1. Business Understanding

Anime Classification

Crunchyroll and Funimation just merged their entire catalouges and now have money to burn, they are looking for the top rated anime for their streaming service and what makes the anime's top rated for their own original animes.

2. Data Understanding

The data im working with for this project comes from this [Kaggle dataset (https://www.kaggle.com/datasets/marlesson/myanimelist-dataset-animes-profiles-reviews)](https://www.kaggle.com/datasets/marlesson/myanimelist-dataset-animes-profiles-reviews) from the year 2020. The target variable for this project is "Score X" which is the overall score for the animes which will be split into multiclass since value counts go from 1 through 10.

The making of the master dataset

In [123]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from imblearn.over_sampling import SMOTE
from imblearn.pipeline import make_pipeline
from imblearn.pipeline import Pipeline as imbpipline
from sklearn.dummy import DummyClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score,\
plot_confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
```

In [2]:
```
reviewsdf = pd.read_csv('data/reviews.csv') # The three datasets
profilesdf = pd.read_csv('data/profiles.csv')
animesdf = pd.read_csv('data/animes.csv')
```

In [3]:
```
reviewsdf.columns #Seeing which columns I can use to combine the datasets
```

Out[3]:
```
Index(['uid', 'profile', 'anime_uid', 'text', 'score', 'scores', 'link'], d
type='object')
```

In [4]:
```
profilesdf.columns
```

Out[4]:
```
Index(['profile', 'gender', 'birthday', 'favorites_anime', 'link'], dtype
='object')
```

In [5]:
```
animesdf.columns
```

Out[5]:
```
Index(['uid', 'title', 'synopsis', 'genre', 'aired', 'episodes', 'members',
       'popularity', 'ranked', 'score', 'img_url', 'link'],
      dtype='object')
```

In [6]:
```
mergedf1 = pd.merge(reviewsdf, animesdf,
                   on='uid',
                   how='outer')
print(mergedf1)
```

...

In [7]:
```
dfmerge = pd.DataFrame(data=mergedf1)
dfmerge
```

Out[7]:

	uid	profile	anime_uid	text	score_x	scores	
0	255938	DesolatePsyche	34096.0	\n \n \n \n ...	8.0	{'Overall': '8', 'Story': '8', 'Animation': '8...	https://myanimelist.ne
1	255938	DesolatePsyche	34096.0	\n \n \n \n ...	8.0	{'Overall': '8', 'Story': '8', 'Animation': '8...	https://myanimelist.ne
2	259117	baekbeans	34599.0	\n \n \n \n ...	10.0	{'Overall': '10', 'Story': '10', 'Animation': ...	https://myanimelist.ne
3	259117	baekbeans	34599.0	\n \n \n \n ...	10.0	{'Overall': '10', 'Story': '10', 'Animation': ...	https://myanimelist.ne
4	253664	skrn	28891.0	\n \n \n \n ...	7.0	{'Overall': '7', 'Story': '7', 'Animation': '9...	https://myanimelist.ne
...	...	...	...	...	...	...	
204577	27829	NaN	NaN	NaN	NaN	NaN	
204578	2649	NaN	NaN	NaN	NaN	NaN	
204579	8676	NaN	NaN	NaN	NaN	NaN	
204580	36043	NaN	NaN	NaN	NaN	NaN	

	uid	profile	anime_uid	text	score_x	scores
204581	33082	NaN	NaN	NaN	NaN	NaN

204582 rows × 18 columns

In [8]:
```
dfmerge.isnull().sum().sum() #Checking for null values
```

Out[8]:
```
2064198
```

In [9]:
```
dfmerge.all()
```

Out[9]:
```
uid              True
profile          True
anime_uid        True
text             True
score_x         False
scores           True
link_x           True
title            True
synopsis         True
genre            True
aired            True
episodes         True
members          True
popularity       True
ranked           True
score_y          True
img_url          True
link_y           True
dtype: bool
```

In [10]:
```
cleandf = dfmerge.dropna()
cleandf #dropping null values
```

episodes	members	popularity	ranked	score_y	img_url	
1.0	360.0	11732.0	8664.0	5.90	https://cdn.myanimelist.net/images/anime/2/705...	https://myanimelis
1.0	360.0	11732.0	8664.0	5.90	https://cdn.myanimelist.net/images/anime/2/705...	https://myanimelis
1.0	100.0	15323.0	12764.0	6.70	https://cdn.myanimelist.net/images/anime/2/745...	https://myanimelist.

In [11]:
```
cleandf.isnull().sum().sum()
```

Out[11]: 0

In [12]:
```
allmerge = pd.merge(cleandf, profilesdf,
                    on='profile',
                    how='outer')
print(allmerge) #repeating the process all over again
```

```
100890          NaN                                                  []
100891          NaN                                                  []
100892          NaN                                                  []

                                                       link
0          https://myanimelist.net/profile/Slushpuppy282 (https://myanimel
ist.net/profile/Slushpuppy282)
1          https://myanimelist.net/profile/Slushpuppy282 (https://myanimel
ist.net/profile/Slushpuppy282)
2          https://myanimelist.net/profile/Slushpuppy282 (https://myanimel
ist.net/profile/Slushpuppy282)
3          https://myanimelist.net/profile/Slushpuppy282 (https://myanimel
ist.net/profile/Slushpuppy282)
4            https://myanimelist.net/profile/ParaParaJMo (https://myanimel
ist.net/profile/ParaParaJMo)
...                                                    ...
100888       https://myanimelist.net/profile/daniel1302 (https://myanimel
ist.net/profile/daniel1302)
100889       https://myanimelist.net/profile/bridgesams (https://myanimel
ist.net/profile/bridgesams)
```

In [13]:
```
allmergedf = pd.DataFrame(data=allmerge)
allmergedf
```

Out[13]:

	uid	profile	anime_uid	text	score_x	scores	
0	29323.0	Slushpuppy282	7588.0	\n \n \n \n ...	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	https://myanimelist.ne
1	29323.0	Slushpuppy282	7588.0	\n \n \n \n ...	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	https://myanimelist.ne
2	29323.0	Slushpuppy282	7588.0	\n \n \n \n ...	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	https://myanimelist.ne
3	29323.0	Slushpuppy282	7588.0	\n \n \n \n ...	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	https://myanimelist.ne
4	30968.0	ParaParaJMo	1253.0	\n \n \n \n ...	9.0	{'Overall': '9', 'Story': '9', 'Animation': '9...	https://myanimelist.ne
...	...	...	...	...	...	...	
100888	NaN	daniel1302	NaN	NaN	NaN	NaN	
100889	NaN	bridgesams	NaN	NaN	NaN	NaN	
100890	NaN	Officer_Anime	NaN	NaN	NaN	NaN	
100891	NaN	Yuez	NaN	NaN	NaN	NaN	
100892	NaN	srry4apologizng	NaN	NaN	NaN	NaN	

100893 rows × 22 columns

In [14]:
```
allmergedf.isnull().sum().sum()
```

Out[14]: 1347096

In [15]:
```
mergeddf = allmergedf.dropna()
mergeddf['scores'].value_counts().sum()
```

Out[15]: 18429

In [16]:
```
mergeddf.isnull().sum().sum()
```

Out[16]: 0

In [17]:
```
#mergeddf
```

...

In [18]:
```
mergeddf.columns
```

Out[18]:
```
Index(['uid', 'profile', 'anime_uid', 'text', 'score_x', 'scores', 'link_
x',
       'title', 'synopsis', 'genre', 'aired', 'episodes', 'members',
       'popularity', 'ranked', 'score_y', 'img_url', 'link_y', 'gender',
       'birthday', 'favorites_anime', 'link'],
      dtype='object')
```

In [19]:
```
#mergeddf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18429 entries, 0 to 25780
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   uid              18429 non-null  float64
 1   profile          18429 non-null  object 
 2   anime_uid        18429 non-null  float64
 3   text             18429 non-null  object 
 4   score_x          18429 non-null  float64
 5   scores           18429 non-null  object 
 6   link_x           18429 non-null  object 
 7   title            18429 non-null  object 
 8   synopsis         18429 non-null  object 
 9   genre            18429 non-null  object 
 10  aired            18429 non-null  object 
 11  episodes         18429 non-null  float64
 12  members          18429 non-null  float64
 13  popularity       18429 non-null  float64
 14  ranked           18429 non-null  float64
 15  score_y          18429 non-null  float64
 16  img_url          18429 non-null  object 
 17  link_y           18429 non-null  object 
 18  gender           18429 non-null  object 
 19  birthday         18429 non-null  object 
 20  favorites_anime  18429 non-null  object 
 21  link             18429 non-null  object 
dtypes: float64(8), object(14)
memory usage: 3.2+ MB
```

In [20]:
```
mergeddf = mergeddf.drop(columns=['link', 'link_y', 'img_url', 'link_x', 'tex
```

In [21]:
```
mergeddf #checking to see if the columns are dropped
```

Out[21]:

	uid	profile	anime_uid	score_x	scores	title	synopsis	
0	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['
1	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['
2	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['
3	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['
4	30968.0	ParaParaJMo	1253.0	9.0	{'Overall': '9', 'Story': '9', 'Animation': '9...	Kokoro no Catchball	An educational anime about the importance of h...	'
...	...	...	...	...	...	...	...	
25775	29103.0	Samurai_Wolf337	8676.0	9.0	{'Overall': '9', 'Story': '9', 'Animation': '9...	Tanoshii Sansuu	Music video for a song about arithmetic by Sei...	['
25776	13167.0	meri_nicole	5060.0	9.0	{'Overall': '9', 'Story': '8', 'Animation': '7...	Zoobles!	The Candy Factory is a place where all Zoobles...	
25777	22745.0	samurai_gaz25	5060.0	10.0	{'Overall': '10', 'Story': '10', 'Animation': ...	Brothers Conflict: Setsubou	Ema finds a special lamp that her father left ...	['H 'Ror 'S
25778	22745.0	samurai_gaz25	5060.0	10.0	{'Overall': '10', 'Story': '10', 'Animation': ...	Brothers Conflict: Setsubou	Ema finds a special lamp that her father left ...	['H 'Ror 'S

	uid	profile	anime_uid	score_x	scores	title	synopsis	
25780	34734.0	Akuteru	2593.0	9.0	{'Overall': '9', 'Story': '10', 'Animation': '...	Minami Kamakura Koukou Joshi Jitenshabu: Kita ...	Unaired episode included with the special edit...	['S 'S 'Sh

18429 rows × 17 columns

Feature Engineering

Using the "Scores" and extracting the scores from story to enjoyment since overall has the same value as the target column that is being used for the classification model

In [22]:
```
mergeddf.scores[0]
```

Out[22]:
```
"{'Overall': '7', 'Story': '7', 'Animation': '6', 'Sound': '9', 'Character': '6', 'Enjoyment': '0'}"
```

In [23]:
```
def scoreExtractor(df):
    Story = [string.split(',')[1].split(':')[1].replace("'",'') for string in
    Animation = [string.split(',')[2].split(':')[1].replace("'",'') for strin
    Sound = [string.split(',')[3].split(':')[1].replace("'",'') for string in
    Character = [string.split(',')[4].split(':')[1].replace("'",'') for strin
    Enjoyment = [string.split(',')[5].split(':')[1].replace("'",'').replace("

    df['Story score'] = pd.Series(Story)
    df['Animation score'] = pd.Series(Animation)
    df['Sound score'] = pd.Series(Sound)
    df['Character score'] = pd.Series(Character)
    df['Enjoyment score'] = pd.Series(Enjoyment)

    return df
```

In [24]:
```
mergeddf = scoreExtractor(mergeddf)
mergeddf.head()
```

Out[24]:

	uid	profile	anime_uid	score_x	scores	title	synopsis	genre	ai
0	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
1	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
2	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
3	29323.0	Slushpuppy282	7588.0	7.0	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
4	30968.0	ParaParaJMo	1253.0	9.0	{'Overall': '9', 'Story': '9', 'Animation': '9...	Kokoro no Catchball	An educational anime about the importance of h...	['Kids', 'Sports']	2

5 rows × 22 columns

In [25]:
```
#mergeddf.info() #seeing that there are missing values in the new columns I m
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18429 entries, 0 to 25780
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   uid              18429 non-null  float64
 1   profile          18429 non-null  object 
 2   anime_uid        18429 non-null  float64
 3   score_x          18429 non-null  float64
 4   scores           18429 non-null  object 
 5   title            18429 non-null  object 
 6   synopsis         18429 non-null  object 
 7   genre            18429 non-null  object 
 8   aired            18429 non-null  object 
 9   episodes         18429 non-null  float64
 10  members          18429 non-null  float64
 11  popularity       18429 non-null  float64
 12  ranked           18429 non-null  float64
 13  score_y          18429 non-null  float64
 14  gender           18429 non-null  object 
 15  birthday         18429 non-null  object 
 16  favorites_anime  18429 non-null  object 
 17  Story score      13359 non-null  object 
 18  Animation score  13359 non-null  object 
 19  Sound score      13359 non-null  object 
 20  Character score  13359 non-null  object 
 21  Enjoyment score  13359 non-null  object 
dtypes: float64(8), object(14)
memory usage: 3.9+ MB
```

Using the median of each column to fill in nan values

In [26]:
```
mergeddf[['Animation score', 'Story score', 'Sound score', 'Character score',
```

In [27]:
```
#Turning the columns into integers from objects to get rid of the float value
mergeddf['Story score'] = mergeddf['Story score'].astype(int)
mergeddf['Animation score'] = mergeddf['Animation score'].astype(int)
mergeddf['Sound score'] = mergeddf['Sound score'].astype(int)
mergeddf['Character score'] = mergeddf['Character score'].astype(int)
mergeddf['Enjoyment score'] = mergeddf['Enjoyment score'].astype(int)
mergeddf['score_y'] = mergeddf['score_y'].astype(int)
```

In [28]:
```
#mergeddf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18429 entries, 0 to 25780
Data columns (total 22 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   uid              18429 non-null  float64
 1   profile          18429 non-null  object 
 2   anime_uid        18429 non-null  float64
 3   score_x          18429 non-null  float64
 4   scores           18429 non-null  object 
 5   title            18429 non-null  object 
 6   synopsis         18429 non-null  object 
 7   genre            18429 non-null  object 
 8   aired            18429 non-null  object 
 9   episodes         18429 non-null  float64
 10  members          18429 non-null  float64
 11  popularity       18429 non-null  float64
 12  ranked           18429 non-null  float64
 13  score_y          18429 non-null  int32  
 14  gender           18429 non-null  object 
 15  birthday         18429 non-null  object 
 16  favorites_anime  18429 non-null  object 
 17  Story score      18429 non-null  int32  
 18  Animation score  18429 non-null  int32  
 19  Sound score      18429 non-null  int32  
 20  Character score  18429 non-null  int32  
 21  Enjoyment score  18429 non-null  int32  
dtypes: float64(7), int32(6), object(9)
memory usage: 3.4+ MB
```

Creating target for model

In [29]:
```
mergeddf['score_x'].value_counts() #Checking the
```

Out[29]:
```
10.0    4684
9.0     4617
8.0     3362
7.0     2358
6.0     1373
5.0      837
4.0      436
3.0      411
2.0      223
1.0      128
Name: score_x, dtype: int64
```

In [30]:
```
mergeddf['score_x'] = mergeddf['score_x'].astype(int)
#Turning the column into integers to get rid of float values
```

In [31]:
```
mergeddf.head()
```

Out[31]:

	uid	profile	anime_uid	score_x	scores	title	synopsis	genre	ai
0	29323.0	Slushpuppy282	7588.0	7	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
1	29323.0	Slushpuppy282	7588.0	7	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
2	29323.0	Slushpuppy282	7588.0	7	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
3	29323.0	Slushpuppy282	7588.0	7	{'Overall': '7', 'Story': '7', 'Animation': '6...	Oyaji no, Imo no Kamisama.	A man wanders into a liquor store and sees a f...	['Slice of Life']	2
4	30968.0	ParaParaJMo	1253.0	9	{'Overall': '9', 'Story': '9', 'Animation': '9...	Kokoro no Catchball	An educational anime about the importance of h...	['Kids', 'Sports']	2

5 rows × 22 columns

Target assignment

- Target 1 for ratings 8 through 10
- Target 2 for ratings 5 through 7
- Target 3 for ratings 1 through 4

In [32]:
```
mergeddf['score_x'].value_counts()
```

Out[32]:
```
10    4684
9     4617
8     3362
7     2358
6     1373
5      837
4      436
3      411
2      223
1      128
Name: score_x, dtype: int64
```

In [33]:
```
mergeddf['target'] = [1 if x >= 8 else 3 if x <= 4 else 2 for x in mergeddf['
```

In [34]:
```
mergeddf['target'].value_counts()
```

Out[34]:
```
1    12663
2     4568
3     1198
Name: target, dtype: int64
```

EDA

now that the data set is cleaned it is the perfect time for EDA

In [35]:
```
mergeddf.describe() #seems animation, character and simply enjoying the anime
```

Out[35]:

	uid	anime_uid	score_x	episodes	members	popularity	
count	18429.000000	18429.000000	18429.000000	18429.000000	1.842900e+04	18429.000000	1
mean	16430.223723	2426.388410	8.018286	13.640241	4.360289e+04	7067.585762	
std	14126.427760	2753.329957	1.952897	39.844532	1.210216e+05	4780.629512	
min	1.000000	1.000000	1.000000	1.000000	2.600000e+01	1.000000	
25%	2460.000000	295.000000	7.000000	1.000000	4.550000e+02	2846.000000	
50%	11275.000000	1535.000000	9.000000	2.000000	3.773000e+03	6442.000000	
75%	31772.000000	3306.000000	10.000000	13.000000	2.511800e+04	11209.000000	
max	40849.000000	10721.000000	10.000000	3057.000000	1.871043e+06	16314.000000	1

In [36]:

```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):
    nunique = df.nunique()
    df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] #
    nRow, nCol = df.shape
    columnNames = list(df)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi =
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('Distribution')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

In [37]:

```
#plotPerColumnDistribution(mergeddf, 25 ,3)
#The average rating looks between 7 through 8
```

...

In [38]:

```
def partial_heatmap(data, start, stop):
    y = data['target']
    df = data.iloc[:, start:stop]
    sns.heatmap(df.corr(), annot=True, fmt='.2f')
    plt.show() #Heat map for correlation
```

In [39]:

```
partial_heatmap(mergeddf, 10, 40)
```

...

In [131]:

```
anime_pop = mergeddf[mergeddf.popularity!=0].sort_values(by='ranked').head(10
anime_pop.head() # Checking the
```

...

In [42]:

```
anime_genre = mergeddf.genre
anime_genre.head(100)
```

...

In [43]:
```
genre_list = []

genre_splited = []

for i in anime_genre.index:
    for j in anime_genre[i].split(", "):
        genre_splited.append(j)
        if j not in genre_list:
            genre_list.append(j)
```

In [44]:
```
genre_list[0:100]
```

...

In [82]:
```
anime_genres_count = pd.Series(genre_splited).value_counts().head(20)

plt.figure(figsize=(15,10))
sns.barplot(x=anime_genres_count.index.tolist(), y=anime_genres_count.tolist(
plt.xlabel('Genres')
plt.ylabel('Anime Count')
plt.title('The Most Popular Genres In The Anime Industry (Regarded all Multi-
plt.xticks(rotation= 100, fontsize=20)
plt.show() #Comedy seems to be the most common genre in the dataset
```

...

Extracting every genre from the genre column

In [47]:
```
for x in mergeddf['genre']:
    print(x.strip('][').replace("'",''))
```

...

In [48]:
```
mergeddf['clean_genre'] = [x.strip('][').replace("'",'') for x in mergeddf['g
```

In [49]:
```
#mergeddf.info()
```

...

In [50]:
```
#list_of_unique_values =' '.join(mergeddf.clean_genre.unique())
#list_of_unique_values = list_of_unique_values.replace(' ',',')
#list_of_unique_genres = list_of_unique_values.split(",")
#list_of_unique_genres = sorted(list(set(list_of_unique_genres)))
#list_of_unique_genres
```

...

In [84]:
```
mergeddf['Action'] = [1 if 'Action' in genre else 0 for genre in mergeddf['ge
mergeddf['Adventure'] = [1 if 'Adventure' in genre else 0 for genre in mergec
mergeddf['Ai'] = [1 if 'Ai' in genre else 0 for genre in mergeddf['genre']]
mergeddf['Arts'] = [1 if 'Arts' or 'Martial' in genre else 0 for genre in mer
mergeddf['Cars'] = [1 if 'Cars' in genre else 0 for genre in mergeddf['genre'
mergeddf['Comedy'] = [1 if 'Comedy' in genre else 0 for genre in mergeddf['ge
mergeddf['Dementia'] = [1 if 'Dementia' in genre else 0 for genre in mergeddf
mergeddf['Demons'] = [1 if 'Demons' in genre else 0 for genre in mergeddf['ge
mergeddf['Drama'] = [1 if 'Drama' in genre else 0 for genre in mergeddf['genr
mergeddf['Ecchi'] = [1 if 'Ecchi' in genre else 0 for genre in mergeddf['genr
mergeddf['Fantasy'] = [1 if 'Fantasy' in genre else 0 for genre in mergeddf['
mergeddf['Game'] = [1 if 'Game' in genre else 0 for genre in mergeddf['genre'
mergeddf['Harem'] = [1 if 'Harem' in genre else 0 for genre in mergeddf['genr
mergeddf['Historical'] = [1 if 'Historical' in genre else 0 for genre in merg
mergeddf['Horror'] = [1 if 'Horror' in genre else 0 for genre in mergeddf['ge
mergeddf['Josei'] = [1 if 'Josei' in genre else 0 for genre in mergeddf['genr
mergeddf['Kids'] = [1 if 'Kids' in genre else 0 for genre in mergeddf['genre'
mergeddf['Life'] = [1 if 'Life' or 'Slice' or 'of' in genre else 0 for genre
mergeddf['Magic'] = [1 if 'Magic' in genre else 0 for genre in mergeddf['genr
mergeddf['Mecha'] = [1 if 'Mecha' in genre else 0 for genre in mergeddf['genr
mergeddf['Military'] = [1 if 'Military' in genre else 0 for genre in mergeddf
mergeddf['Music'] = [1 if 'Music' in genre else 0 for genre in mergeddf['genr
mergeddf['Mystery'] = [1 if 'Mystery' in genre else 0 for genre in mergeddf['
mergeddf['Parody'] = [1 if 'Parody' in genre else 0 for genre in mergeddf['ge
mergeddf['Police'] = [1 if 'Police' in genre else 0 for genre in mergeddf['ge
mergeddf['Power'] = [1 if 'Power' or 'Super' in genre else 0 for genre in mer
mergeddf['Psychological'] = [1 if 'Psychological' in genre else 0 for genre i
mergeddf['Romance'] = [1 if 'Romance' in genre else 0 for genre in mergeddf['
mergeddf['Samurai'] = [1 if 'Samurai' in genre else 0 for genre in mergeddf['
mergeddf['School'] = [1 if 'School' in genre else 0 for genre in mergeddf['ge
mergeddf['Sci-Fi'] = [1 if 'Sci-Fi' in genre else 0 for genre in mergeddf['ge
mergeddf['Seinen'] = [1 if 'Seinen' in genre else 0 for genre in mergeddf['ge
mergeddf['Shouju'] = [1 if 'Shouju' in genre else 0 for genre in mergeddf['ge
mergeddf['Shounen'] = [1 if 'Shounen' in genre else 0 for genre in mergeddf['
mergeddf['Space'] = [1 if 'Space' in genre else 0 for genre in mergeddf['genr
mergeddf['Sports'] = [1 if 'Sports' in genre else 0 for genre in mergeddf['ge
mergeddf['Supernatural'] = [1 if 'Supernatural' in genre else 0 for genre in
mergeddf['Thriller'] = [1 if 'Thriller' in genre else 0 for genre in mergeddf
mergeddf['Vampire'] = [1 if 'Vampire' in genre else 0 for genre in mergeddf['
```

In [165]:
```
#mergeddf.info() #checking to see if the values have any nulls
```

...

OHE and SMOTE

one hot encoding object columns and using smote for the imbalanace of the target column

In [53]:

```
# Defined a OneHotEncoder function for ease of access
def OHE(X_train, categories):
    onehot = OneHotEncoder(sparse=False, handle_unknown = 'ignore')
    x_train_cat = pd.DataFrame(onehot.fit_transform(X_train[categories]))
    x_train_cat.columns = onehot.get_feature_names(categories)

    # Reset indices to avoid merging conflicts
    x_train_cat.reset_index(drop=True, inplace=True)
    X_train.reset_index(drop=True, inplace=True)

    # Joined the OHE dataframe to the dataframe that is passed into the funct
    x_train_df = X_train.drop(categories, axis = 1).join(x_train_cat)
    return x_train_df
def confusion_and_metrics(model, X_test, y_test):
    # Accuracy Score
    print(f"Accuracy Score: {model.score(X_test, y_test):.3f}")


    # Plot confusion matrix for visualization
    plot_confusion_matrix(model, X_test, y_test);
# Defined a function to take in column name and output log-odds coefficient a
def print_odds(dataframe, column_name):
    # Prints out the name of the column and it's original log_odds value
    print(f"{column_name}: {dataframe[column_name][0]}")

    # Prints out the odds value of the column
    print(f"Odds: {np.exp(dataframe[column_name][0])}")
```

In [54]:

```
X = mergeddf.drop(['target', 'profile','clean_genre', 'birthday','uid', 'genr
y = mergeddf['target']
```

In [55]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, rar
```

In [56]:
```
# Called the OHE function we made and assigned new dataframe variables
train_ohe = OHE(X_train, ['gender'])
test_ohe = OHE(X_test, ['gender'])
train_ohe
```

Out[56]:

	episodes	members	popularity	ranked	score_y	Story score	Animation score	Sound score	Character score
0	23.0	98110.0	1115.0	2921.0	7	10	8	9	9
1	12.0	2611.0	7330.0	8649.0	5	8	9	8	9
2	6.0	22001.0	3043.0	3668.0	7	8	8	8	7
3	1.0	176401.0	619.0	459.0	8	9	10	10	10
4	14.0	2425.0	7480.0	3872.0	6	8	9	8	9
...	...	...	...	...	...	...	...	...	...
13816	72.0	2148.0	7834.0	637.0	7	8	8	7	9
13817	1.0	200.0	13386.0	11640.0	5	10	10	10	10
13818	1.0	243.0	12779.0	9733.0	5	8	7	9	7
13819	1.0	77.0	15805.0	14574.0	5	7	10	9	7
13820	1.0	50882.0	1881.0	854.0	7	10	10	8	9

13821 rows × 53 columns

In [57]:
```
train_ohe.columns
```

Out[57]:
```
Index(['episodes', 'members', 'popularity', 'ranked', 'score_y', 'Story sco
re',
       'Animation score', 'Sound score', 'Character score', 'Enjoyment scor
e',
       'Action', 'Adventure', 'Ai', 'Arts', 'Cars', 'Comedy', 'Dementia',
       'Demons', 'Drama', 'Ecchi', 'Fantasy', 'Game', 'Harem', 'Historica
l',
       'Horror', 'Josei', 'Kids', 'Life', 'Magic', 'Martial', 'Mecha',
       'Military', 'Music', 'Mystery', 'Parody', 'Police', 'Power',
       'Psychological', 'Romance', 'Samurai', 'School', 'Sci-Fi', 'Seinen',
       'Shouju', 'Shounen', 'Space', 'Sports', 'Supernatural', 'Thriller',
       'Vampire', 'gender_Female', 'gender_Male', 'gender_Non-Binary'],
      dtype='object')
```

In [59]:
```
y_test.value_counts() #class imbalance
```

Out[59]:
```
1    3174
2    1143
3     291
Name: target, dtype: int64
```

In [60]:
```
from imblearn.under_sampling import RandomUnderSampler
from collections import Counter
from imblearn.over_sampling import SMOTE

smote=SMOTE('not minority')
X_smote, y_smote = smote.fit_sample(train_ohe, y_train)
X_test_smote, y_test_smote = smote.fit_sample(test_ohe, y_test)

counter = Counter(y_train)
test_counter = Counter(y_test_smote)
print(counter)
print(test_counter) #SMOTING for class imbalance using
```

```
C:\Users\ilene\anaconda3\envs\learn-env\lib\site-packages\imblearn\utils\_v
alidation.py:635: FutureWarning: Pass sampling_strategy=not minority as key
word args. From version 0.9 passing these as positional arguments will resu
lt in an error
  warnings.warn("Pass {} as keyword args. From version 0.9 "

Counter({1: 9489, 2: 3425, 3: 907})
Counter({1: 3174, 2: 3174, 3: 291})
```

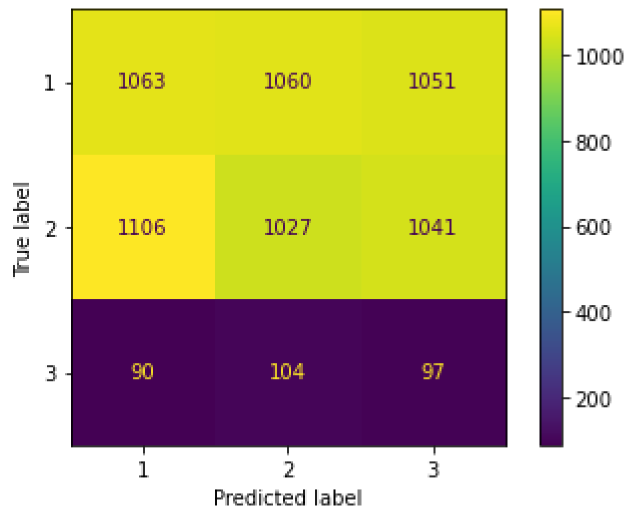
Dummy Model

In [61]:
```
# Created Dummy Classifier model to look at simple accuracy score
from sklearn.dummy import DummyClassifier
dummy = DummyClassifier(strategy='uniform')
dummy.fit(X_train, y_train)
y_pred = dummy.predict(X_train)
y_test_pred = dummy.predict(X_test)
y_pred_df = pd.DataFrame(y_pred)
dummy.score(X_test, y_test)
```

Out[61]:
```
0.3352864583333333
```

In [62]:
```
confusion_and_metrics(dummy, X_test_smote, y_test_smote) #Accuracy score from
```

```
Accuracy Score: 0.326
```



Accuracy score printed out is **32%** which isn't the best we can do

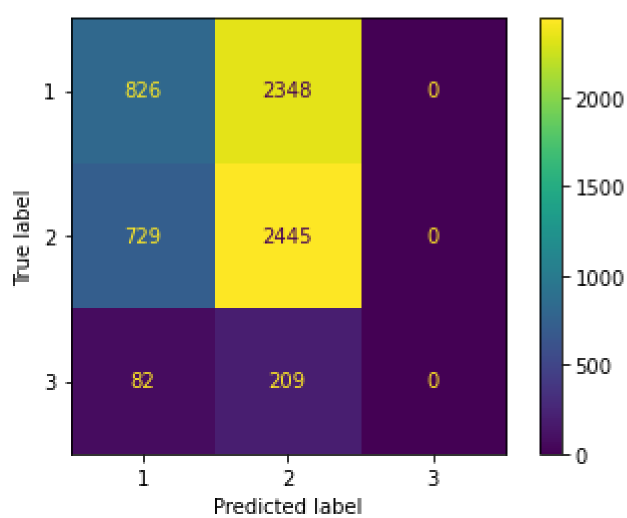
First model

In [167]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(random_state=42, max_iter=1000,multi_class='multir
model.fit(X_smote, y_smote)
model.score(X_test_smote, y_test_smote)
```

Out[167]:
```
0.49269468293417684
```

In [168]:
```
confusion_and_metrics(model, X_test_smote, y_test_smote)
```

```
Accuracy Score: 0.493
```



The Logistic Regression accuracy score went up **17%** from the dummy classifier

Second model

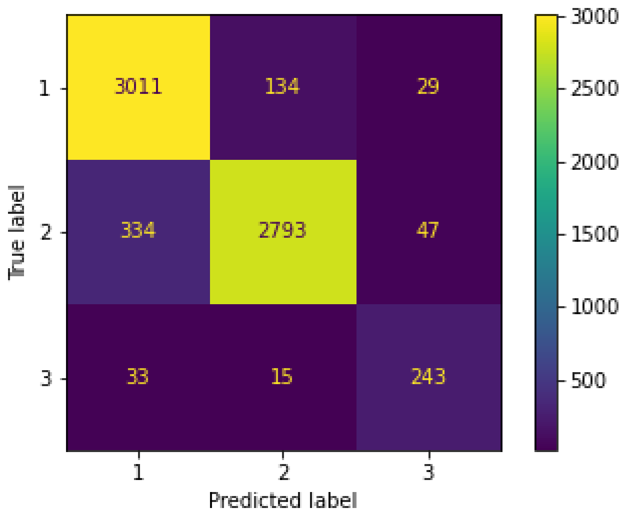
In [169]:
```
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.model_selection import GridSearchCV

dt = DecisionTreeClassifier (random_state = 10)
dt.fit(X_smote, y_smote)
dt.score(X_test_smote, y_test_smote)
```

Out[169]:
```
0.9108299442687152
```

In [170]:
```
confusion_and_metrics(dt, X_test_smote, y_test_smote)
```

```
Accuracy Score: 0.911
```



The best score so far from a decision tree being **91%**

In [71]:
```
# Feature columns
X = X_smote
# Target column - H1N1 Knowledge
y = y_smote

from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt

# Instantiate model
modelfeatures = ExtraTreesClassifier()
modelfeatures.fit(X,y)
print(modelfeatures.feature_importances_) # use built in class 'feature_impor
# Plot graph of feature importances for better visualization
feat_importances = pd.Series(modelfeatures.feature_importances_, index=X.colu
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
```

...

In [110]:
```
df_new = mergeddf[['Action', 'target']]
df_new.value_counts() #Looking at the distribution of the action genre and th
```

Out[110]:
```
Action  target
0       1         9364
        2         3518
1       1         3299
        2         1050
0       3          942
1       3          256
dtype: int64
```

In [112]:
```
plt.figure(figsize=(15,10))
sns.barplot(x=df_new['target'], y=df_new['Action'])
plt.xlabel('Genres')
plt.ylabel('Anime Count')
plt.title('The Most Popular Genres In The Anime Industry (Regarded all Multi-
plt.xticks(rotation= 100, fontsize=20)
plt.show() #graphing the distribution to see the overall ratings are
```

...

In [114]:
```
plt.figure(figsize=(15,10))
sns.barplot(x=mergeddf['target'], y=mergeddf['gender'])
plt.xlabel('Genres')
plt.ylabel('Anime Count')
plt.title('The Most Popular Genres In The Anime Industry (Regarded all Multi-
plt.xticks(rotation= 100, fontsize=20)
plt.show() #Looking at the scores distributed by gender
```

...

In [74]:
```
import datetime
```

In [75]:
```
#Using the Birthday column to get the age of the user ratings
mergeddf['birthday'] = pd.to_datetime(mergeddf['birthday'], errors = 'coerce'
mergeddf=mergeddf[mergedd#.notn]

birth_date = mergeddf.birthday
gender = mergeddf.gender
#spent = users.user_days_spent_watching

age = []
for each in birth_date:
    age.append(round((datetime.datetime.now()-each).days/365.25,1))
```

In [76]:
```
mergeddf['age'] = age #appears to be missing values
#mergeddf.info()
```

...

In [166]:

```
mergeddf[['age']] = mergeddf[['age']].fillna(mergeddf[['age']].median()).asty
mergeddf['age'].value_counts() #Using fillna to replace it with the median
#also seems to be extreme outliers
```

...

In [121]:

```
genres_with_comedy = [] #Comedy being the most relevant genre in anime seeing

for i in anime_genre.index:
    if anime_genre[i].find('Comedy') > -1:
        for j in anime_genre[i].split(", "):
            if j != 'Comedy':
                genres_with_comedy.append(j)
```

In [124]:

```
genres_with_comedy_count = pd.Series(genres_with_comedy).value_counts().

fig = {
  "data": [
    {
      "values": genres_with_comedy_count.tolist(),
      "labels": genres_with_comedy_count.index.tolist(),
      "domain": {"x": [0, .8]},
      "name": "Number Of Students Rates",
      "hoverinfo":"label+percent+name",
      "hole": .4,
      "type": "pie"
    },],
  "layout": {
        "title":"Top 10 Multi-label Tags With Comedy"
    }
}
iplot(fig)
```

...

Conclusion

Overall the best model was the Decision tree classifier it had the best accuracy score it correctly identified ratings being from ranges **1-4 for 3 being not a good rating**, **5-7 being average scoring**, and **8-10 being a great rating**. The logistic regression only correctly identifying the target **49%** of the time. For more accurate scores and using different evaluation methdos more feature engingeering can be done and more predictors for the logistic regression since it can be even more powerful.

Recommendations

I recommend that Crunchyroll:

- To license popular animes that their competitors don't have the rights to
- When making their own original animes to focus on the soundtrack
- To start off with comedy multi labels for the genre for their original animes

Future considerations

Next steps to do for Crunchyroll

- Look into ROI when buying the rights to the animes
- Model NLP for sentiment analysis on twitter

In [ ]: