MMA 2021S 865, Individual Assignment 1

Version 2: Updated Janurary 13, 2021.

- [Adaure, Ibe]
- [20254264]
- [Section 2]
- [The Road Less Travelled]
- [17/10/2021]

Question 1 - ELI5

"If you can't explain it simply, you don't understand it well enough." – Albert Einstein

Explaining technical concepts to a non-technical audience is an underappreciated skill; one which the MMA program aims to give its students; and one that will truly set you apart in the job market. The only way to gain a skill is by practice, so here we go.

Answer each question below as though you were talking to a 5 year old (equivalently: a grandma, or a completely non-technical manager, or an Ivey grad). Use your own words. Use analogies where possible. Examples are better than theory. Keep it short, but be complete. Use simple, plain English. Do not use business buzzwords like *actualize, empower, fungible, leverage, or synergize*. Do not use technical buzzwords that most people don't know like *model, agile, bandwidth, IoT, blockchain, AR, VR, actionable insights*. Inform the audience without going into too much technical detail, and without embarrassing yourself. Your goal is to truly help them understand, not to give what you feel is a "technically precise" answer and move on (but they still don't understand!). Don't be that guy!

Please keep each answer to 1000 characters or less.

Finally, feel free to use Markdown syntax (https://www.markdownguide.org/basic-syntax/) to format your answer.

Part 1: What is "Big Data" and how is it different than "regular data"?

We all use smartphones. Consider all the data that these smartphones generate, including but not limited to photos, texts, emails, games, social media, searches and music, to mention a few. The average smartphone user generates about 40 Exabytes of Data monthly. Now, imagine that multiplied by all the smartphone users in the world. The average computer system can not handle that amount of data, hence the term Big Data due to its size.

The main distinction between Big Data versus regular data includes the following: Volume, Velocity, Variety.

We will examine the meaning of Big Data characteristics from the point of view of the healthcare industry.

Volume: Imagine how much information the healthcare system collects on the average patient ranging from previous patient records to test results. An average hospital generates approximately 2,314 Exabytes annually. Regular data does come in such a large amount.

Velocity: Patient records and test results are generated at a very high speed, particularly in real-time. The medical staff who handle patients are generating data at high speed when they connect with their patients. This is not a characteristic that is seen in regular data as it is mainly collected in the absence of real-time.

Variety: In a hospital setting, data can come in the form of traditional rows and columns (excel sheets). It can also come through visual images and audio. Regular data is typically structured and usually comes in rows and columns.

Part 2: What is Hadoop? Hint: What problems in previous data storage and processing was Hadoop designed to solve? How did Hadoop accomplish that?

Before the development of Hadoop, traditional data storage/management systems did not meet the capacity and speed requirements. As a result, Hadoop was created to cater to those needs.

Hadoop is software that is freely available to the public. It uses a distributed file system: If you have a huge file, your file will be broken down into smaller chunks and stored in various machines. Additionally, when you break the file into a smaller piece, you also make copies of it which goes into different nodes. As a result, if one machine fails, your data will be safe on another.

Hadoop also uses a map-reduce technique where Task A is shared into smaller tasks, namely B, C and D. Instead of one machine, the functions from B, C, and D are sent to three different devices that work on it at the same time. This increases the processing time.

Part 3: How does Big Data and the cloud help Machine Learning?

Machine learning algorithms can learn data patterns better if trained on a continuously growing dataset. Big Data allows this to happen as the volume, velocity and variety of its nature increases the richness of the dataset these ML are trained on.

When Big Data is combined with Machine Learning, we can look forward to seeing more precise results. The combination allows the machines to discover hidden patterns and analytics that ordinarily would not have been caught on regular data.

The cloud provides storage capacity and increased computing power when Big Data is combined with ML models. The need for this is because Big Data can not be stored and processed on local machines. The cloud allows for Big Data to be combined with machine learning; otherwise, it would almost be impossible to combine both on an average computer.

Ultimately, both Big Data and the Cloud aid in predictive modelling processes and outcomes.

Part 4: What is NoSQL?

No SQL database is not a SQL database. SQL databases are relational and come in the form of an excel sheet. However, a NoSQL Database is a non-relational database. Its method of storage does not come in the traditional form of rows and columns. Instead, non-relational databases store data based on the best storage option for the type of data being stored. These storage options could vary from the traditional rows and columns to time series and graphs, to name a few.

Part 5: Name three ways topic modeling could help a bank.

Customer Service Prioritization: this can help banks identify the topics amongst customer reviews with the highest frequency. Let's assume that the data in question is based on transcripts from a client's conversation with a CSR. Topic modelling can be used to identify inquiries/complaints with the highest frequency. The bank can then use this insight as a guide to prioritize treatment/resolution for the topics with the most increased occurrence or consequences if left unresolved.

Customer Service Redirection: Quite a number of businesses have transitioned into using chat boxes as the first step of getting their issues resolved. Based on the conversation between the chatbot and customer, topic modelling can be used to transfer customers who insist on speaking with a customer service rep to the appropriate CSR rep trained to handle their current concern. This can increase overall customer satisfaction. The customers are connected to specialized CSR's who can resolve their issues effectively.

Business Development Research: Banks being businesses, are typically looking for ways to improve their services. They can use topic modelling to gain insight based on customer experience on what its SWOT's are. They can then use this information to drive business strategy to ensure that the core needs of its customers are being met.

Part 6: What is Apache Spark, exactly, and what are its pros and cons?

Apache Spark is a unified analytics engine that is open to the public. It is used for data processing on a large scale. It offers a platform for programming entire clusters with the ability to continue operating despite system failures or malfunctions while distributing the data across different nodes, which operate on the data in parallel.

Pros:

It makes processing huge data sets possible. It handles these data sets in a reasonably quick manner.

It does a pretty good job implementing machine learning models for larger data sets.

It appears to be a rapidly advancing software. The new features make the software even more straightforward to use.

Cons:

It requires some advanced ability to understand and structure the modelling of big data.

The graphics produced by Apache Spark are not very advanced and require further development.

It takes an enormous amount of time to crunch through multiple nodes across huge data sets.

Question 2: Sentiment Analysis via the ML-based approach

Download the “Product Sentiment” dataset from the course portal: sentiment_train.csv and sentiment_test.csv.

Part 1.a. Loading and Prep

Load, clean, and preprocess the data as you find necessary.

In [1]:
```
import pandas as pd
# TODO: import other libraries as necessary

df_train = pd.read_csv(r'C:\Users\ibead\Desktop\MMA 823\sentiment_train.csv')

print(df_train.info())
print(df_train.head())

df_test = pd.read_csv(r'C:\Users\ibead\Desktop\MMA 823\sentiment_test.csv')

print(df_train.info())
print(df_train.head())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2400 entries, 0 to 2399
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Sentence  2400 non-null   object
 1   Polarity  2400 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 37.6+ KB
None
                                            Sentence  Polarity
0                           Wow... Loved this place.         1
1                                 Crust is not good.         0
2          Not tasty and the texture was just nasty.         0
3  Stopped by during the late May bank holiday of...         1
4  The selection on the menu was great and so wer...         1
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2400 entries, 0 to 2399
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Sentence  2400 non-null   object
 1   Polarity  2400 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 37.6+ KB
None
                                            Sentence  Polarity
0                           Wow... Loved this place.         1
1                                 Crust is not good.         0
2          Not tasty and the texture was just nasty.         0
3  Stopped by during the late May bank holiday of...         1
4  The selection on the menu was great and so wer...         1
```

In [2]:
```
#Loading Packages and Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

import re
import string
from wordcloud import WordCloud

import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.stem.wordnet import WordNetLemmatizer

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer

from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import SGDClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_
```

In [3]:
```
#combine DF's for EDA

df_combined = pd.concat([df_train, df_test])
```

In [4]:
```
#reviewing variables
df_combined.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3000 entries, 0 to 599
Data columns (total 2 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Sentence  3000 non-null   object
 1   Polarity  3000 non-null   int64 
dtypes: int64(1), object(1)
memory usage: 70.3+ KB
```

In [5]:
```
#type of variables
df_combined.dtypes
```

Out[5]:
```
Sentence    object
Polarity     int64
dtype: object
```

In [6]:
```
df_combined['Sentence'][1]
```

Out[6]:
```
1                                    Crust is not good.
1    For people who are first timers in film making...
Name: Sentence, dtype: object
```

In [7]:
```
#Checking for Missing Data
df_combined.isnull().sum()
```

Out[7]:
```
Sentence    0
Polarity    0
dtype: int64
```

In [8]:
```
#Checking for Shape
df_combined.shape
```

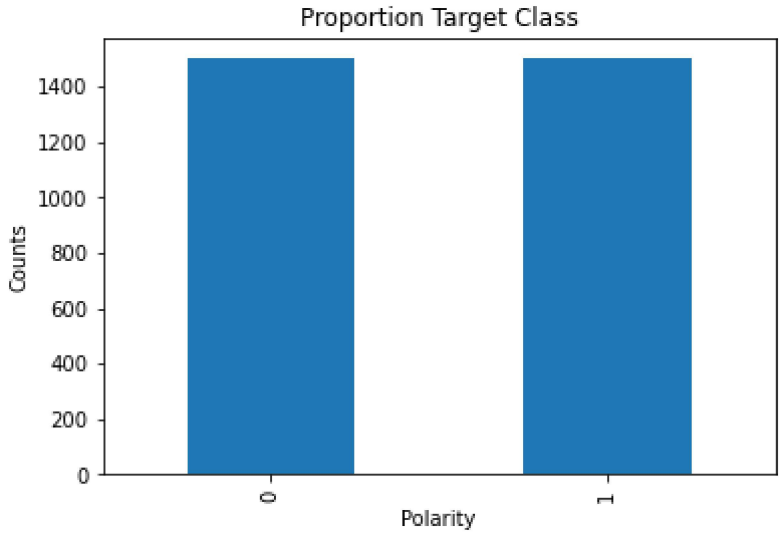
Out[8]:
```
(3000, 2)
```

In [9]:
```
#Checking for imbalance
df_combined['Polarity'].value_counts()
```

Out[9]:
```
0    1500
1    1500
Name: Polarity, dtype: int64
```

In [10]:
```
df_combined["Polarity"].value_counts().plot(kind='bar')
plt.xlabel("Polarity")
plt.ylabel("Counts")
plt.title("Proportion Target Class")
```

Out[10]:
```
Text(0.5, 1.0, 'Proportion Target Class')
```



Data Processing

In [11]:
```
#Tokenization
def tokens(words):
    words = re.sub("[^a-zA-Z]"," ", words)
    text = words.lower().split()
    return " ".join(text)
```

In [12]:
```
df_train['Sentence'] = df_train['Sentence'].apply(tokens)
df_train.head()
```

Out[12]:

	Sentence	Polarity
0	wow loved this place	1
1	crust is not good	0
2	not tasty and the texture was just nasty	0
3	stopped by during the late may bank holiday of...	1
4	the selection on the menu was great and so wer...	1

In [13]:
```
#Convert to string
df_train['Sentence'] = df_train['Sentence'].astype(str)
```

In [14]:
```
#Removing Stop Words
stop_words = stopwords.words('english')
print(stop_words[::10])
```

```
['i', "you've", 'himself', 'they', 'that', 'been', 'a', 'while', 'through',
'in', 'here', 'few', 'own', 'just', 're', 'doesn', 'ma', "shouldn't"]
```

In [15]:
```
def stopwords(review):
    text = [word.lower() for word in review.split() if word.lower() not in st
    return " ".join(text)
```

In [16]:
```
df_train['Sentence'] = df_train['Sentence'].apply(stopwords)
df_train.head()
```

Out[16]:

	Sentence	Polarity
0	wow loved place	1
1	crust good	0
2	tasty texture nasty	0
3	stopped late may bank holiday rick steve recom...	1
4	selection menu great prices	1

In [17]:
```
#Removing Numbers if any

df_train['Sentence'][267]
```

Out[17]: 'thus far visited twice food absolutely delicious time'

In [18]:
```
def numbers(text):
    new_text = []
    for word in text.split():
        if not re.search('\d', word):
            new_text.append(word)
    return ' '.join(new_text)
```

In [19]:
```
df_train['Sentence'] = df_train['Sentence'].apply(numbers)
df_train.head()
```

Out[19]:

	Sentence	Polarity
0	wow loved place	1
1	crust good	0
2	tasty texture nasty	0
3	stopped late may bank holiday rick steve recom...	1
4	selection menu great prices	1

In [20]:
```
df_train['Sentence'][267]
```

Out[20]: 'thus far visited twice food absolutely delicious time'

In [21]:
```
#Lemmatizer
lem = WordNetLemmatizer()

def lemma(text):
    lem_text = [lem.lemmatize(word) for word in text.split()]
    return " ".join(lem_text)
```

Part 1.b. Modeling

Use your favorite ML algorithm to train a classification model. Don't forget everything that we've learned in our ML course: hyperparameter tuning, cross validation, handling imbalanced data, etc. Make reasonable decisions and try to create the best-performing classifier that you can.

In [22]:
```
#Splitting Dataset
X = df_train['Sentence']
y = df_train['Polarity']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rand
```

In [23]:
```
#Vectorization

vect = CountVectorizer(min_df=5, ngram_range=(1,2)).fit(X_train)

X_train_vectorized = vect.transform(X_train)

len(vect.get_feature_names())
```

Out[23]: 519

In [24]:
```
model = LogisticRegression()
model.fit(X_train_vectorized, y_train)

ytest = np.array(y_test)
predictions = model.predict(vect.transform(X_test))
```

Part 1.c. Assessing

Use the testing data to measure the accuracy and F1-score of your model.

In [25]:
```
# Setting X and Y
X_test_df = df_test['Sentence']
y_test_df = df_test['Polarity']

X_train_df, X_test_df, y_train_df, y_test_df = train_test_split(X, y, test_si
```

In [26]:
```
model2 = LogisticRegression()
model2.fit(X_train_vectorized, y_train)

ytest2 = np.array(y_test_df)
predictions = model.predict(vect.transform(X_test_df))
```

In [27]:
```
print('accuracy %s' % accuracy_score(predictions, y_test_df))
print(classification_report(ytest, predictions))
```

```
accuracy 0.7729166666666667
              precision    recall  f1-score   support

           0       0.74      0.84      0.79       243
           1       0.81      0.70      0.75       237

    accuracy                           0.77       480
   macro avg       0.78      0.77      0.77       480
weighted avg       0.78      0.77      0.77       480
```

Part 2. Given the accuracy and F1-score of your model, are you satisfied with the results, from a business point of view? Explain.

The f1 and accuracy score are above average. Therefore this model's ability to classify the reviews is better than guessing, so I'm relatively satisfied. From a business perspective, this will allow one to refer to the reviews to make accommodations or adjustments to future customer experiences, however, with a grain of salt.

The model can serve as a guideline; however, to be sure of my customers' true sentiments about the service, I will supplement its findings by creating a survey that can provide context to the reviews.

Part 3. Show five example instances in which your model's predictions were incorrect. Describe why you think the model was wrong. Don't just guess: dig deep to figure out the root cause.

16 - I would have casted her in that role after ready the script.(Reasoning: I think there aren't any critical words in this sentence that indicate any sentiments. As a result, I believe my model took a random guess and predicted this as negative as opposed to positive. Even as a reader, this sentence comes across as neutral to me. It's not very clear what direction this review is pointing to.)

510- This is one of the worst Sandra Bullock movie since Speed 2 But not quite that bad. (Reasoning: I think that this sentence is somewhat contradictory because it includes positive sentiments and negative sentiments, nonetheless, my model indicated that the positive sentiment "not quite that bad" was the primary influencer and, as a result, predicted that the review was positive when in actuality it was negative.)

584- It's a sad movie, but very good.(Reasoning: I think that this sentence is somewhat contradictory to the ML model (not to the human mind) because it includes positive sentiments and negative sentiments, nonetheless, my model indicated that the negative sentiment "sad" was the primary influencer an, as a result, predicted that the review was negative when in actuality it was positive.)

600 -Exceptionally bad! (Reasoning: I think that this sentence is somewhat contradictory to the ML model (not to the human mind) because it includes positive sentiments and negative sentiments, nonetheless, my model indicated that the negative sentiment "Exceptionally" was the primary influencer an, as a result, predicted that the review was positive when in actuality it was negative.)

523- Not much dialogue, not much music, the whole film was shot as elaborately and aesthetically like a sculpture.(Reasoning: I think there aren't any keywords in this sentence that indicate any sentiments; as a result, I believe my model took a random guess and predicted this as negative as opposed to positive. Even as a reader, this sentence comes across as neutral to me. It's not very clear what direction this review is pointing to as it was quite passive.