Movie Recommendation System

The approached tactics is as following:

- Using the ‘CleanedDataset’, we will be using only 2 features (‘title’, ‘keywords’)
- Calculate the TF-IDF score for each Movie’s keywords
- Calculate the Cosine-Similarity for each score
- The highest ‘num’ scores will be the Recommended Movies

Movie Recommendation System

```
import pandas as pd
import warnings

warnings.filterwarnings('ignore')

df = pd.read_csv( filepath_or_buffer: "C:/Users/ziadh/OneDrive/Desktop/AI/Datasets/CleanedData.csv", parse_dates=['release_date'])
df.drop( labels: ['Unnamed: 0'], inplace=True, axis=1)

df = df[['title', 'keywords']]
df['keywords'] = df['keywords'].str.replace(',', '')

df = df.iloc[0:30000]
```

I imported the 'CleanedDataset' and used only the 'title' & 'keywords' features and limited the new dataset to only 30k rows for storage purposes.

Movie Recommendation System

I calculated the TF-IDF score for the ‘keywords’ feature (avoiding English stop words), calculated the Cosine-Similarity for the TF-IDF score and created a Series called ‘indices’ with the movie titles as it’s indices and the indices of ‘CleanedDataset’ (‘title’ feature) as it’s values.

```
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

englishStopWords = stopwords.words('english')

vectorizer = TfidfVectorizer(stop_words=englishStopWords)
tfidf = vectorizer.fit_transform(df['keywords'])
cosineSim = cosine_similarity(tfidf)
cosineSim = pd.DataFrame(cosineSim)
indices = pd.Series(df.index, index=df['title'])
```

Movie Recommendation System

```
def movieRecommendationSystem(title, cosineSim=cosineSim, num=5):
    i = indices[title]
    indexes = cosineSim[i].nlargest(num+1).index.tolist()
    recommendedMovies = indices[indexes].index.tolist()
    return recommendedMovies[1:num+1]
```

The ‘movieRecommendationSystem’ function takes the title and the cosine similarity matrix and the number of required movies to be recommended as parameters.

The approach is as follows:

- ‘i’ contains the index of the movie title from the Series ‘indices’
- ‘indexes’ contains a list of the indices of the largest ‘num’ + 1 Cosine Similarity scores
- ‘recommendedMovies’ contains a list of the recommended movies titles from the ‘indices’ Series

Movie Recommendation System

And here are a few examples on the system

```python
recommendedMovies = movieRecommendationSystem('Harry Potter and the Philosopher\'s Stone')
for movie in recommendedMovies:
    print(movie)
```

recommendationSystem ×

```
Harry Potter and the Goblet of Fire
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Deathly Hallows: Part 2
Harry Potter and the Half-Blood Prince
Suck Me Shakespeer 2
```

```python
recommendedMovies = movieRecommendationSystem('Avengers: Endgame')
for movie in recommendedMovies:
    print(movie)
```

recommendationSystem ×

```
The Avengers
Ant-Man and the Wasp: Quantumania
Marvel Studios: Assembling a Universe
Guardians of the Galaxy Vol. 2
Doctor Strange in the Multiverse of Madness
```

```python
recommendedMovies = movieRecommendationSystem('Interstellar')
for movie in recommendedMovies:
    print(movie)
```

recommendationSystem ×

```
The Time Guardian
Happy Accidents
Star Wreck: In the Pirkinning
Stowaway
Holiday on Mars
```

KPIs

1- Genres Financial KPIs

- Revenue generated over the past 5 years per genre.
- Profitability Ratio per Genre.

2-Comapnies Strategies KPIs

- Percentage of used Genres per Company.
- Analyzing tactics approached by top status companies like Average Runtime, and Adult Movies Percentage.

3- Movie-Specific KPIs

- 5 movies that reached international recognition status for each genre & company.

4- Movie Recommendation System KPIs

- Diversity of recommended Movies.

Conclusion

There are many ways to analyze data and that was my approach to get the best insights out of the TMDB Dataset as we have learned of the tactics and strategies took by the Top 20 production Companies and that movies are only but a mix of the right strategies & Genres.

Thank You

Presented by Ziad Hosny