Portfolio Milestone

M.S. Applied Data Science Program

Michael Johnson

March 18, 2022

SUID: 516408416

GitHub: https://github.com/AirJohnson3/Portfolio_Project

Syracuse University
School of Information Studies

Table of Contents

Portfolio Introduction

Data collection and subsequent analysis maintained a significant role throughout history and remains vital to the decision-making process today. The science behind understanding that data drives some of the most important advancements through human history, from collecting the positions of stars to fighting the COVID-19 pandemic through information analysis. Companies and governments throughout the world rely on some level of descriptive, predictive, or prescriptive analytics all driven by leveraging data science to derive insights into problems or questions.

The base level of all data starts with a problem or question that could range from a basic “How can we make more money?” to advanced problems like developing self-driving cars. The “science” aspect in data science elicits the idea that undertaking data science problems follows the similar scientific method as many other scientific fields, which all start with asking a question. The initial question or questions of interest drive the acquisition or collection of data for understanding and then provide insight into the modeling and deployment processes as part of the complete data science lifecycle (The Team Data Science Process lifecycle, 2022).

Understanding and working within the scope of the data science lifecycle is an important part of analyzing and building machine learning models. That knowledge provides the foundation of data science within the Applied Data Science graduate program at Syracuse University. Through the program, students identified actionable insights with data analytics through building visualizations and predictive models relying on a mixture of theory and application (Applied Data Science Master’s Degree, 2022). Leveraging an understanding for the tools like Python, R programming, and Excel for analyzing data and building models was an integral part of succeeding in the program.

The following portfolio serves to exemplify the seven learning objectives critical in the Applied Data Science program (Stinnett, 2022):

1. Describe a broad overview of the major practice areas in data science.
2. Collect and organize data.
3. Identify patterns in data via visualization, statistical analysis, and data mining.
4. Develop alternative strategies based on the data.
5. Develop a plan of action to implement the business decisions derived from the analyses.
6. Demonstrate communication skills regarding data and its analysis for relevant professionals in their organization.
7. Synthesize the ethical dimensions of data science practice.

Course Highlight 1: Data Warehouse and Database Administration

Course Description

The IST 722 Data Warehouse course provides concepts, principles, and tools for designing, implementing, and using data warehouses. The course also introduces database constructs such as Operational Data Store (ODS), data warehouse, and data marts. Part of the course discussions include examining the differences between Ralf Kimball’s and Bill Inmon's approaches to data warehouses, roles and responsibilities in the design and implementation of a data warehouse, project management guidelines and techniques, requirements gathering, dimensional modeling, Extract Transform and Load architecture, analytical reporting concepts, data governance and current trends in the data warehouse domain. The course leverages a portfolio of SQL Server tools that include SQL Server DBMS, SQL Server Integration Services (SSIS), and SQL Server Analysis Service (SSAS) (Khan, 2022).

IST 659 is an introductory course to database management systems examining data structures, file organizations, concepts, and principles of database management systems (DBMS) as well as data analysis, database design, data modeling, database management, and database implementation. More specifically, it introduces hierarchical, network, and relational data models; entity-relationship modeling; basics of Structured Query Language (SQL); data normalization; and database design. Using Microsoft’s Access and SQL Server DBMSs as implementation vehicles, this course provides hands-on experience in database design and implementation through assignments, lab exercises, and course projects. This course also introduces advanced database concepts such as transaction management and concurrency control, distributed databases, multitier client/server architectures, web-based database applications, data warehousing, and NoSQL (Harper, 2021).

Learning Objectives

Learning objectives for IST 659 include:

- Describe fundamental data and database concepts.
- Explain and use the database development lifecycle.
- Create databases and database objects using popular database management system products.
- Solve problems by constructing database queries using Structured Query.
- Learn to use the database language SQL.
- Design databases using data modeling and data normalization techniques.
- Develop insights into future data management tool and technique trends.
- Recommend and justify strategies for managing data security, privacy, audit/control, fraud detection, backup, and recovery.
- Critique the effectiveness of DBMS in computer information systems.

Learning objectives for IST 722 include:

- Show technical knowledge
    - Define analytics requirements based on business process understanding.
    - Describe various database constructs - Data Warehouse, Data Mart, ODS.
    - Describe the components of a data warehouse.
    - Differentiate between Ralf Kimball’s and Bill Inmon's approaches.
    - Know how to apply various integration approaches in a merger - ETL, EII, EAI.
    - Describe a Master Data Management (MDM) solution.
    - Create database objects using popular database management system products.

    - Design and implement data warehouse and business intelligence components.
    - Gain extensive hands-on with SSIS (ETL), SSAS (Cube) and Power BI tools.
- Manage the development of solutions
    - Define the roles and responsibilities in the design and development of data warehouses.
    - Differentiate various requirements gathering and dimensional modeling techniques.
    - Relate business processes with objects in the data warehouse.
    - Define project management guidelines.
- Manage information technology
    - Identify business processes in the data warehouse.
    - Describe the data governance concepts.
    - List some of the current trends in Data Warehouse.

Project Requirements

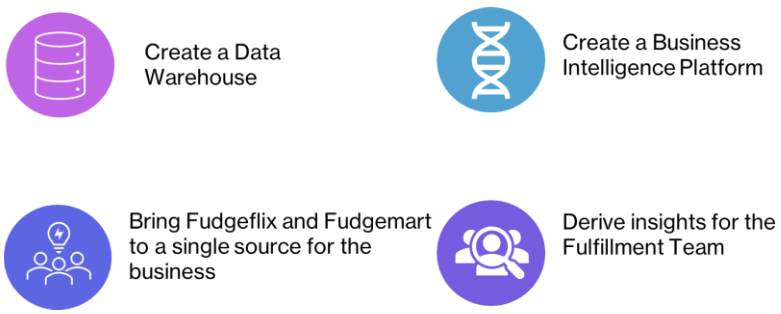
The goal for this project is to build a data warehouse solution from scratch after combing databases from two companies and providing a business intelligence solution for analyzing the merged data. Successfully guiding the merge requires ensuring minimal loss in business processes throughout the transition. All aspects of the business should remain operational from sales to inventory to order fulfillment and everything in between. Fudgemart, Inc. is a fictitious conglomerate with two subsidiary companies:

- Fudgemart: a fictitious online retailer, like Amazon.com or Walmart.com. The database consists of customers, products, and vendors, and has familiar business processes you would find in any online retailer.

- Fudgeflix: a fictitious online DVD by mail and video on demand service, like Amazon instant video or Netflix. The database for Fudgeflix contains concepts such as accounts, subscriptions, and video titles as well as other things associated with an online video streaming service.

Project Development

Initial project steps and base goals:



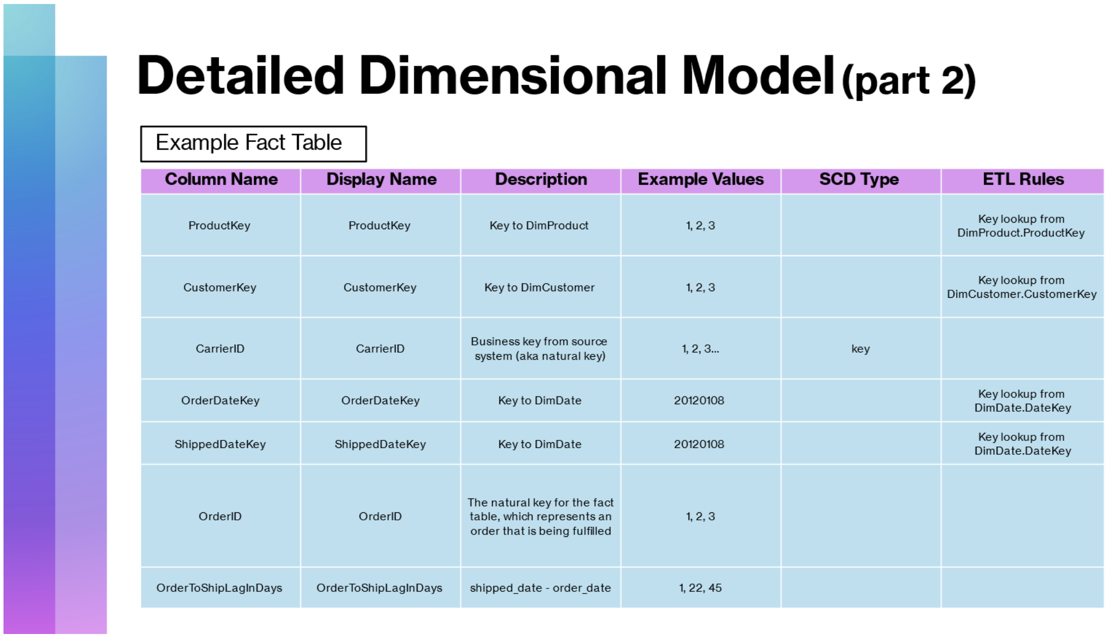
Create a Data Warehouse

Create a Business Intelligence Platform

Bring Fudgeflix and Fudgemart to a single source for the business

Derive insights for the Fulfillment Team

This example shows the creation for the detailed dimensional model for the data warehouse fact table:



Detailed Dimensional Model **(part 2)**

Example Fact Table

Column Name	Display Name	Description	Example Values	SCD Type	ETL Rules
ProductKey	ProductKey	Key to DimProduct	1, 2, 3		Key lookup from DimProduct.ProductKey
CustomerKey	CustomerKey	Key to DimCustomer	1, 2, 3		Key lookup from DimCustomer.CustomerKey
CarrierID	CarrierID	Business key from source system (aka natural key)	1, 2, 3...	key	
OrderDateKey	OrderDateKey	Key to DimDate	20120108		Key lookup from DimDate.DateKey
ShippedDateKey	ShippedDateKey	Key to DimDate	20120108		Key lookup from DimDate.DateKey
OrderID	OrderID	The natural key for the fact table, which represents an order that is being fulfilled	1, 2, 3		
OrderToShipLagInDays	OrderToShipLagInDays	shipped_date - order_date	1, 22, 45		

Merging the two companies required fitting each of the database structures into the same structure. The following is the SQL code to build the Data Warehouse:

```
IF EXISTS (SELECT Name from sys.extended_properties where Name = 'Description')
    EXEC sys.sp_dropextendedproperty @name = 'Description'
EXEC sys.sp_addextendedproperty @name = 'Description', @value = 'Default description - you should change this.'
;

-- Create a schema to hold user views (set schema name on home page of workbook).
-- It would be good to do this only if the schema doesn't exist already.
--GO
--CREATE SCHEMA group_two
--GO

/* Drop table group_two.FactOrderFulfillment */
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = OBJECT_ID(N'group_two.FactOrderFulfillment') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE group_two.FactOrderFulfillment
;

/* Create table group_two.FactOrderFulfillment */
CREATE TABLE group_two.FactOrderFulfillment (
   [ProductKey]  int   NOT NULL
,  [CustomerKey]  int   NOT NULL
,  [CarrierID]  nvarchar(50)    NOT NULL
,  [OrderDateKey]  int   NOT NULL
,  [ShippedDateKey]  int   NULL
,  [OrderID]  int   NOT NULL
,  [OrderToShipLagInDays]  int   NULL
, CONSTRAINT [PK_group_two.FactOrderFulfillment] PRIMARY KEY NONCLUSTERED (OrderID, ProductKey)
) ON [PRIMARY]
;

/* Drop table group_two.DimDate */
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = OBJECT_ID(N'group_two.DimDate') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE group_two.DimDate
;

/* Create table group_two.DimDate */
CREATE TABLE group_two.DimDate (
   [DateKey]  int NOT NULL
,  [Date]  datetime   NULL
,  [FullDateUSA]  nchar(11)   NOT NULL
,  [DayOfWeek]  tinyint   NOT NULL
,  [DayName]  nchar(10)   NOT NULL
,  [DayOfMonth]  tinyint   NOT NULL
,  [DayOfYear]  int   NOT NULL
,  [WeekOfYear]  tinyint   NOT NULL
,  [MonthName]  nchar(10)   NOT NULL
,  [MonthOfYear]  tinyint   NOT NULL
,  [Quarter]  tinyint   NOT NULL
,  [QuarterName]  nchar(10)   NOT NULL
,  [Year]  int   NOT NULL
,  [IsWeekday]  bit  DEFAULT 0 NOT NULL
, CONSTRAINT [PK_group_two.DimDate] PRIMARY KEY CLUSTERED
( [DateKey] )
) ON [PRIMARY]
;
```

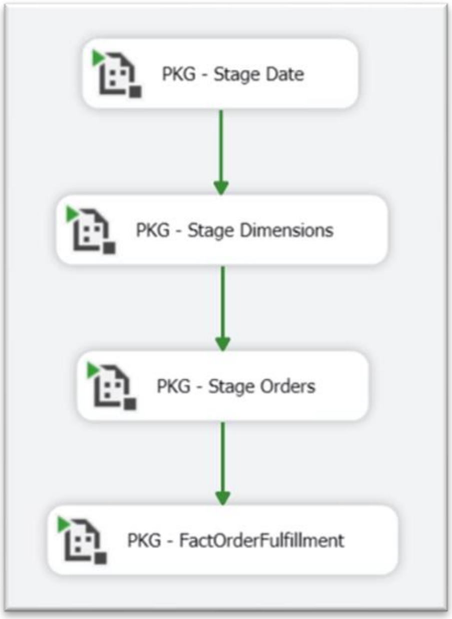
```
INSERT INTO group_two.DimDate (DateKey, Date, FullDateUSA, DayOfWeek, DayName, DayOfMonth, DayOfYear,
WeekOfYear, MonthName, MonthOfYear, Quarter, QuarterName, Year, IsWeekday)
VALUES (-1, '', 'Unk date', 0, 'Unk date', 0, 0, 0, 'Unk month', 0, 0, 'Unk qtr', 0, 0)
;

GO
UPDATE group_two.DimDate
SET Date = Null
WHERE DateKey = -1;
GO

-- User-oriented view definition
GO
IF EXISTS (select * from sys.views where object_id=OBJECT_ID(N'[group_two].[Date]'))
DROP VIEW [group_two].[Date]
GO
CREATE VIEW [group_two].[Date] AS
SELECT [DateKey] AS [DateKey]
, [Date] AS [Date]
, [FullDateUSA] AS [FullDateUSA]
, [DayOfWeek] AS [DayOfWeek]
, [DayName] AS [DayName]
, [DayOfMonth] AS [DayOfMonth]
, [DayOfYear] AS [DayOfYear]
, [WeekOfYear] AS [WeekOfYear]
, [MonthName] AS [MonthName]
, [MonthOfYear] AS [MonthOfYear]
, [Quarter] AS [Quarter]
, [QuarterName] AS [QuarterName]
, [Year] AS [Year]
, [IsWeekday] AS [IsWeekday]
FROM group_two.DimDate
GO

/* Drop table group_two.DimProduct */
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = OBJECT_ID(N'group_two.DimProduct') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
DROP TABLE group_two.DimProduct
;

/* Create table group_two.DimProduct */
CREATE TABLE group_two.DimProduct (
  [ProductKey]  int IDENTITY  NOT NULL
,  [ProductID]  int   NOT NULL
,  [product_department]  varchar(20)   NOT NULL
,  [product_name]  varchar(200)   NOT NULL
,  [RowIsCurrent]  bit   DEFAULT 1 NOT NULL
,  [RowStartDate]  datetime  DEFAULT '12/31/1899' NOT NULL
,  [RowEndDate]  datetime  DEFAULT '12/31/9999' NOT NULL
,  [RowChangeReason]  nvarchar(200)   NULL
, CONSTRAINT [PK_group_two.DimProduct] PRIMARY KEY CLUSTERED
( [ProductKey] )
) ON [PRIMARY]
;

SET IDENTITY_INSERT group_two.DimProduct ON
;
INSERT INTO group_two.DimProduct (ProductKey, ProductID, product_department, product_name, RowIsCurrent,
RowStartDate, RowEndDate, RowChangeReason)
VALUES (-1, -1, 'Unk Department', 'Unk Product', 1, '12/31/1899', '12/31/9999', 'N/A')
```

```
;

SET IDENTITY_INSERT group_two.DimProduct OFF

;


-- User-oriented view definition

GO

IF EXISTS (select * from sys.views where object_id=OBJECT_ID(N'[group_two].[Product]'))

DROP VIEW [group_two].[Product]

GO

CREATE VIEW [group_two].[Product] AS

SELECT [ProductKey] AS [ProductKey]

, [ProductID] AS [ProductID]

, [product_department] AS [product_department]

, [product_name] AS [product_name]

, [RowIsCurrent] AS [Row Is Current]

, [RowStartDate] AS [Row Start Date]

, [RowEndDate] AS [Row End Date]

, [RowChangeReason] AS [Row Change Reason]

FROM group_two.DimProduct

GO


/* Drop table group_two.DimCustomer */

IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = OBJECT_ID(N'group_two.DimCustomer') AND

OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE group_two.DimCustomer

;


/* Create table group_two.DimCustomer */

CREATE TABLE group_two.DimCustomer (

  [CustomerKey]  int IDENTITY  NOT NULL
,  [CustomerID]  int   NOT NULL
,  [customer_city]  varchar(50)   NULL
,  [customer_state]  char(2)   NULL
,  [customer_zip]  varchar(10)   NULL
,  [RowIsCurrent]  bit   DEFAULT 1 NOT NULL
,  [RowStartDate]  datetime  DEFAULT '12/31/1899' NOT NULL
,  [RowEndDate]  datetime  DEFAULT '12/31/9999' NOT NULL
,  [RowChangeReason]  nvarchar(200)   NULL
, CONSTRAINT [PK_group_two.DimCustomer] PRIMARY KEY CLUSTERED
( [CustomerKey] )
) ON [PRIMARY]
;

SET IDENTITY_INSERT group_two.DimCustomer ON
;
INSERT INTO group_two.DimCustomer (CustomerKey, CustomerID, customer_city, customer_state, customer_zip,
RowIsCurrent, RowStartDate, RowEndDate, RowChangeReason)
VALUES (-1, -1, 'Unknown City', 'ZZ', '00000-0000', 1, '12/31/1899', '12/31/9999', 'N/A')
;
SET IDENTITY_INSERT group_two.DimCustomer OFF
;
-- User-oriented view definition
GO
IF EXISTS (select * from sys.views where object_id=OBJECT_ID(N'[group_two].[Customer]'))
DROP VIEW [group_two].[Customer]
GO
CREATE VIEW [group_two].[Customer] AS
SELECT [CustomerKey] AS [CustomerKey]
, [CustomerID] AS [CustomerID]
, [customer_city] AS [customer_city]
, [customer_state] AS [customer_state]

, [customer_zip] AS [customer_zip]

, [RowIsCurrent] AS [Row Is Current]

, [RowStartDate] AS [Row Start Date]

, [RowEndDate] AS [Row End Date]

, [RowChangeReason] AS [Row Change Reason]

FROM group_two.DimCustomer

GO

-- User-oriented view definition

GO

IF EXISTS (select * from sys.views where object_id=OBJECT_ID(N'[group_two].[OrderFulfillment]'))

DROP VIEW [group_two].[OrderFulfillment]

GO

CREATE VIEW [group_two].[OrderFulfillment] AS

SELECT [ProductKey] AS [ProductKey]

, [CustomerKey] AS [CustomerKey]

, [CarrierID]  AS  [CarrierID]

, [OrderDateKey] AS [OrderDateKey]

, [ShippedDateKey] AS [ShippedDateKey]

, [OrderID] AS [OrderID]

, [OrderToShipLagInDays] AS [OrderToShipLagInDays]

FROM group_two.FactOrderFulfillment

GO


ALTER TABLE group_two.FactOrderFulfillment ADD CONSTRAINT

   FK_group_two_FactOrderFulfillment_ProductKey FOREIGN KEY

   (

   ProductKey

   ) REFERENCES group_two.DimProduct

   ( ProductKey )
   ON UPDATE  NO ACTION
   ON DELETE  NO ACTION
;

ALTER TABLE group_two.FactOrderFulfillment ADD CONSTRAINT
  FK_group_two_FactOrderFulfillment_CustomerKey FOREIGN KEY
  (
  CustomerKey
  ) REFERENCES group_two.DimCustomer
  ( CustomerKey )
   ON UPDATE  NO ACTION
   ON DELETE  NO ACTION
;

ALTER TABLE group_two.FactOrderFulfillment ADD CONSTRAINT
  FK_group_two_FactOrderFulfillment_OrderDateKey FOREIGN KEY
  (
  OrderDateKey
  ) REFERENCES group_two.DimDate
  ( DateKey )
   ON UPDATE  NO ACTION
   ON DELETE  NO ACTION
;

ALTER TABLE group_two.FactOrderFulfillment ADD CONSTRAINT
  FK_group_two_FactOrderFulfillment_ShippedDateKey FOREIGN KEY
  (
  ShippedDateKey
  ) REFERENCES group_two.DimDate
  ( DateKey )
```

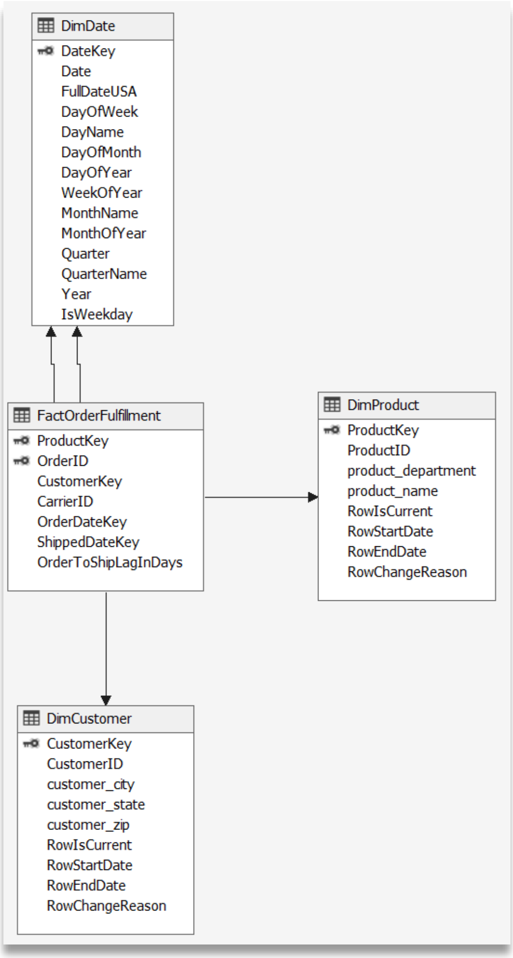
```
    ON UPDATE  NO ACTION
    ON DELETE  NO ACTION
;
```

The ETL process consisted of loading Date, Product, Customer, and Orders into the staging database. During staging, data conversion and derived columns fixed any differences in data type between companies.

Linking primary keys to the correct Fudgemart and Fudgeflix databases was integral to this process for mapping the correct orders to the appropriate customers and products.



The single fact table represents the order fulfillment business process for the merged organization. Date, Product and Customer dimensions provide meaningful context for measuring order fulfillment performance. The following star schema shows the data warehouse structure with the fact table and surrounding dimension tables:



DimDate
- DateKey
- Date
- FullDateUSA
- DayOfWeek
- DayName
- DayOfMonth
- DayOfYear
- WeekOfYear
- MonthName
- MonthOfYear
- Quarter
- QuarterName
- Year
- IsWeekday

FactOrderFulfillment
- ProductKey
- OrderID
- CustomerKey
- CarrierID
- OrderDateKey
- ShippedDateKey
- OrderToShipLagInDays

DimProduct
- ProductKey
- ProductID
- product_department
- product_name
- RowIsCurrent
- RowStartDate
- RowEndDate
- RowChangeReason

DimCustomer
- CustomerKey
- CustomerID
- customer_city
- customer_state
- customer_zip
- RowIsCurrent
- RowStartDate
- RowEndDate
- RowChangeReason

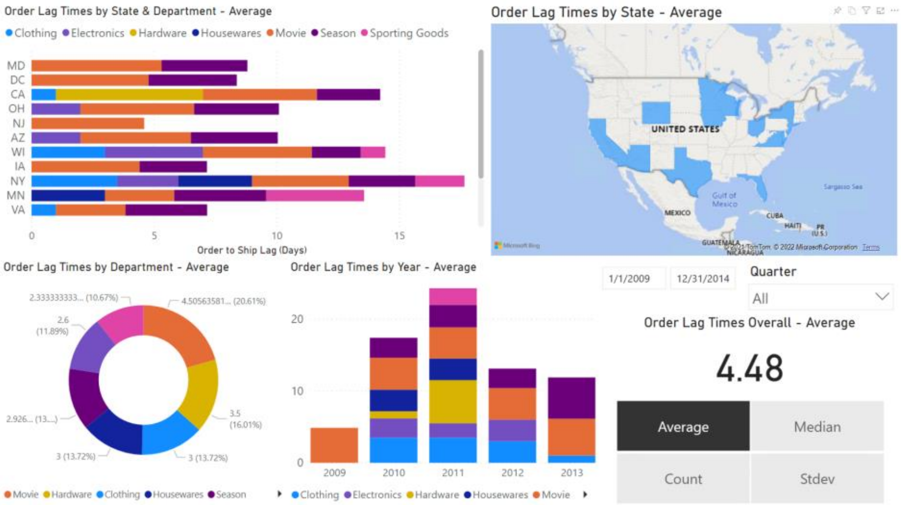
The following were goals for reaching success on the business intelligence side:

Empower the business to perform Analytics

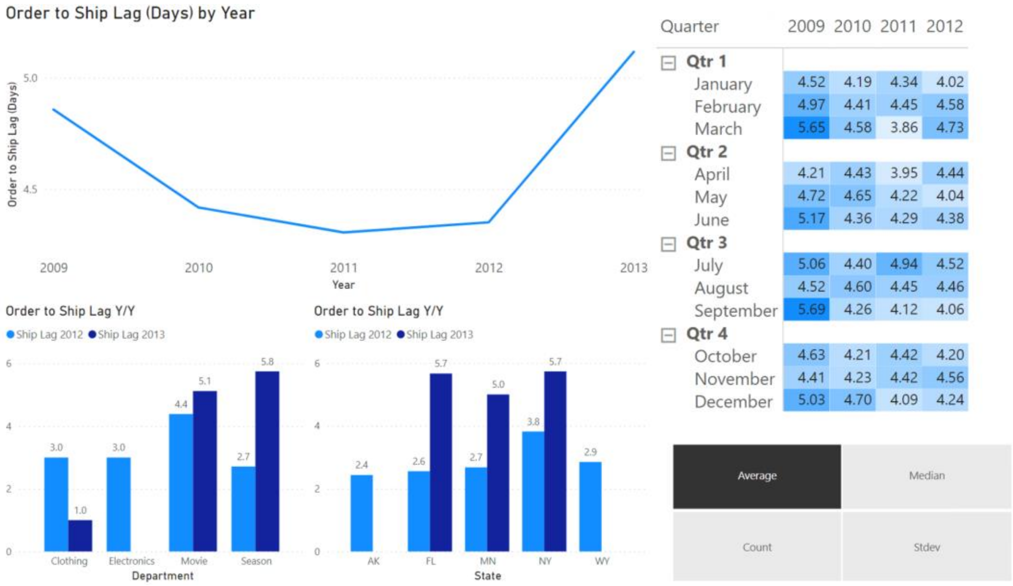
Allow access to multiple dimensions to slice and filter data

Provide tools necessary for visualization

With the structured data in place, visualizations help depict the business process for order fulfillment and allowed the organization to better understand the weaker points:


Order Lag Times by State & Department - Average
● Clothing ● Electronics ● Hardware ● Housewares ● Movie ● Season ● Sporting Goods
MD
DC
CA
OH
NJ
AZ
WI
IA
NY
MN
VA
0
5
10
15
Order to Ship Lag (Days)
Order Lag Times by State - Average
UNITED STATES
MEXICO
Gulf of Mexico
Sargasso Sea
CUBA
HAITI
PR (U.S.)
GUATEMALA
NICARAGUA
Microsoft Bing
© 2022 TomTom © 2022 Microsoft Corporation Terms
Order Lag Times by Department - Average
2.333333333... (10.67%)
4.50563581... (20.61%)
2.6 (11.89%)
3.5 (16.01%)
2.926... (13....)
3 (13.72%)
3 (13.72%)
● Movie ● Hardware ● Clothing ● Housewares ● Season
Order Lag Times by Year - Average
20
10
0
2009
2010
2011
2012
2013
● Clothing ● Electronics ● Hardware ● Housewares ● Movie
1/1/2009
12/31/2014
Quarter
All
Order Lag Times Overall - Average
4.48
Average
Median
Count
Stdev

Looking at the order fulfillment geographically and by department was important but the organization also required a time-based analysis:



Quarter	2009	2010	2011	2012
⊟ **Qtr 1**				
January	4.52	4.19	4.34	4.02
February	4.97	4.41	4.45	4.58
March	5.65	4.58	3.86	4.73
⊟ **Qtr 2**				
April	4.21	4.43	3.95	4.44
May	4.72	4.65	4.22	4.04
June	5.17	4.36	4.29	4.38
⊟ **Qtr 3**				
July	5.06	4.40	4.94	4.52
August	4.52	4.60	4.45	4.46
September	5.69	4.26	4.12	4.06
⊟ **Qtr 4**				
October	4.63	4.21	4.42	4.20
November	4.41	4.23	4.42	4.56
December	5.03	4.70	4.09	4.24

Based on the analysis, the following recommendations go to the organization’s decision-makers:

Capture received date to perform end-to end BI

Reduce lead time for processing movie orders

Offer promotional discounts to customers who experience high lead time

Transform Fudgeflix into a full-fledged streaming service

Course Reflection

From a data engineering standpoint, the data warehouse and database management objectives listed above help in efficiently capturing, organizing, and promoting adequate data storage for a successful business warehouse. From a data science lens, knowledge around these objectives is critical for understanding data and performing successful data analysis. The projects in both classes also highlighted some important aspects to keep in mind when constructing a database or data warehouse:

- Gathering information from all parties in the organization prior to starting a project helps in capturing every bit of the organization's information needs.
- Conceptual and logical designs are critical to developing a well thought out database or data warehouse and serve as the linchpin for effectively communicating with both technical developers and business-focused individuals within an organization.
- During the course projects, both professors instructed students to make assumptions about the data and how organizations use the data. However, effective communication with key stake holders is an absolute requirement during the development of databases and data warehouses in the real world.

These two courses fulfill requirements for all seven of the learning objectives through both the data warehouse construction and the subsequent business intelligence analysis based on the data.

- **Describe a broad overview of the major practice areas in data science:**

  These courses exemplify the major practices of effectively using SQL to build and query databases and data warehouses. Additionally, the Data Warehouse course also required applying the principles needed for business intelligence to analyze the data in the scope of major business processes.

- **Collect and organize data:**

  Collecting and organizing data was a pivotal aspect for creating databases and data warehouses and required using tools for planning data storage based on an organization's data requirements.

- **Identify patterns in data via visualization, statistical analysis, and data mining:**

  Both courses required using PowerBI, Excel Pivot Tables, and SQL queries to evaluate, analyze, and visualize the data stored within the database and data warehouse. Aggregating and creating visuals on the order fulfillment business process side enhanced the understanding for how the data warehouse managed the flow of data.

- **Develop alternative strategies based on the data:**

  Specifically in the Data Warehouse course, merging of both databases together required producing alternative solutions for combining the data. Each database had a unique database structure that needed adjustment to fit into a holistic data warehouse.

- **Develop a plan of action to implement the business decisions derived from the analyses:**

  The focus point for the business process was analyzing order fulfillment from the data warehouse and identifying areas of improvement across both time and geographic elements. The order lag times were the vital metric for evaluating the company's ability to fulfill orders made by customers.

- **Demonstrate communication skills regarding data and its analysis for relevant professionals in their organization:**

  Each team member served distinct functions for the scope of the data warehouse project and required effective communication to ensure each piece fit together. Prior to building the data warehouse, writing down the over-arching business processes guided the effective communication between decision-makers and technical professionals.

- **Synthesize the ethical dimensions of data science practice.**

  The Database Management course stressed the use of database security to protect the identity of individuals and restrict access to customer financial information.

Course Highlight 2: Natural Language Processing

Course Description

The aim of the Natural Language Processing (NLP) course is to develop an understanding of how to process written text and produce a linguistic analysis used in various applications. The course primarily covers the techniques of NLP in the levels of linguistic analysis, going through tokenization, word-level semantics, part-of-speech tagging, syntax, semantics, and the higher discourse level. The course also includes NLP techniques, such as information retrieval, question answering, sentiment analysis, summarization, and dialogue systems, in applications (Larche, 2021).

Learning Objectives

- Demonstrating the levels of linguistic analysis, the computational techniques used to understand text at each level, and challenges using those techniques.
- Processing text through the language levels using the resources of the Natural Language Toolkit (NLTK) and use of the programming language Python.
- Describing the use of NLP in real-world applications.

Project Requirements

Text classification tasks:

1. The first step is to process the text, tokenize the text, and choose whether to do further pre-processing or filtering.
2. The second step is to produce the features in the notation of the NLTK by writing feature functions in Python. Start with the "bag-of-words" features and collect all the words in the corpus then select some number of most frequent words to be the word features. Use

the NLTK Naïve Bayes classifier to train and test a classifier on the created feature sets. Use cross-validation to obtain precision, recall and F-measure scores or produce the features as a csv file and use Weka or Sci-Kit Learn to train and test a classifier using cross-validation scores.

3. For a base level completion of experiments, carry out at least several experiments using two different sets of features and compare the results.
4. Draft a report that describes the data processing, the features, and the classification experiments.

Project Development

Introduction

Found on Kaggle, the Rotten Tomatoes movie review dataset is a corpus of movie reviews used for sentiment analysis that was originally collected by Bo Pang and Lillian Lee in 2005. The data consists of comments left by Rotten Tomatoes users that have been broken down as follows: each comment (phrase) gets assigned a SentenceId number and is then parsed word by word, receiving a different PhraseId with each parsing. The initial phrase is the most complete one and is therefore used in our analysis. The training dataset also contains a sentiment rating indicating the strength and polarity of each phrase. Those ratings are as follows:

0 - negative

1 - somewhat negative

2 - neutral

3 - somewhat positive

4 - positive

Goals:

The purpose of this project is to accurately classify movie reviews based on the sentiment ratings for each review. Achieving this goal requires building multiple classification algorithms, applying those to text processed in a variety of ways, and comparing the results to find the best algorithm and preprocessing technique. The outcome results from this analysis will include the best document grouping, the best featureset, and the best model type to accurately classify the movie reviews.

To achieve these goals, the movie review corpus went through three primary steps:

1. Text Processing
2. Feature Engineering
3. Experiments

Step 1: Text Processing

Processing the Text:

Processing the Kaggle Movie Reviews corpus and preparing it for tokenizing, filtering, and pre-processing involved splitting the reviews up into two separate documents. The first document included grouping the data by sentence ID and retaining just the complete reviews with sentiment ratings instead of having them broken down into smaller pieces. The second document uses the entire dataset with each of the broken out reviews and their associated sentiment ratings. The comparison between both documents is the foundation of this analysis and will drive the comparison between each of the featuresets built using the documents.

Tokenizing the Text:

Tokenization process for each of the documents is the primary analysis and will serve as the baseline for comparison to the filtered documents, the pre-processed documents, and a combination of filtering and pre-processing. The tokenization function (called process_text) imports the document lists created in the processing section and performs four primary functions on each of the reviews that will remain constant throughout the filtering and pre-processing steps. After pulling in the document, the tokenizer uses the NLTK tokenize function, turns all words into lowercase, filters out a list of punctuations, and appends the resulting tokens with their sentiment ratings to a list. This is the baseline comparison to see if filtering the documents and pre-processing the documents holds any bearing for the classification models.

Filtering the Text:

Filtering text documents is the first comparison to only tokenizing the documents and includes all the same tokenization functions with additional features built in to filter the documents. The first function removes all non-alphabetic characters, relying on pattern matching using regular expressions with the following syntax: pattern = re.compile(r'^[^a-z]+$').

The next function in filtering the data includes removing words under two letters in length as a majority of the words would be filtered out during the next step. This removes basic words like “on” and “a” as part of preparing a smaller list of stopwords needed. The final filtering step includes removing all the base stopwords in the NLTK stopwords list, as well as additional stopwords defined by their common usage and lack of importance based on the frequency of showing up in the documents. The following is a list of the NLTK stopwords and the additional words filtered from the documents:

NLTK Stopwords: ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your','yours',

'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'her', 'hers',

'herself', 'it', 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves',

'what', 'which', 'who', 'whom', 'this', 'that', 'these', 'those', 'am', 'is', 'are',

'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does',

'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until',

'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into',

'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down',

'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here',

'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more',

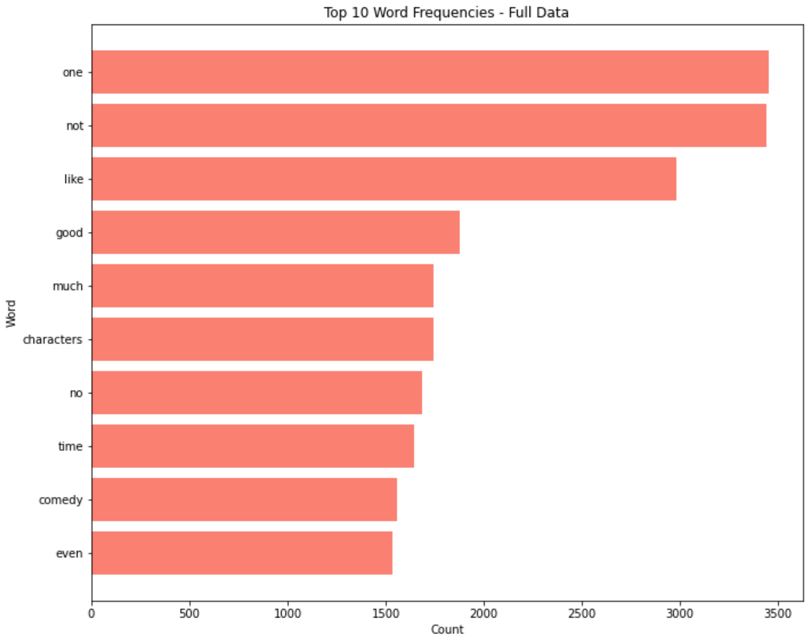
'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so',

'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', 'should', 'now']

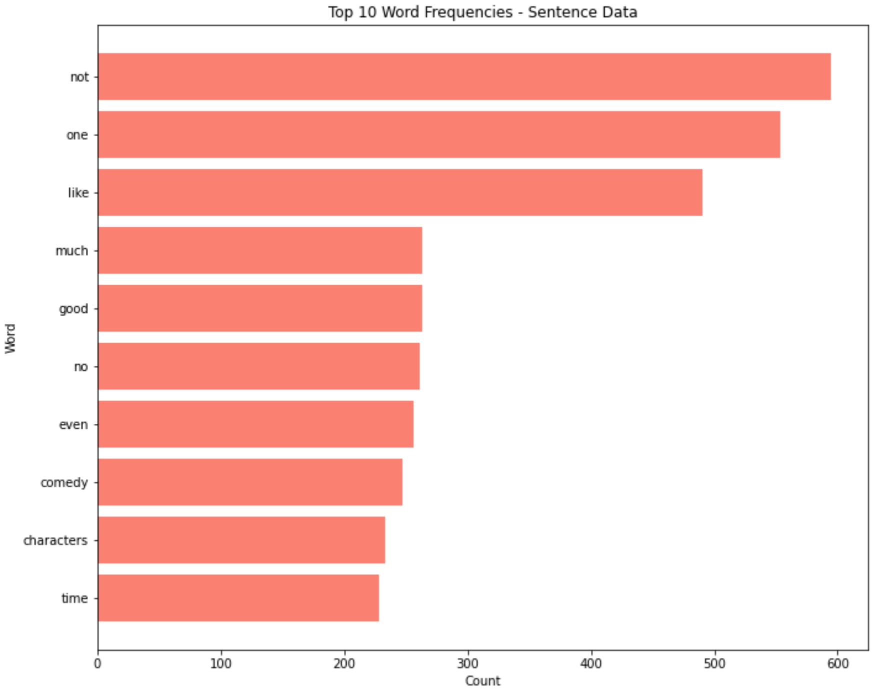
Additional Stopwords: ['can','has','have','had','must','shan','do', 'should','was', 'were','won','are','cannot','does','could','did','is','might','need','would', "'s", 'film', 'movie', 'story', '-rrb-', '-lrb-',"'re", "n't"]

Note: any stopwords that fell into the negation words list were removed from the filtering to allow for comparing negative featuresets to the rest of the data.

Top 10 word frequencies for all data:



Top 10 word frequencies for just sentence data:



Although the sentence data uses complete reviews and the full data has the split out reviews, it makes sense for the frequencies to align between the two. The sentence data has significantly less frequencies than the full data and actually flips the first and second most common words as well as a couple other words throughout the text. This signals the model performance between datasets could align in terms of rankings but the sentence-only data should have much lower confidence levels due to the reduced word count.

Pre-processing the Text:

Pre-processing text documents is the second comparison to only tokenizing the documents and includes all the same tokenization functions with additional features built in to filter the documents. The first function includes stemming the words using the Porter stemmer from NLTK. The Lancaster stemmer was also tested but removed from the pre-processing due to the aggressive stemming observed in the documents. As a more gentle stemming function, the Porter algorithm was also preferred due to being the standard stemming algorithm used for most NLP tasks. Stemming involves chopping off the ends of the words and was used first in this analysis to prepare the data for lemmatization. Using the NLTK lemmatizer, the words were lemmatized to return only the base form of a word.

Pre-processing and Filtering the Text:

The final point of comparison to the tokenized documents is a combination of both filtering and pre-processing. This section repeats the same functions observed in the filtering and pre-processing sections, while ensuring that words filtered out do not end up in the end documents.

Final Text Processing Experiments:

Experiments will run against two primary datasets: all data available and only the first line of each review containing the full sentence.

The following is a breakdown of the planned experiments based on all the data:

- All tokenized data
- All filtered and tokenized data
- All pre-processed and tokenized data
- All pre-processed, filtered, and tokenized data

The following is a breakdown of the planned experiments based on only full sentences:

- Full tokenized sentences
- Full filtered and tokenized sentences
- Full pre-processed data and tokenized sentences
- Full pre-processed, filtered, and tokenized sentences

The eight final text processing documents listed above will go through each of the nine feature engineering stages, resulting in a total of 72 varying featuresets compared during the experiments stage. Although there are quite a bit of featuresets to compare, the end result should meet the goals listed within the introduction section with adequate comparison between the different text processing techniques. Each of the documents were randomized using the same seed number to reproduce the experiments so all processing started with the same documents. To save on performance, only the first 1,000 reviews of the randomized documents will go through the feature engineering and modeling.

Note: this method leaves room for introducing additional documents for comparison. Each of the functions within both the filtering and the pre-processing could be split out to identify the best document processing for this dataset. The individual filtering and pre-processing steps outlined above and their combinations would result in hundreds of featureset to test and model.

Step 2: Feature Engineering

The goal in this stage is to produce the featuresets by passing the processed documents through functions that identified word features, bigram features, and additional word similarities captured in various lexicons and outputting the result into dictionaries. Each dictionary consists of a word or bigram with a true or false rating and the sentiment number for the tokenized review. As mentioned previously, the 8 documents from the processing stage all go through the feature engineering stage for a total of 72 featuresets. Each of the featuresets will pass through the modeling stage to find the best combination of processing and feature engineering for the data.

The feature engineering functions all rely on the same word features function to ensure the same word count is used for each of the featuresets. The function defines a list of all the words and sentiments in each document, finds the frequency of the words in the reviews, and uses the top 1,000 most common words to produce the word features.

Bag of Words/Unigrams Features:

The BoW approach consists in collecting all the words in the corpus and asserting two things: the vocabulary of the known words and a measure of the presence of the known words. This approach allows us to get a count of words as well as their frequency. The resulting featureset contains the word and a true or false rating to capture whether the word was in the word features list. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the bag of words featureset using token processing only:

```
[{'V_the': False,
  'V_a': True,
  'V_and': True,
  'V_of': False,
  'V_to': False,
  "V_'s": False,
  'V_that': False,
  'V_in': False,
  'V_is': False,
  'V_it': False},0]
```

Bigrams Features:

Similar to the unigram count, the bigram feature allows us to get a count and frequency of two word combinations that occur in the corpus. The resulting featureset contains all the bigrams within each review and a true or false rating for each bigram of the review to capture whether the bigrams within the document were in the bigram word features list with the top 1,000 most common bigrams. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the bigram featureset using token processing only:

```
[{'B_1_drips': False,
  'B_102-minute_infomercial': False,
  'B_94-minute_travesty': False,
  'B_a.s._byatt': False,
  'B_accurately_reflects': False,
  'B_across_america': False,
  'B_act_abroad': False,
  'B_acted_skateboards': False,
  'B_added_clout': False,
  'B_adding_flourishes': False},0]
```

Bigrams and Bag of Words Features:

The resulting featureset contains both the bigrams and bag of words unigrams within each review and a true or false rating for each bigram and unigram of the review. This captures whether the bigrams and unigrams within the document were in the bigram and unigram word features list with the top 1,000 most common bigrams and unigrams. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the bigram and bag of words featureset using token processing only (note: bigrams are also part of the featureset, they are applied at the end):

```
[{'V_the': False,
  'V_a': True,
  'V_and': True,
  'V_of': False,
  'V_to': False,
  "V_'s": False,
  'V_that': False,
  'V_in': False,
  'V_is': False,
  'V_it': False},0]
```

Part-of-speech Features:

The PoS process consists in categorizing words of the corpus in correspondence with a particular part of speech, depending on the definition of the word and its context. The various elements of the corpus get tagged as nouns, verbs, adjectives, adverbs, etc. The resulting featureset contains all the words within each review and a true or false rating for whether the word is in the part-of-speech tagging word features list with the top 1,000 most common words. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the part-of-speech featureset using token processing only:

```
[{'contains(the)': False,
  'contains(a)': True,
  'contains(and)': True,
  'contains(of)': False,
  'contains(to)': False,
  "contains('s)": False,
  'contains(that)': False,
  'contains(in)': False,
  'contains(is)': False,
  'contains(it)': False},0]
```

Negative Features:

This feature lists words with a negative connotation (such as “no”, “not”, ”never”) but also approximate negators like “hardly” and “rarely”. The negative feature involves two approaches: one is to negate the word following the negation word and the other is to negate all words following the negation word up to the next punctuation mark.The resulting featureset contains all the words within each review and a true or false rating for whether the word is in the negative word features list with the top 1,000 most common words. This feature set also contains the unigrams and their associated true or false rating based on the word features. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the negative featureset using token processing only:

```
[{'V_the': False,
  'V_NOTthe': False,
  'V_a': True,
  'V_NOTa': False,
  'V_and': True,
  'V_NOTand': False,
  'V_of': False,
  'V_NOTof': False,
  'V_to': False,
  'V_NOTto': False},0]
```

Subjectivity Lexicon:

Subjectivity refers to expression of opinions, evaluations, feelings and speculations (thus incorporating sentiment). The subjectivity lexicon therefore includes clues relating to these perceptions usually grouped as predefined lists of words associated with the emotional context (such as positive/negative)

```
[{'V_the': False,
 'V_a': True,

 'V_and': True,

 'V_of': False,

 'V_to': False,

 "V_'s": False,

 'V_that': False,

 'V_in': False,

 'V_is': False,

 'V_it': False},0]
```

Sentiment Lexicon (LIWC):

Linguistic Inquiry and Word Count is a text analysis program that calculates the degree to which various categories of words (such as positive or negative emotions, self-references or causal words) are used in a text. The resulting featureset contains all the words within each review and a true or false rating for whether the word is in the LIWC lexicon positive or negative word features list with the top 1,000 most common words. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the LIWC featureset using token processing only:

```
[{'contains(the)': False,

 'contains(a)': False,
 'contains(and)': False,

 'contains(of)': False,

 'contains(to)': False,

 "contains('s)": False,

 'contains(that)': False,

 'contains(in)': False,

 'contains(is)': False,

 'contains(it)': False},0]
```

Subjectivity and Sentiment Lexicon (LIWC):

The resulting featureset contains both the words that were in the LIWC lexicon and the subjectivity lexicon with a true or false rating for each word of the review. This captures whether the words within the document were in the top 1,000 most common word features built using the respective lexicons. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the Subjectivity and LIWC featureset using token processing only:

```
[{'contains(the)': False,

 'contains(a)': False,

 'contains(and)': False,

 'contains(of)': False,
 'contains(to)': False,

 "contains('s)": False,

 'contains(that)': False,

 'contains(in)': False,

 'contains(is)': False,

 'contains(it)': False},0]
```

Additional Lexicon:

The Harvard IV (4) Inquirer was also used, it consists of various categories of words expressing specific concepts, some of which are feelings (pleasure and pain), social categories (race, kinship), objects (tools, food, vehicle), and cognitive abilities like knowing, thinking or problem solving.The resulting featureset contains all the words within each review and a true or false rating for whether the word is in the Harvard IV-4 lexicon with positive or negative and strengths for each word in the word features list within the lexicon. The featureset also contains the sentiment rating for the review to serve as the classification point for modeling. The following output is the first 10 words of the first review within the Harvard IV-4 Inquirer featureset using token processing only:

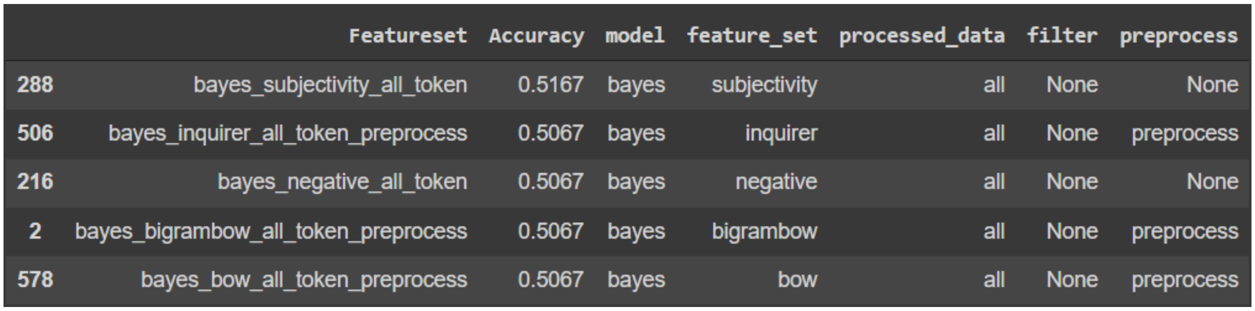
```
[{'V_the': False,

 'V_a': True,

 'V_and': True,

 'V_of': False,
 'V_to': False,

 "V_'s": False,

 'V_that': False,

 'V_in': False,

 'V_is': False,

 'V_it': False},0]
```

Step 3: Experiments

Naive Bayes Classifier:

Based on Bayes theorem, the Naive Bayes classifier builds a model by establishing relationships between features in a very general way; it works in a supervised manner by predicting a test outcome based on a training dataset. This classifier is a probabilistic classifier which means that given an input, it predicts the probability of the input being classified for all the classes. It is also called conditional probability.

Top 5 Results:



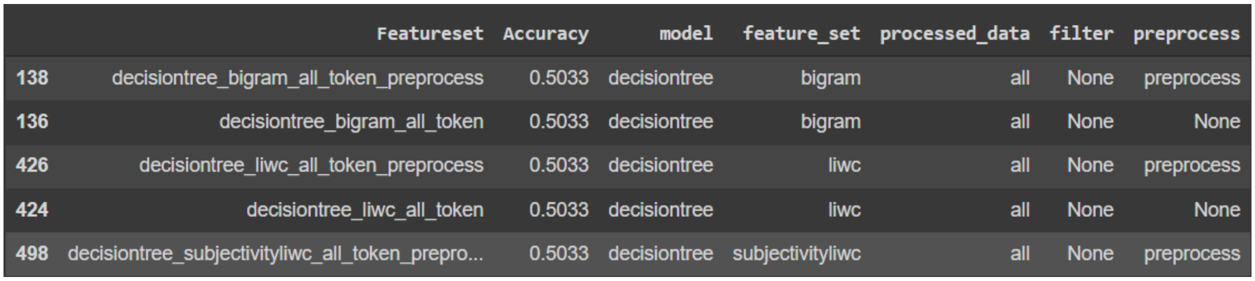
	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
288	bayes_subjectivity_all_token	0.5167	bayes	subjectivity	all	None	None
506	bayes_inquirer_all_token_preprocess	0.5067	bayes	inquirer	all	None	preprocess
216	bayes_negative_all_token	0.5067	bayes	negative	all	None	None
2	bayes_bigrambow_all_token_preprocess	0.5067	bayes	bigrambow	all	None	preprocess
578	bayes_bow_all_token_preprocess	0.5067	bayes	bow	all	None	preprocess

With the Naive Bayes classifier, we had a 0.5167 accuracy using the subjectivity featureset with all data and no filtering or pre-processing.

SKLearn Classifiers:

Decision Tree: it functions by breaking down a dataset into smaller and smaller subsets based on different criteria. Many decision trees linked together result in a Random Forest.

Top 5 Results:

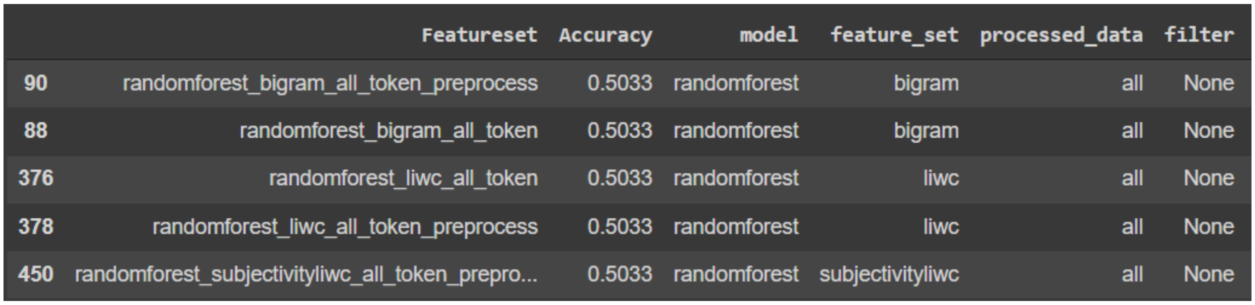


	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
138	decisiontree_bigram_all_token_preprocess	0.5033	decisiontree	bigram	all	None	preprocess
136	decisiontree_bigram_all_token	0.5033	decisiontree	bigram	all	None	None
426	decisiontree_liwc_all_token_preprocess	0.5033	decisiontree	liwc	all	None	preprocess
424	decisiontree_liwc_all_token	0.5033	decisiontree	liwc	all	None	None
498	decisiontree_subjectivityliwc_all_token_prepro...	0.5033	decisiontree	subjectivityliwc	all	None	preprocess

The Decision Tree model gave us a 0.5033 accuracy using the bigram featureset with all data and no filtering or pre-processing.

Random Forest: as mentioned above, it is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of the dataset.

Top 5 Results:

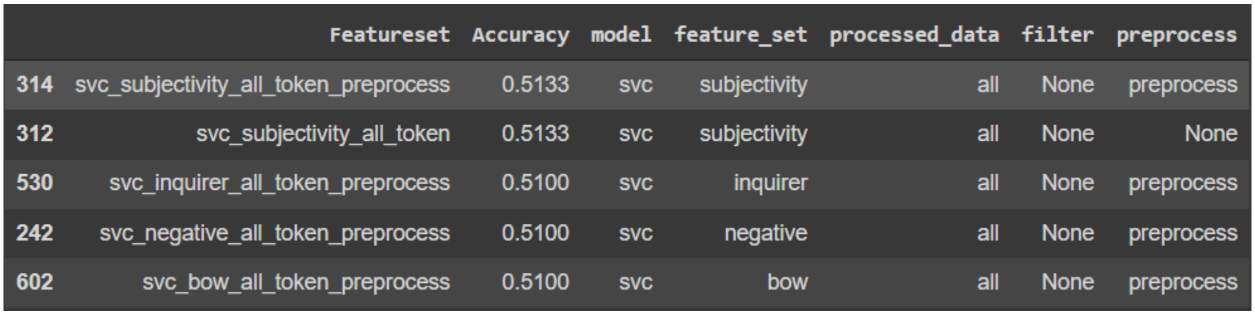


	Featureset	Accuracy	model	feature_set	processed_data	filter
90	randomforest_bigram_all_token_preprocess	0.5033	randomforest	bigram	all	None
88	randomforest_bigram_all_token	0.5033	randomforest	bigram	all	None
376	randomforest_liwc_all_token	0.5033	randomforest	liwc	all	None
378	randomforest_liwc_all_token_preprocess	0.5033	randomforest	liwc	all	None
450	randomforest_subjectivityliwc_all_token_prepro...	0.5033	randomforest	subjectivityliwc	all	None

Using the Random Forest classifier with a bigram featureset, all data and no filtering, the accuracy we obtained was 0.5033.

SVC: the Support Vector Classifier (SVC) works by drawing a line between the different clusters of data points to group them into classes. To increase confidence in the class allocation, the classifier will try to maximize the distance between the line it draws and the points on either side of it.

Top 5 Results:

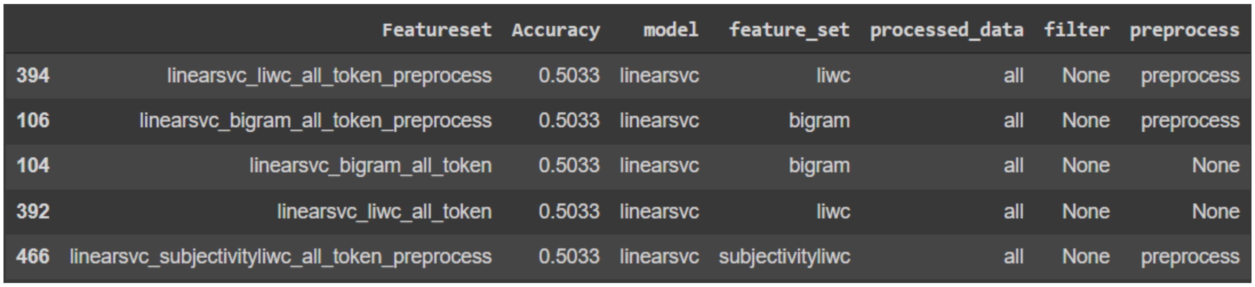


	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
314	svc_subjectivity_all_token_preprocess	0.5133	svc	subjectivity	all	None	preprocess
312	svc_subjectivity_all_token	0.5133	svc	subjectivity	all	None	None
530	svc_inquirer_all_token_preprocess	0.5100	svc	inquirer	all	None	preprocess
242	svc_negative_all_token_preprocess	0.5100	svc	negative	all	None	preprocess
602	svc_bow_all_token_preprocess	0.5100	svc	bow	all	None	preprocess

With the SVC model, using the subjectivity featureset, all data, no filtering and pre-processed data, we got an accuracy of 0.5133.

Linear SVC: this method applies a linear kernel function to perform classification and it performs well with a large number of samples. If we compare it with the SVC model, the Linear SVC has additional parameters such as penalty normalization which applies 'L1' or 'L2' and loss function. The kernel method can not be changed in linear SVC, because it is based on the kernel linear method.

Top 5 Results:

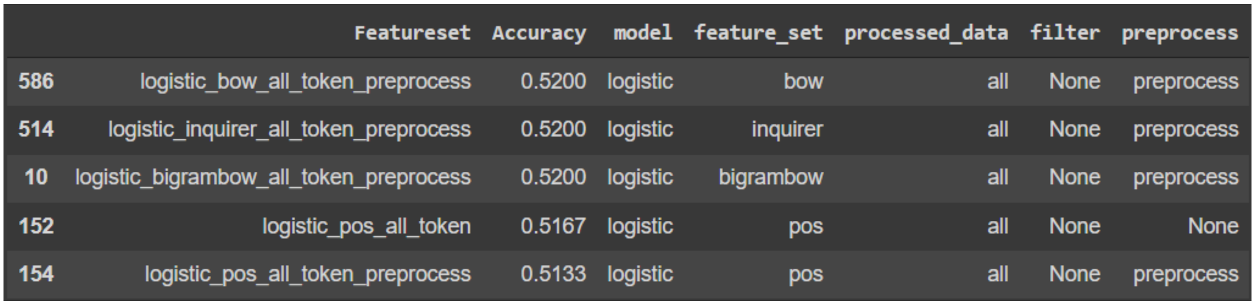


	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
394	linearsvc_liwc_all_token_preprocess	0.5033	linearsvc	liwc	all	None	preprocess
106	linearsvc_bigram_all_token_preprocess	0.5033	linearsvc	bigram	all	None	preprocess
104	linearsvc_bigram_all_token	0.5033	linearsvc	bigram	all	None	None
392	linearsvc_liwc_all_token	0.5033	linearsvc	liwc	all	None	None
466	linearsvc_subjectivityliwc_all_token_preprocess	0.5033	linearsvc	subjectivityliwc	all	None	preprocess

With pre-processed data, no filter and the LIWC featureset, the linear SVC model gave us a 0.5033 accuracy.

Logistic Regression: used for binary classification tasks (yes/no, spam/not spam) the logistic regression picks a threshold and decides where the selected value falls on either side of it.

Top 5 Results:

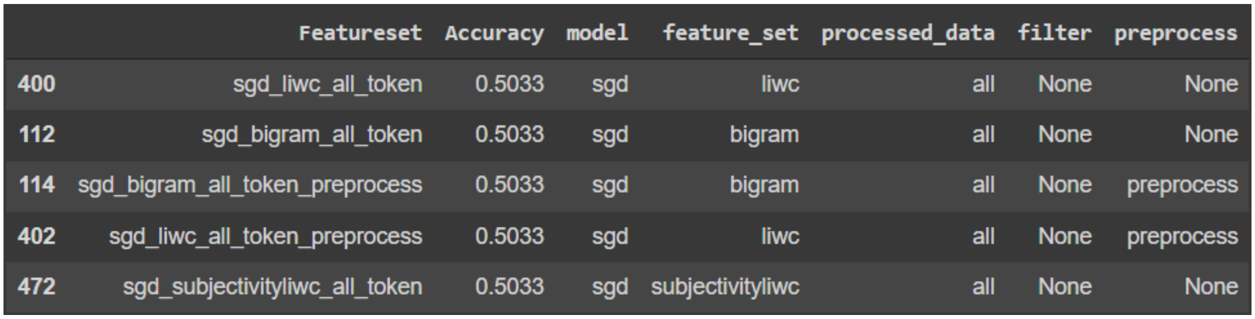


	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
586	logistic_bow_all_token_preprocess	0.5200	logistic	bow	all	None	preprocess
514	logistic_inquirer_all_token_preprocess	0.5200	logistic	inquirer	all	None	preprocess
10	logistic_bigrambow_all_token_preprocess	0.5200	logistic	bigrambow	all	None	preprocess
152	logistic_pos_all_token	0.5167	logistic	pos	all	None	None
154	logistic_pos_all_token_preprocess	0.5133	logistic	pos	all	None	preprocess

With the Logistic Regression classifier, we had a 0.52 accuracy using the BoW featureset with all data and no filtering but with pre-processing.

SGD: the Stochastic Gradient Descent approach consists in fitting linear classifiers and regressors under convex loss functions by implementing a plain stochastic gradient descent learning routine.

Top 5 Results:

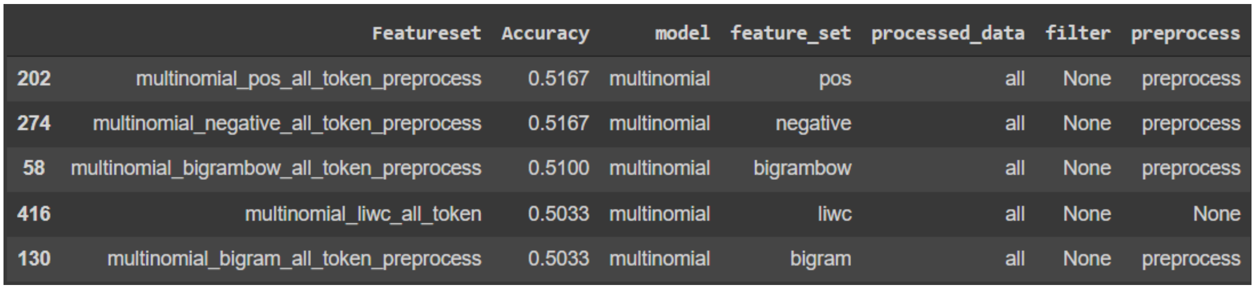


	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
400	sgd_liwc_all_token	0.5033	sgd	liwc	all	None	None
112	sgd_bigram_all_token	0.5033	sgd	bigram	all	None	None
114	sgd_bigram_all_token_preprocess	0.5033	sgd	bigram	all	None	preprocess
402	sgd_liwc_all_token_preprocess	0.5033	sgd	liwc	all	None	preprocess
472	sgd_subjectivityliwc_all_token	0.5033	sgd	subjectivityliwc	all	None	None

With no pre-processed data, no filter and the LIWC featureset, the SGD model gave us a 0.5033 accuracy.

Multinomial Naive Bayes: this classifier guesses the tag of a text, such as an email or a newspaper story, using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance.

Top 5 Results:

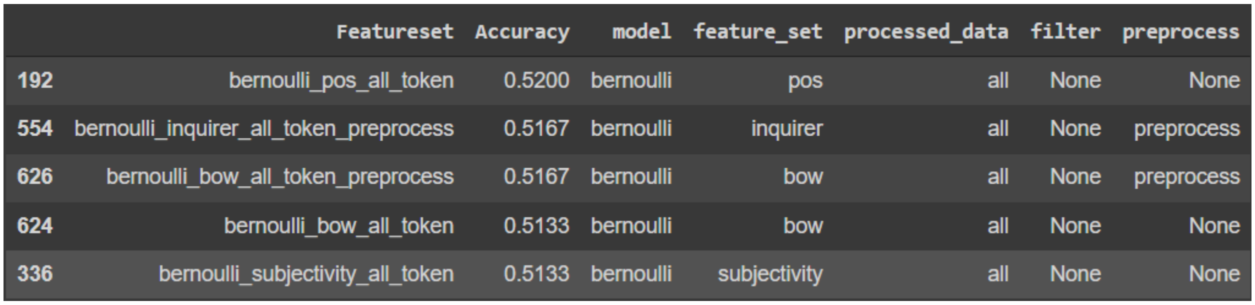


	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
202	multinomial_pos_all_token_preprocess	0.5167	multinomial	pos	all	None	preprocess
274	multinomial_negative_all_token_preprocess	0.5167	multinomial	negative	all	None	preprocess
58	multinomial_bigrambow_all_token_preprocess	0.5100	multinomial	bigrambow	all	None	preprocess
416	multinomial_liwc_all_token	0.5033	multinomial	liwc	all	None	None
130	multinomial_bigram_all_token_preprocess	0.5033	multinomial	bigram	all	None	preprocess

With the Multinomial Naive Bayes classifier, we had a 0.5167 accuracy using the PoS featureset with all data and no filtering or pre-processing.

BernoulliNB: a Naive Bayes classifier, based on the Bernoulli distribution, that’s used for discrete data, where features are only in binary form.

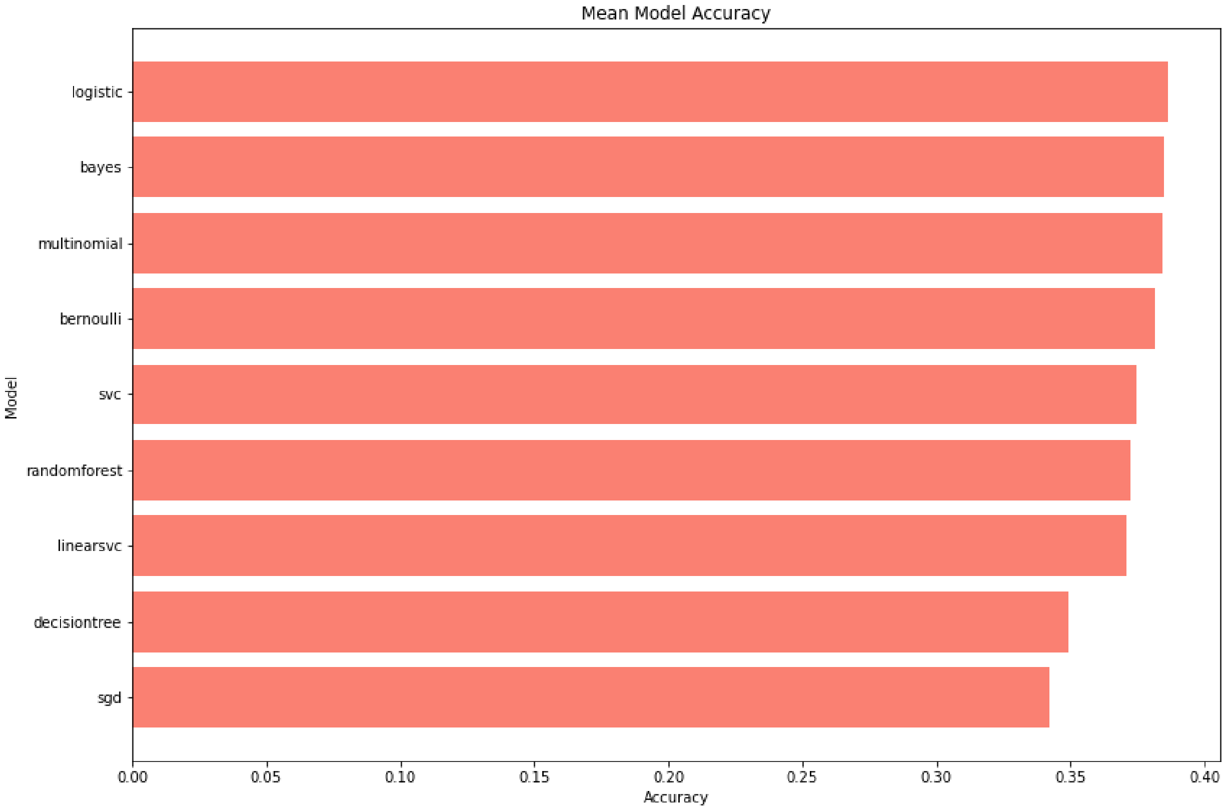
Top 5 Results:



	Featureset	Accuracy	model	feature_set	processed_data	filter	preprocess
192	bernoulli_pos_all_token	0.5200	bernoulli	pos	all	None	None
554	bernoulli_inquirer_all_token_preprocess	0.5167	bernoulli	inquirer	all	None	preprocess
626	bernoulli_bow_all_token_preprocess	0.5167	bernoulli	bow	all	None	preprocess
624	bernoulli_bow_all_token	0.5133	bernoulli	bow	all	None	None
336	bernoulli_subjectivity_all_token	0.5133	bernoulli	subjectivity	all	None	None

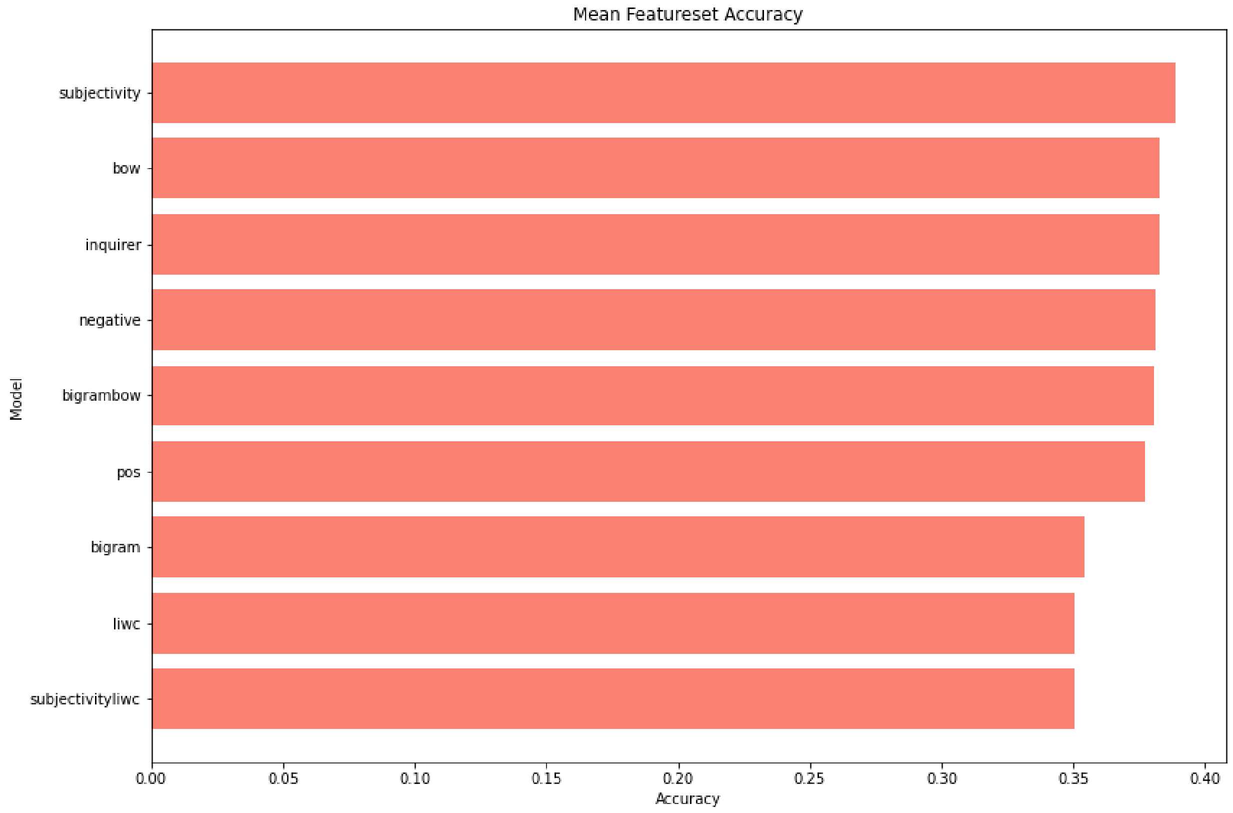
With the Bernoulli Naive Bayes classifier, we had a 0.52 accuracy using the PoS featureset with all data and no filtering or pre-processing.

Model Comparison:



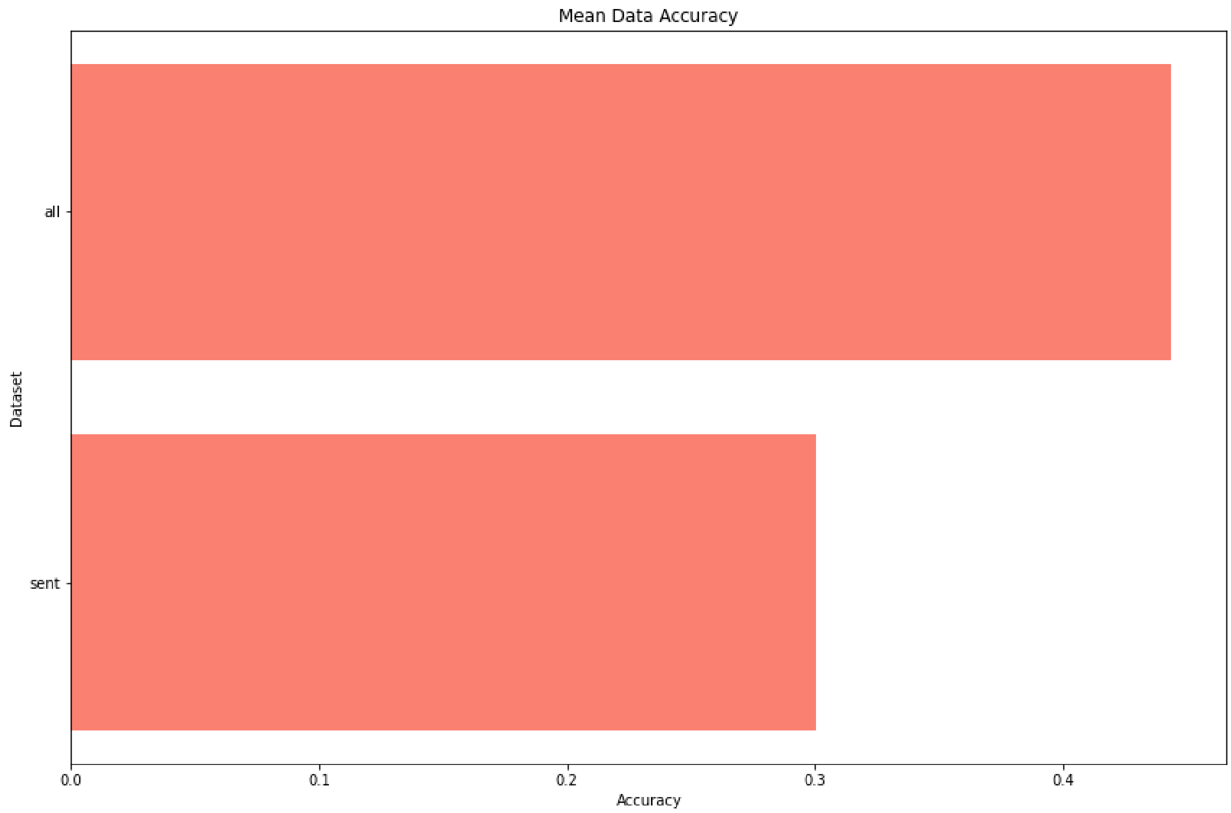
In a model comparison, the logistic regression model (0.386) barely edged the three Bayesian models - Naive Bayes (0.385), Bernoulli NB (0.384) and multinomial NB(0.382).

Featureset Comparison:



In featureset comparison, the subjectivity model has an accuracy of 0.389, better than the 0.383 accuracies of the BoW and Harvard inquirer.

Dataset Comparison:



In a comparison of the datasets, the ‘all’ dataset had a far better accuracy (0.44) than the ‘sent’ dataset (0.3).

Cross-Validation for Top 3 Models and Featuresets:

Cross-validation: it is a technique used to protect against overfitting in a predictive model, particularly in a case where the amount of data may be limited. In cross-validation, you make a fixed number of folds (or partitions) of the data, run the analysis on each fold, and then average the overall error estimate.

Precision is a measure of how many of the positive predictions made are correct (true positives), accuracy describes the number of correct predictions over all predictions, recall is a measure of how many of the positive cases the classifier correctly predicted over all the positive cases in the data and the F1-score is a measure combining both precision and recall that basically weighs the two rations in a balanced way.

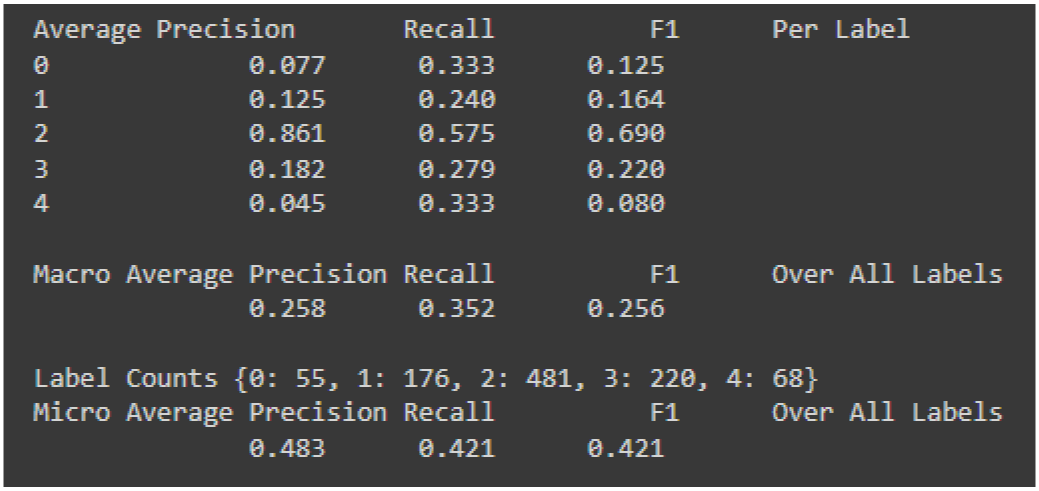
The best model and dataset:

Model: Logistic regression

Featureset: Bigram BOW

Data: All data

The Cross Validation results:



```
Average Precision       Recall          F1      Per Label
0              0.077      0.333      0.125
1              0.125      0.240      0.164
2              0.861      0.575      0.690
3              0.182      0.279      0.220
4              0.045      0.333      0.080

Macro Average Precision Recall          F1      Over All Labels
               0.258      0.352      0.256

Label Counts {0: 55, 1: 176, 2: 481, 3: 220, 4: 68}
Micro Average Precision Recall          F1      Over All Labels
               0.483      0.421      0.421
```

The cross validation results indicated a macro average precision of 0.258, a recall of 0.352, and a F1 score of 0.256 for the best model (logistic regression) and featureset (BoW bigram) combination. Although this model and featureset combination was the best in accuracy, the precision and F1 scores were not the best out of the three.

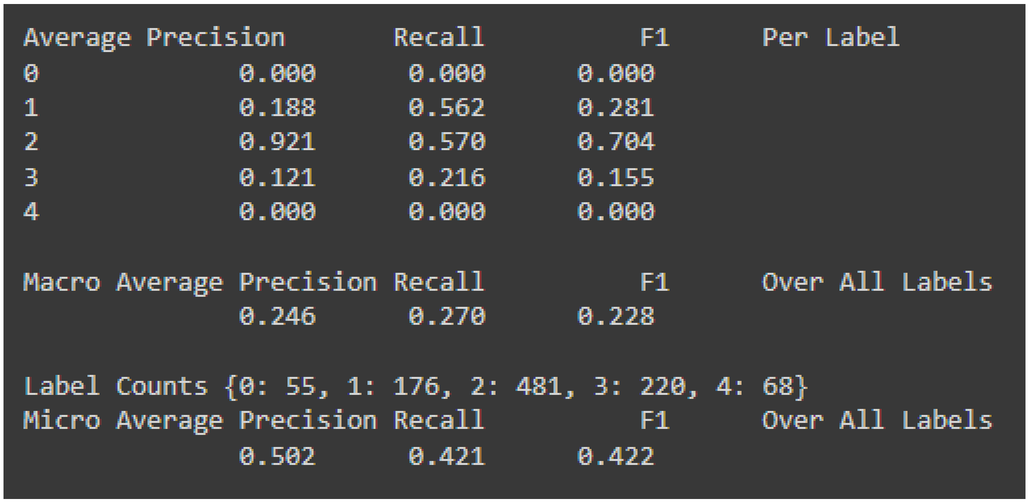
Second best model and dataset:

Model: Bernoulli

Featureset: PoS

Data: All data

The Cross Validation results:



```
Average Precision       Recall          F1      Per Label
0             0.000      0.000      0.000
1             0.188      0.562      0.281
2             0.921      0.570      0.704
3             0.121      0.216      0.155
4             0.000      0.000      0.000

Macro Average Precision Recall          F1      Over All Labels
              0.246      0.270      0.228

Label Counts {0: 55, 1: 176, 2: 481, 3: 220, 4: 68}
Micro Average Precision Recall          F1      Over All Labels
              0.502      0.421      0.422
```

The cross validation results indicated a macro average precision of 0.246, a recall of 0.270, and a F1 score of 0.228 for the second best model and featureset combination. The precision and recall scores came in the worst out of the three models and featuresets, with a significantly worse recall score than the other two models and featuresets.

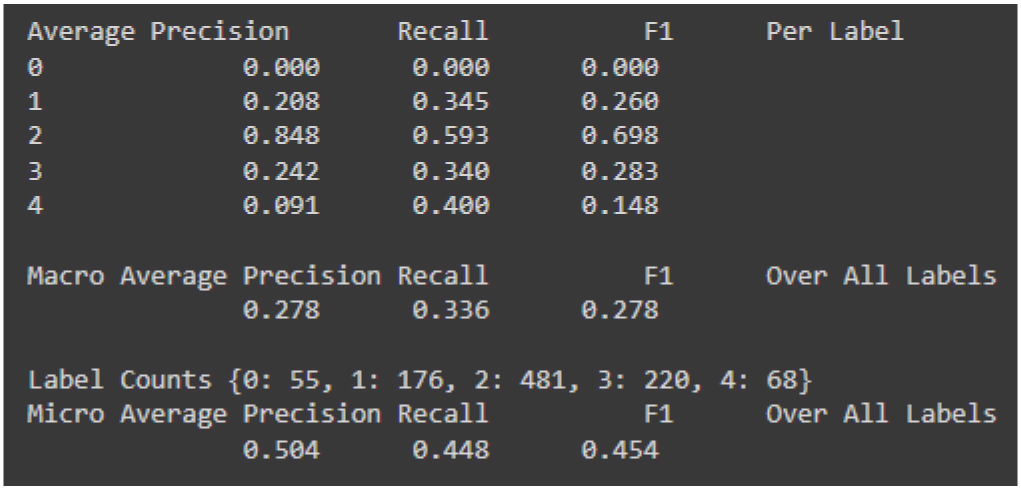
Third best model and dataset:

Model: Logistic regression

Featureset: Harvard Inquirer IV-4

Data: All data

The Cross Validation results:

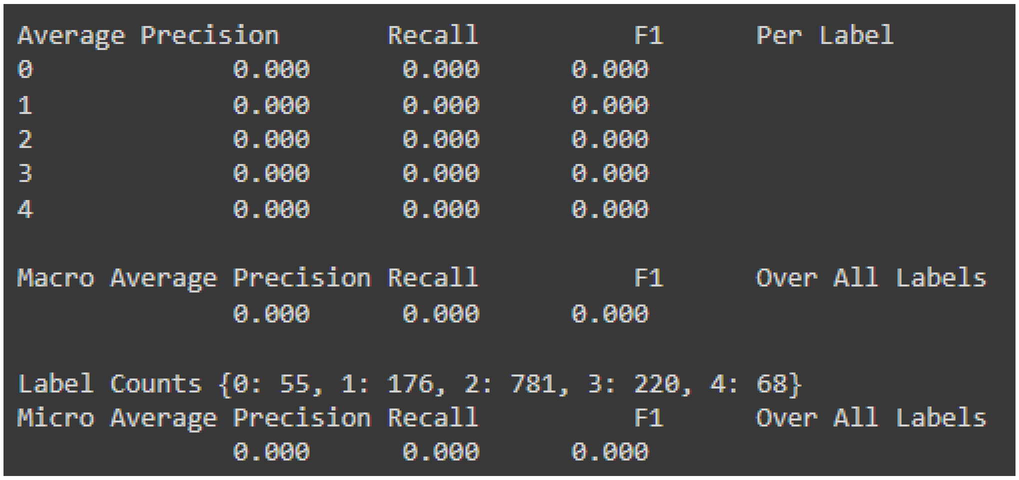


```
Average Precision       Recall          F1      Per Label
0             0.000      0.000      0.000
1             0.208      0.345      0.260
2             0.848      0.593      0.698
3             0.242      0.340      0.283
4             0.091      0.400      0.148

Macro Average Precision Recall          F1      Over All Labels
              0.278      0.336      0.278

Label Counts {0: 55, 1: 176, 2: 481, 3: 220, 4: 68}
Micro Average Precision Recall          F1      Over All Labels
              0.504      0.448      0.454
```

The cross validation results indicated a macro average precision of 0.278, a recall of 0.336, and a F1 score of 0.278 for the third best model and featureset combination. Although this model and featureset was the third best in accuracy, the precision and F1 scores were the best out of the three.

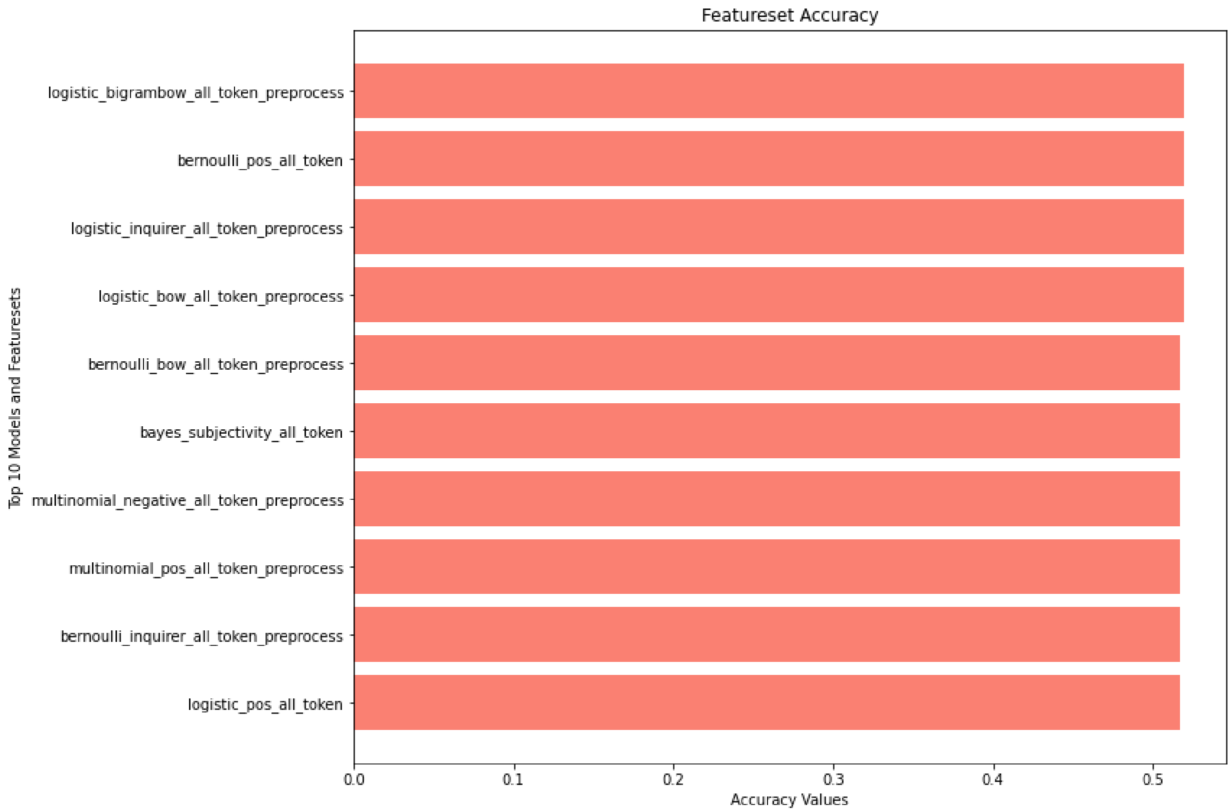
Cross-Validation for Testing Data:

The cross validation scores resulted in 0 for each of the three scores when training the logistic model with the best featureset and trying to predict the testing featureset. With every rating having a sentiment of 2 for the testing data, the model trained on a variety of sentiment scores using the training data could not accurately predict the sentiment score of 2 for the testing data given the bigram and BoW featureset built using the testing data.



```
Average Precision       Recall          F1      Per Label
0             0.000      0.000      0.000
1             0.000      0.000      0.000
2             0.000      0.000      0.000
3             0.000      0.000      0.000
4             0.000      0.000      0.000

Macro Average Precision Recall          F1      Over All Labels
              0.000      0.000      0.000

Label Counts {0: 55, 1: 176, 2: 781, 3: 220, 4: 68}
Micro Average Precision Recall          F1      Over All Labels
              0.000      0.000      0.000
```

Conclusion



The findings from this analysis highlight the difficulties in classifying movie reviews based solely on the text alone. With the model results, the highest accuracy was 52 percent and resulted in a four-way tie. Each of the top four models relied on processing the full data and three out of the four benefited from the use of pre-processing with the fourth not requiring any pre-processing or filtering. The top featuresets for the model outputs were the bigram BoW, PoS, Inquirer, and the BoW. Logistic regression came out as the top model, holding three out of the top four spots with Bernoulli being the last. As hypothesized in the introduction, the full dataset resulted in higher accuracy scores with only using the primary sentence with each review showing abysmal results nearly 20 percent lower.The top three featuresets by average model accuracies resulted in subjectivity being the best, followed by BoW and Inquirer.

Course Reflection

The NLP course provided a challenge for developing a unique and valuable skillset in a rising branch of data science designed to increase knowledge for the logistical properties in evaluating unstructured text data. NLTK as the base library for analysis requires users to define the text processing methods needed in generating insight from text mining. The final project in the course pulls elements from each lesson using a real-world example based on evaluating customer reviews for an organization. The project highlighted some important aspects to keep in mind when developing an NLP model:

- Understand the text and methods needed for processing that text is integral to developing a solution. Having the end-state in mind is a necessary part before construction the solution and will save time in completing a project.
- Identify the scale of the project prior to starting to set limitations in text analysis to improve on the ability to properly estimate the time needed to complete the project.

- Experimenting with a variety of feature sets and models is necessary in getting to the best result. This sentiment holds true across all aspects of data analysis and evaluating unstructured data holds no differences in that regard.

This course and project fulfill the requirements for six out of the seven portfolio learning objectives through analyzing textual reviews from customers.

- **Describe a broad overview of the major practice areas in data science:**
  NLP is an emerging focus area in data science and follows a similar path in terms of the data science lifecycle. As more companies rely on NLP analysis, the field will continue to grow through the addition of new techniques and text processing pipelines.
- **Collect and organize data:**
  Text mining has a heavy requirement in effectively collecting and organizing data. Due to the unstructured nature of text, processing and filtering out unnecessary words or grammar will elevate the subsequent analysis.
- **Identify patterns in data via visualization, statistical analysis, and data mining:**
  The use of statistical analysis is imperative to the analysis and allows for comparing the frequency between sentiments. Visualizing the frequency and distribution of the reviews helps in eliminating certain words that provide no added benefit in the analysis.
- **Develop alternative strategies based on the data:**
  Unstructured data requires multiple strategies for analysis that include developing alternative feature sets or relying on several models. Relying on one technique to evaluate customer reviews limits the successful analysis for the project.
- **Demonstrate communication skills regarding data and its analysis for relevant professionals in their organization:**

The project exemplifies the need for effective communication and trust within an organization. Although these reviews focused on movies, companies often rely on NLP to evaluate customer sentiment on products or services. These reviews could focus on a variety of business processes and every professional in the organization from order fulfillment to sales, and up through to decision-makers might need the results obtained from data analysis.

- **Synthesize the ethical dimensions of data science practice:**

  The ethics behind evaluating customer sentiment require great care both in ensuring bias does not affect the analysis and to ensure customers remain anonymous for protecting their privacy.

Course Highlight 3: Applied Machine Learning for Data Analysis

Course Description

The Applied Machine Learning course is an introduction to data mining techniques through getting familiar with real-world applications, challenges involved in these applications, and future directions of the field while getting hands-on experience with open-source software packages. This course covers popular data mining methods for extracting knowledge from data and include the principles and theories of data mining methods and applying data mining to problems. The focus of this course is to understand data and how to formulate data mining tasks to solve problems with machine learning. The course also includes understanding the key tasks of data mining, including data preparation, concept description, association rule mining, classification, clustering, and data analysis (Block, 2021).

Learning Objectives

- Document, analyze, and translate data mining needs into technical designs and solutions.
- Apply data mining concepts, algorithms, and evaluation methods to real-world problems.
- Employ data storytelling and dive into the data, find useful patterns, and articulate how to find patterns and explain why the patterns are valuable and trustworthy.

Project Requirements

Define a problem on the dataset and describe it in terms of its real-world organizational or business application. The problem may use one or more of the types of data mining algorithms: Classification, Clustering and Association Rules, in an investigation of the solution to the problem. This investigation must include some aspects of experimental comparison between different algorithms or classifiers and some experiments with tuning parameters of the algorithms. If there are a larger number of attributes, the use of feature selection will help in reducing the number of attributes. Use summary statistics and visualization techniques to help explain the findings and justify the choice of algorithms.

Project Development

Introduction

Recommender systems provide the backbone of systems designed to both enhance user workflow and to increase viewers for the overall platform. Using the Netflix data, successful models relying on various techniques help in building out a recommendation network that includes predicting the IMDb score and parental ratings for suggested viewing. When used in conjunction with Association Rule Mining techniques for genre, the recommender system will have robust capabilities that include a variety of models for people viewing everything from G to R ratings.

Four primary techniques make up the base for conducting analysis, exploring the data set, and building a predictive model that centers on the IMDb score and a parental rating classifier. Clustering and Classification on the description illuminate the parental rating and serve as a basis for comparing different descriptions. Association Rule Mining will help suggest what genre of movies or TV shows a viewer should watch next based on their most recently viewed. Identifying predictive solutions using Movie Length, Genre, Parental Rating, and TV Show or Movie as part of a Support Vector Machine (SVM), KNN, Random Forest, or Decision Tree model.

About the Data

The Netflix data set came from Satpreet Makhija (2021) on Kaggle, and it contains all the movies and TV shows from Netflix in 2021. The overall data is contained within the following columns: Description, Director, Genre, Cast, Rating, Duration and IMDb Score. Discretization of the parental ratings included combining ratings that aligned with certain age groups, like TV-MA and R. Although these have varying definitions for who should and should not be watching, the suggested age groups were close enough to combine together with a reasonable assumption that parents would not balk at the differences between G versus Y-7 or PG-13 versus TV-14.

Cleaning Data

Filtering the original data included removing some columns and limiting the data based on location and type. As a United States-focused recommender system, the data includes all production studios that shot scenes specifically within the United States at some point in the process. Additionally, the data contains no blank values wherever data was missing for director and cast records. Having complete data for those variables could be an important part of future modifications to the recommender system that rely on using specific directors or cast members. Additionally, the data only contains movies as TV shows have varying data for the cast, directors, and other key variables that went into predicting IMDb score like duration.

Discretization and cleaning of the data included the following focus areas:

- Removing “/10” from the IMDb score column.
- Removing “min” from the duration column.
- Removing the Date Added column.
- Putting the length of movie in 30 minute bins.

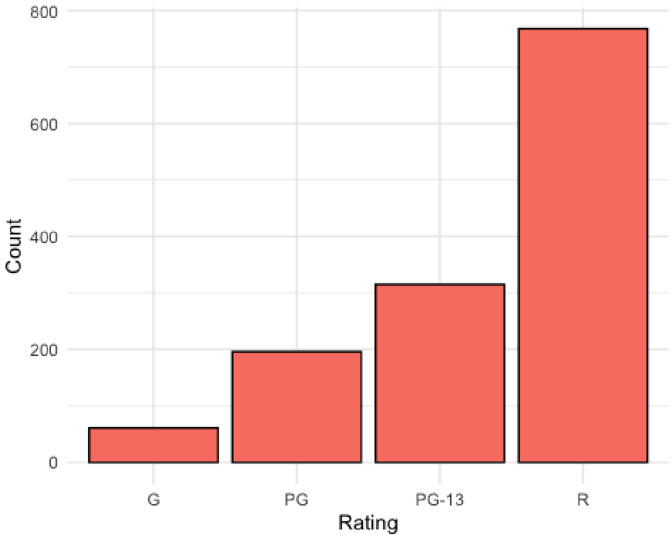
- Creating an International flag field for production country that sets a value of 0 for movies that only include the United states.
- Cleaning up the rating by combining the TV ratings and movie ratings in the following format:
  - TV-Y transformed to G.
  - TV-Y7 transformed to PG.
  - TV-14 transformed to PG13.
  - TV-MA transformed to R.

The following output is the structure of the data, including variable types and field names:

```
str(CleanFlix)

## tibble [1,340 × 14] (S3: tbl_df/tbl/data.frame)
##  $ Show Id          : chr [1:1340] "c844460f-6178-4f87-929e-80816c74ca35" "0e5fc89e-be6a-
44d1-9923-533f117e46c3" "775f89b9-8fa1-479a-8837-d039d524da39" "60cac8ed-cbe1-4bca-a2
2b-ebcf7b8ac920" ...
##  $ Title            : chr [1:1340] "#realityhigh" "1922" "1BR" "2 Hearts" ...
##  $ Description      : chr [1:1340] "When nerdy high schooler Dani finally attracts the interest
of her longtime crush, she lands in the cross hairs"| __truncated__ "A farmer pens a confession a
dmitting to his wife's murder, but her death is just the beginning of a macabre tal"| __truncated__
"Seeking her independence, a young woman moves to Los Angeles and settles into a cozy apart
ment complex with a d"| __truncated__ "In parallel love stories, the lives of college student Chris
and wealthy businessman Jorge intersect in a profo"| __truncated__ ...
##  $ Director         : chr [1:1340] "Fernando Lebrija" "Zak Hilditch" "David Marmor" "Lance H
```

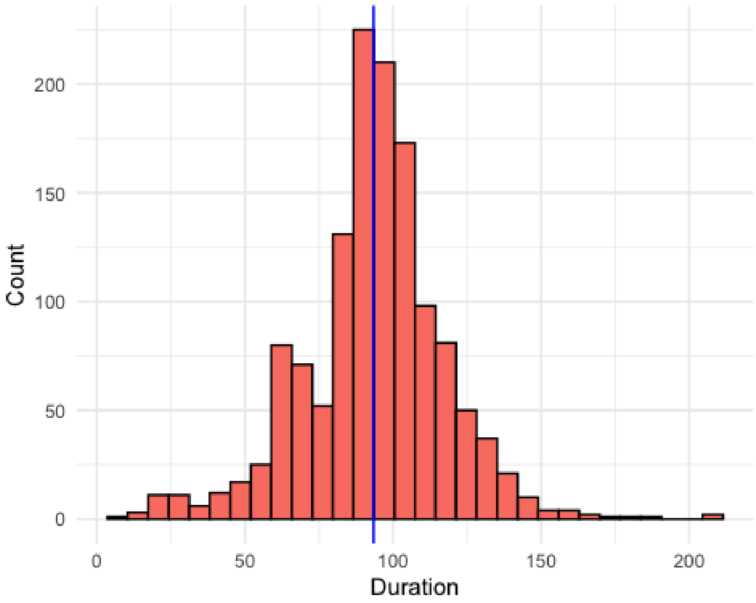
```
ool" ...
##  $ Genres            : chr [1:1340] "Comedies" "Dramas, Thrillers" "Horror Movies, Independent
Movies, Thrillers" "Dramas, Faith & Spirituality, Romantic Movies" ...
##  $ Cast              : chr [1:1340] "Nesta Cooper, Kate Walsh, John Michael Higgins, Keith Powe
rs, Alicia Sanz, Jake Borelli, Kid Ink, Yousef Erakat"| __truncated__ "Thomas Jane, Molly Parke
r, Dylan Schmid, Kaitlyn Bernard, Bob Frazer, Brian d'Arcy James, Neal McDonough" "Nicole
Brydon Bloom, Giles Matthey, Taylor Nichols, Alan Blumenfeld, Celeste Sully, Susan Davis, Cl
ayton Hoff, "| __truncated__ "Jacob Elordi, Adan Canto, Radha Mitchell, Tiera Skovbye, Kari M
atchett, Tahmoh Penikett" ...
##  $ Production Country: chr [1:1340] "United States" "United States" "United States" "United
States" ...
##  $ Release Date      : num [1:1340] 2017 2017 2019 2020 2009 ...
##  $ Rating            : chr [1:1340] "PG-13" "R" "R" "PG-13" ...
##  $ Duration          : num [1:1340] 99 103 90 101 158 144 117 92 91 92 ...
##  $ Imdb Score        : num [1:1340] 5.1 6.4 5.7 5.9 6 6.6 6.4 5.8 4.7 6.1 ...
##  $ Content Type      : chr [1:1340] "Movie" "Movie" "Movie" "Movie" ...
##  $ Duration_bins     : Ord.factor w/ 7 levels "30"<"60"<"90"<..: 4 4 3 4 6 5 4 4 4 4 ...
##  $ internation_flag  : logi [1:1340] FALSE FALSE FALSE FALSE FALSE TRUE ...
```

Exploratory Data Analysis



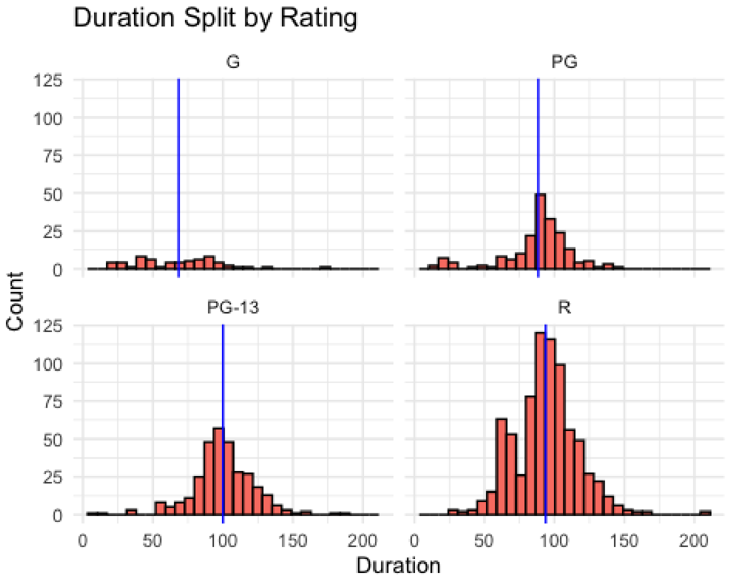
```
## [1] "Total Number of R Movies: 768"

## [1] "Total Number of PG-13 Movies: 315"

## [1] "Total Number of PG Movies: 196"

## [1] "Total Number of G Movies: 61"
```

Looking at the count of movies with each rating shows a disparaging difference. Movies that fall within the “R” category represent over 57 percent of all movies and could skew results in favor of that rating. PG-13 movies make up just over 23 percent of the total with PG lagging behind at just over 14 percent and G at just over 4 percent of the total movies.



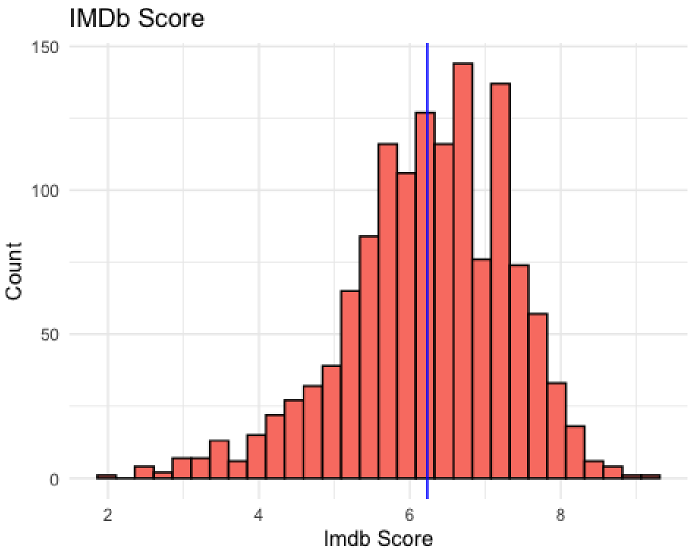
```
## [1] "Average Duration in Minutes: 93.4440298507463"
```

The duration of movies has an interesting shape close to a normal distribution. The average time of movies just over 90 minutes shows a goal for most lengths at about the hour and a half mark with some outliers that run over 200 minutes in length.



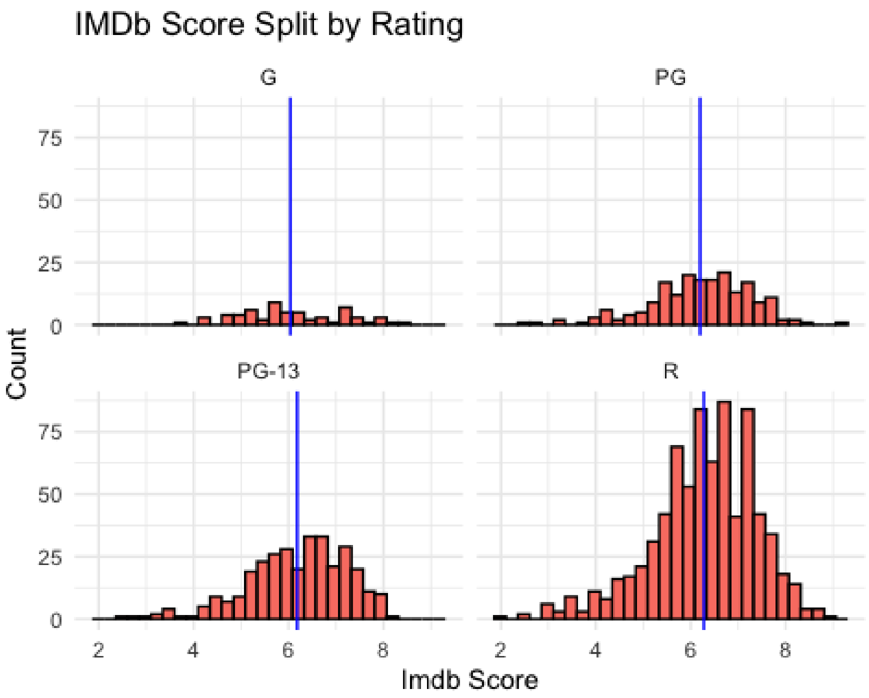
```
## [1] "Mean Duration for R Movies: 93.97265625"
## [1] "Mean Duration for PG-13 Movies: 100.047619047619"

## [1] "Mean Duration for PG Movies: 88.5051020408163"

## [1] "Mean Duration for G Movies: 68.5573770491803"
```

Taking a closer look at the duration by parental rating indicates PG-13 movies have the longest run time, followed closely by R and PG. Movies with a G parental rating tend to be on the shorter side of the overall count, possibly due to the attention span of the audience.



```
## [1] "Average IMDb Score: 6.23044776119403"
```

The distribution of IMDb Scores has a negative skew with movies having more favorable scores with outliers approaching a score of 2 and 9 with a scale of 0 to 10. Based on the average IMDb around 6.2, the scale of IMDb scores indicates an imbalance where an expectation for movie averages would fall around the 5 mark.

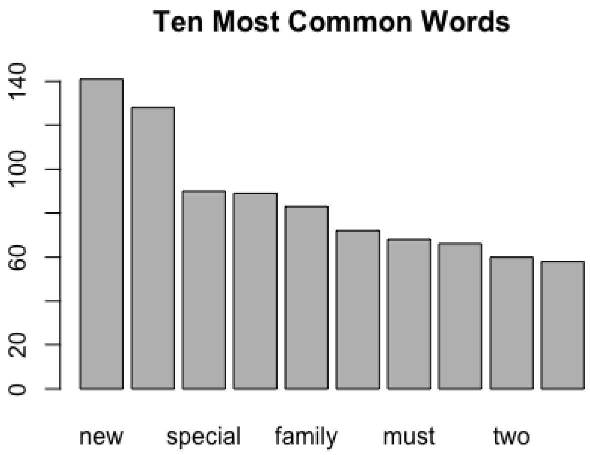


```
## [1] "Mean IMDb Score for R Movies: 6.27330729166667"

## [1] "Mean IMDb Score for PG-13 Movies: 6.1847619047619"

## [1] "Mean IMDb Score for PG Movies: 6.19540816326531"

## [1] "Mean IMDb Score for G Movies: 6.03934426229508"
```

The first indication of possibly favoring R movies due to the sheer number within the data shows when comparing the IMDb scores between ratings. R movies average an IMDb score of 6.27, the highest out of the other movie ratings.

Alongside with looking at the base counts, the description of each of the movies could help with identifying the parental rating for future movies entering the recommender system. The first step includes pulling out the description for each movie and vectorizing the words with normalizing the frequencies between each movie.

Clustering



```
## [1] "Avergae Number of Words Per Description: 14"
```

After removing all common words, stopwords, punctuation, numbers, and whitespace, the average number of words for each description comes out to 14 total words. This might limit the success of using clustering to identify trends in the data that could help with sorting movies between various categories. The goal is correctly identifying parental rating, but clustering will also show if there is anything within the data that could sort the movies. The ten most common words highlight the sentiment for overall movies on Netflix with “new” taking the top spot, followed closely by “life”.

The next step in the clustering process identifies the best number of clusters to use based on measured values of cohesion within clusters and separation between clusters. Looking for values with higher separation and lower cohesion will highlight the best number of clusters for the data if any sorted groups form. Looking at clusters between 2 and 20 should allow for the best distribution of clusters without worrying about having too many clusters for the data.

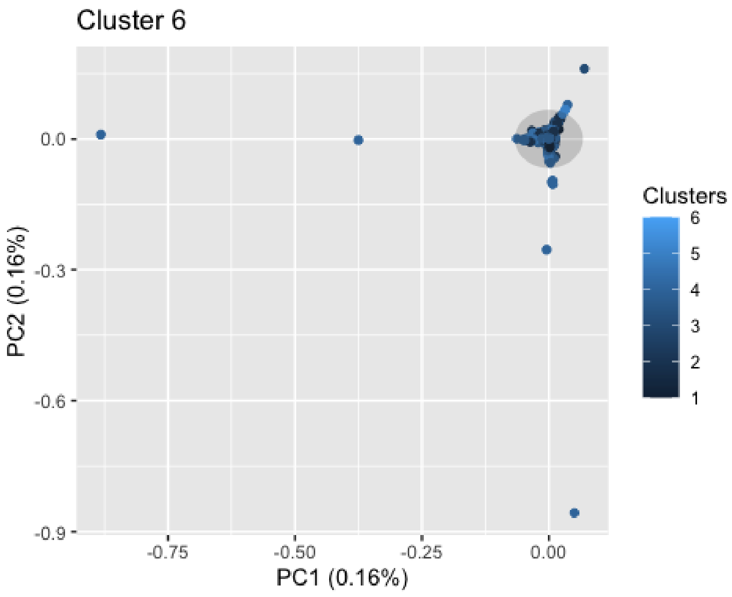
The following is the distribution of cohesion and separation for the top five clusters:

Best Cluster Results

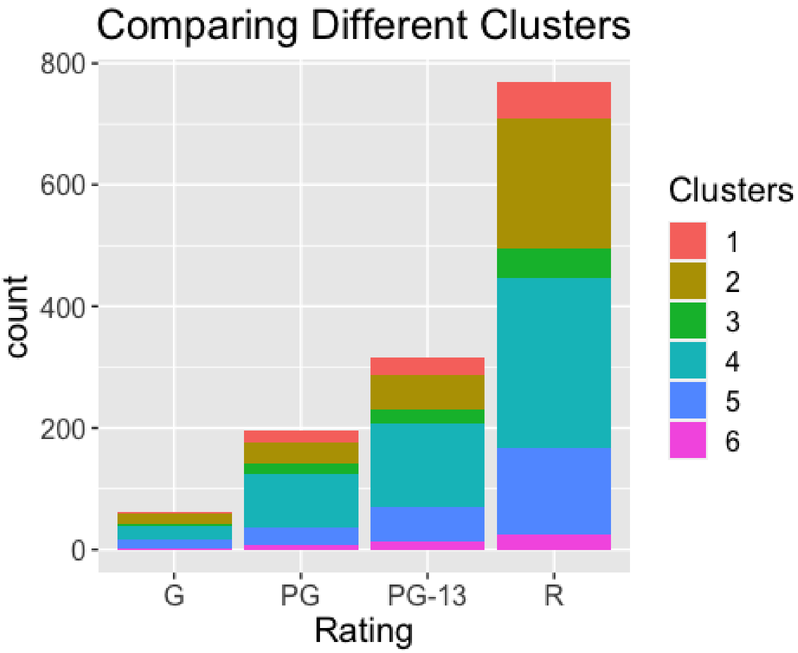
```
##    Number     Cohesion   Separation ScaleCohesion ScaleSeparation     Combined
## 1       2 0.0003949957 3.197654e-06     1.6972491      -1.6972491 3.774758e-15
## 3       4 0.0003917749 6.418429e-06     1.2243561      -1.2243561 3.774758e-15
## 4       5 0.0003899077 8.285574e-06     0.9502111      -0.9502111 3.774758e-15
## 5       6 0.0003910025 7.190776e-06     1.1109556      -1.1109556 3.774758e-15
## 12     13 0.0003808130 1.738034e-05    -0.3851354       0.3851354 3.774758e-15
```

The best number of clusters after scaling the cohesion and separation for each of the clusters tested and adding them together came out to 6 clusters.

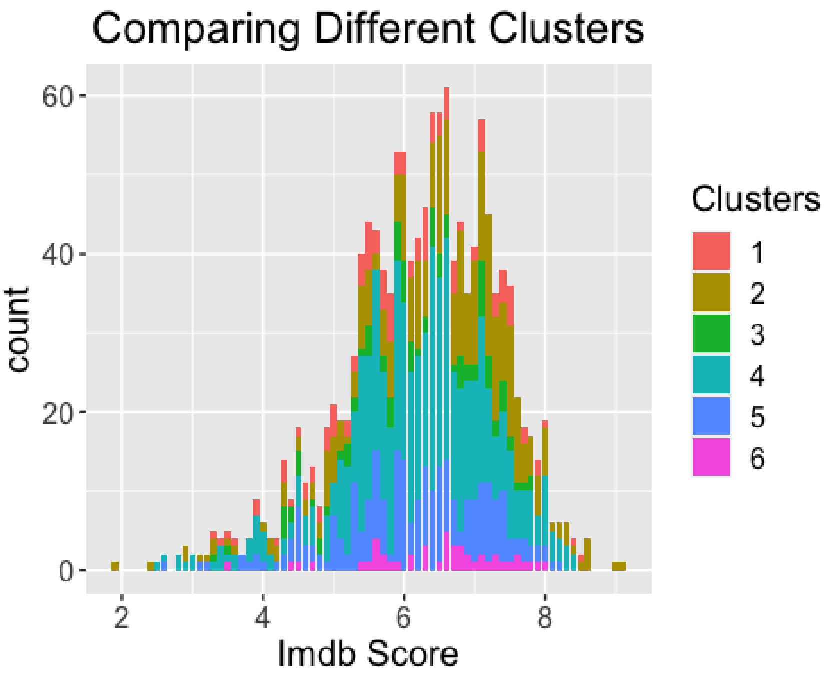
The next step is running through the data with 6 clusters using the kmeans function on the normalized word frequencies.



The graph of 6 clusters ends up looking pretty similar to the rest of the cluster amounts, even though it was the best in terms of cohesion and separation. The low frequency of words in the description with an average of 14 unique words per description highlights the limitations of relying on clustering to group this data.



Looking at the bar chart next specifically for the goal of grouping parental rating shows a spread of parental values for all ratings. Ideally, the clustering would show each cluster falling within a specific rating.

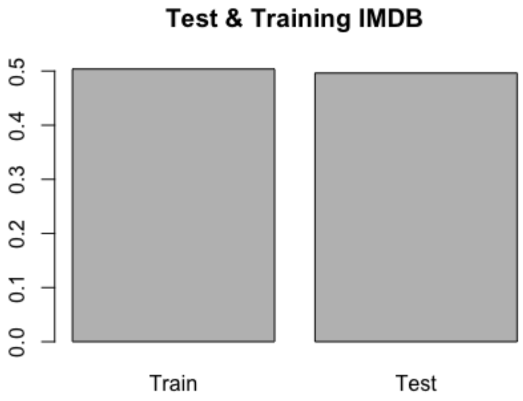


As a point of interest, clustering also shows difficulty in identifying movies with all IMDb scores. Although the results do not indicate clustering as a viable method, forcing the goal of finding parental rating from description with classification may be possible.

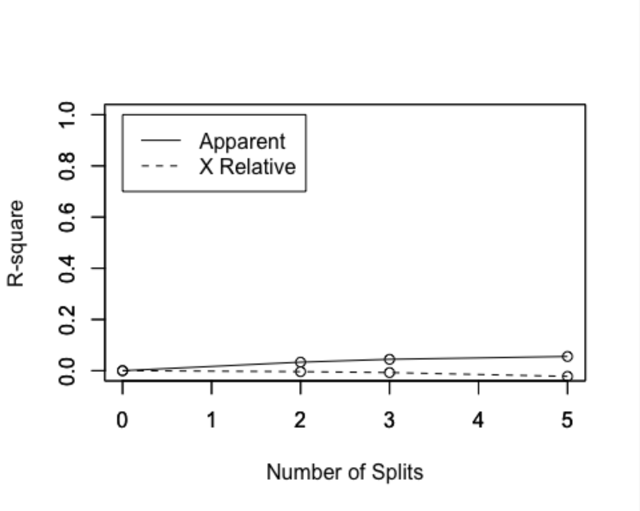
Classification

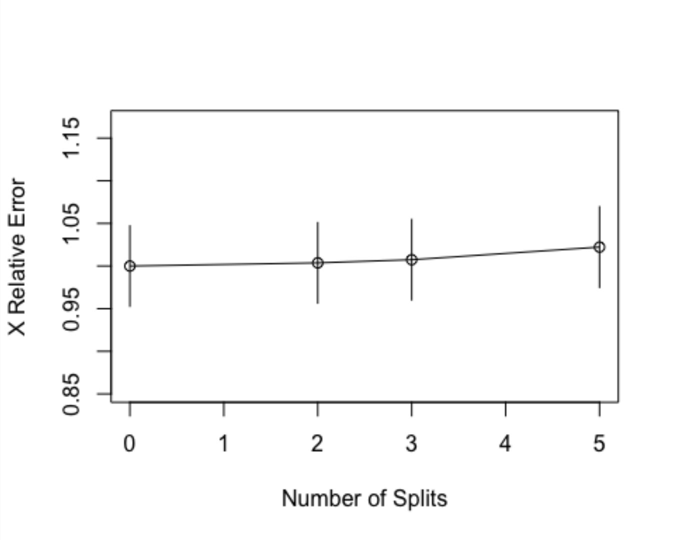
Next steps is to break the data into a Test and Train set to be about 50% for both groups.

```
## Train  Test
##   675   665
```



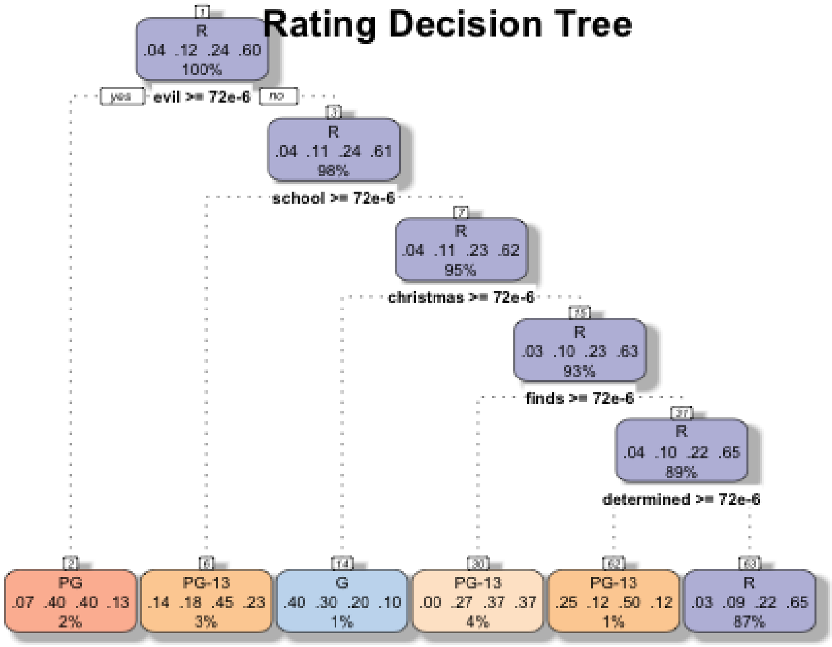
```
##

## Classification tree:

## rpart(formula = Rating ~ ., data = TrainClassFlix, method = "class",

##     control = rpart.control(cp = 0, maxdepth = 5))

##

## Variables actually used in tree construction:

## [1] christmas  determined evil       finds      school

##

## Root node error: 270/675 = 0.4

##

## n= 675

##

##           CP nsplit rel error xerror     xstd

## 1 0.0166667      0   1.00000 1.0000 0.047140

## 2 0.0111111      2   0.96667 1.0037 0.047169

## 3 0.0055556      3   0.95556 1.0074 0.047198
## 4 0.0000000      5   0.94444 1.0222 0.047307
```





The R-Square chart shows a difference in the relationship between Apparent and X Relative showing as the number of splits increases, the separation between the two also increases.

The X Relative Error decreases to the lowest point at one split and includes more error at increased split levels.



Due to the low frequency of words throughout all the movies, the max depth set at 5 helps cut down the decision tree for use in classifying the movies based on parental rating. As the max depth increases, the accuracy of the model decreases.

Based on the chart, using the word "evil" is the first word to group within the entire decision tree. With only 2 percent coming off the total 100 percent, this decision chart shows the difficulties of not having a wide variety of movies between all ratings and a low word frequency for each description.

The following confusion matrix indicates the difficulties of using classification as a way to identify parental rating.

```
##        true
## Rating    G  PG PG-13   R
##   G       4   6     0   2
##   PG      0   4     7   3

##   PG-13   3  11    19  26

##   R      25  97   126 332

## [1] "Correct Ratings: 0.53984962406015"

## [1] "Incorrect Ratings: 0.46015037593985"
```

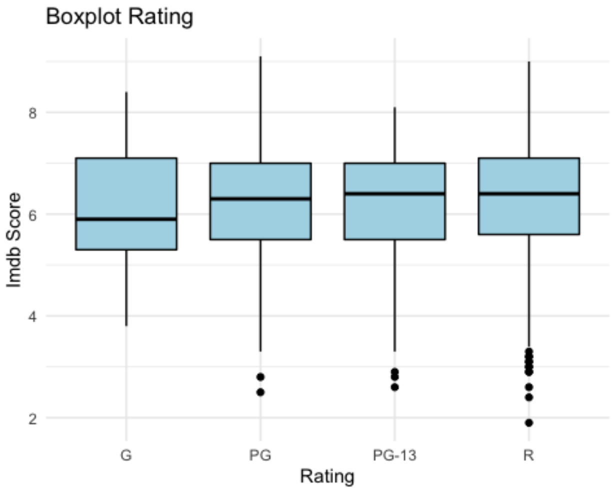
Exploratory Data Analysis on IMDb Scores

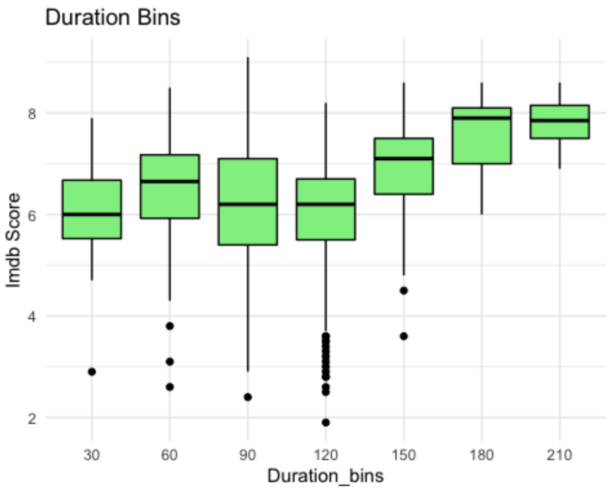
When building the model the target variable will be predicting IMDb score. Before jumping into the models, performing some exploratory analysis will help in identify different attributes and show the overall structure of the data.

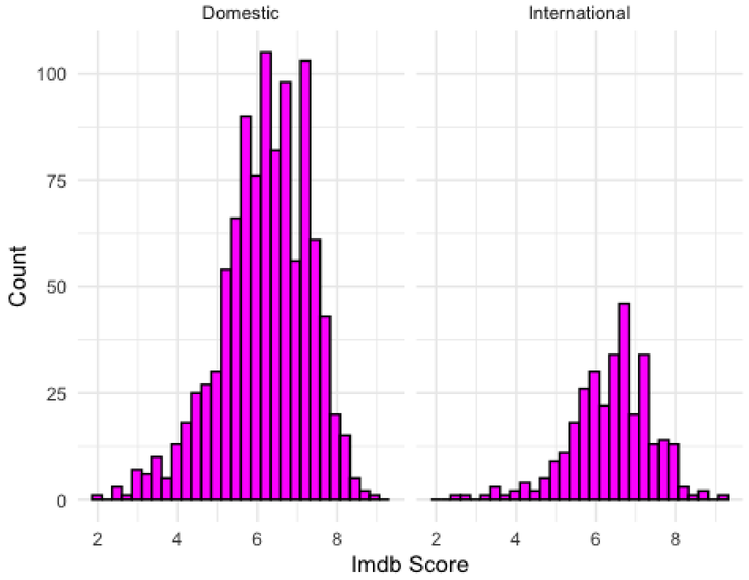
- Boxplot Rating
  - Breaking IMDb score by rating shows that PG, PG-13, and R all have around the same average and are distributed similarly.
  - The tails and outliers are different for PG, PG-13, and R, but not too different.
  - When looking at the Rating score for G, the average and median is different than the remaining cohort of data.
- Boxplot Duration Bins
  - When looking at the boxplot, the duration bin between 1.5 and 2.0 hours appear to have somewhat normally distributed data.
  - The remaining duration bins either become smaller and/or are not as evenly distributed.
- Histogram International

  - Breaking the data out as International or domestic, there is quite more data in the domestic category.
  - Both have a sizable amount of data and are left skewed.
  - Domestic here means that the film is exclusively released in the United States and no other country.

- Overall
  - When breaking data into multiple dimensions, the information can be less actionable due to not having enough data.
  - The data here, as different dimensions are applied, shows that in most cases the data is not unevenly distributed.
  - As a point of caution, adding additional dimensions into the model will decrease the predictive capabilities.



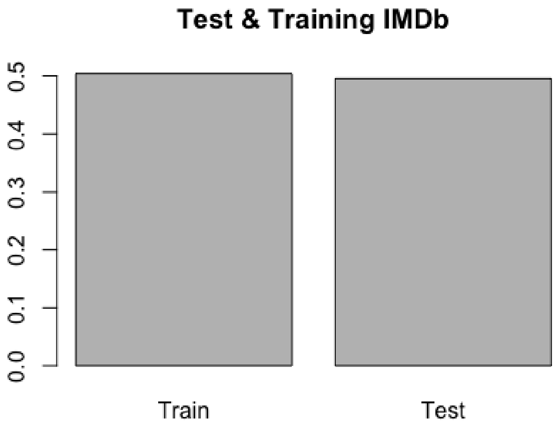




Creating Test/Train Data

The next step is to break the data into a Test and Train set with about 50% for both groups.

```
## Train  Test
##   676   664
```



Predicting IDMB Scores

Model Results 1

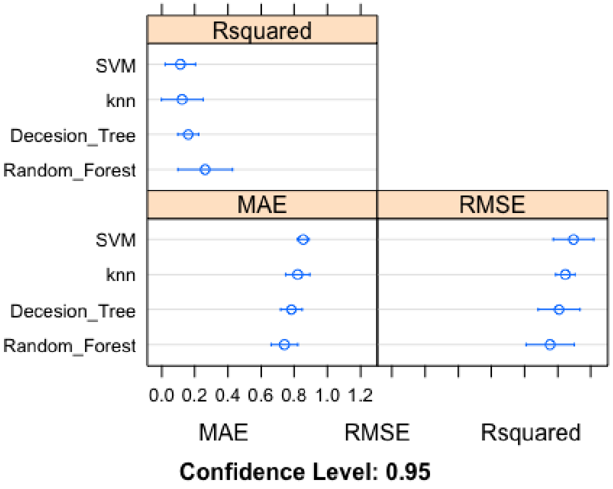
Looking at Rating, Duration Bins, International Flag, Genres and Director to help build each of the models

Interpreting the Results

- When using cross validation on the Train data set, Random Forest looks to be the best among SVM, knn, and Decesion Tree.
- The Rsquared is about 2.2 for Random Forest and the Root Mean Square Error is lower the rest of the other models.
- That being said, overall the Rsquared values are not at a favorable state to accept the model.
- Additionally, when trying to test the model, Director and Genre are too unique to be used and, therefore, the model errors out.

```
## 
## Call:
```

```
## summary.resamples(object = results)

##

## Models: Decesion_Tree, knn, SVM, Random_Forest

## Number of resamples: 3

##

## MAE

##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's

## Decesion_Tree 0.7612581 0.7690432 0.7768283 0.7832051 0.7941786 0.8115289    0

## knn           0.7896707 0.8079498 0.8262290 0.8212933 0.8371045 0.8479801    0

## SVM           0.8446052 0.8467244 0.8488435 0.8547796 0.8598668 0.8708901    0

## Random_Forest 0.7040233 0.7295214 0.7550194 0.7410392 0.7595471 0.7640747    0

##

## RMSE

##                    Min.   1st Qu.    Median      Mean   3rd Qu.     Max. NA's

## Decesion_Tree 0.9615736 0.9813088 1.0010440 1.0083673 1.0317641 1.062484    0

## knn           1.0198361 1.0369283 1.0540204 1.0463624 1.0596255 1.065231    0

## SVM           1.0478487 1.0702159 1.0925830 1.0954701 1.1192808 1.145979    0

## Random_Forest 0.8887343 0.9304109 0.9720875 0.9539377 0.9865393 1.000991    0

##

## Rsquared

##                     Min.    1st Qu.    Median      Mean   3rd Qu.      Max.

## Decesion_Tree 0.13672021 0.14757147 0.1584227 0.1606770 0.1726554 0.1868882

## knn           0.06947011 0.10061544 0.1317608 0.1236772 0.1507808 0.1698007
## SVM           0.07308266 0.09729015 0.1214976 0.1133102 0.1334239 0.1453502
## Random_Forest 0.20382041 0.22715921 0.2504980 0.2627454 0.2922080 0.3339179
##               NA's
## Decesion_Tree    0
## knn              0
## SVM              0
## Random_Forest    0
```



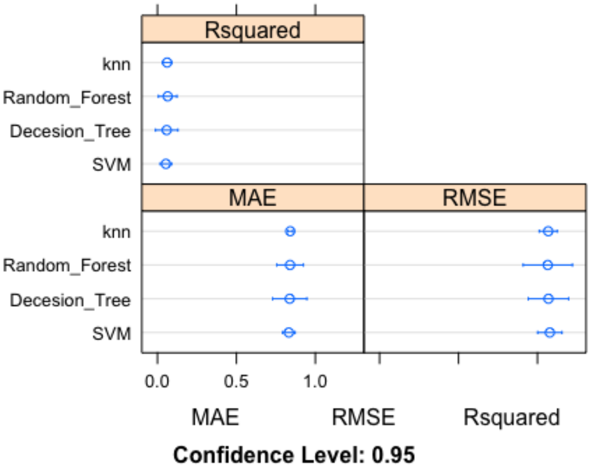
Model Results 2

Looking at Rating, Duration Bins, and International Flag to help build the models.

Interpreting the Results

- All of the models have an Rsquared just over 0.
- The RMSE’s are also still too high for any of the models to be considered successful.
- Overall, the model would be hard to accept in applying to real world solutions like predicting and improving the IMDb score based on these attributes.

```
##
## Call:
## summary.resamples(object = results)
##
## Models: Decesion_Tree, knn, SVM, Random_Forest
## Number of resamples: 3
##
## MAE
##                    Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## Decesion_Tree 0.8071586 0.8127176 0.8182765 0.8377500 0.8530456 0.8878147    0
## knn           0.8335939 0.8387641 0.8439344 0.8411651 0.8449507 0.8459670    0
## SVM           0.8157723 0.8245742 0.8333760 0.8322313 0.8404607 0.8475455    0
## Random_Forest 0.8016061 0.8267182 0.8518304 0.8399497 0.8591215 0.8664126    0
##
## RMSE
##                    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## Decesion_Tree 1.0234610 1.041895 1.060329 1.069676 1.092784 1.125239    0
## knn           1.0542846 1.055067 1.055850 1.068258 1.075244 1.094639    0
## SVM           1.0593274 1.060535 1.061743 1.078382 1.087909 1.114075    0
## Random_Forest 0.9971346 1.037642 1.078150 1.065359 1.099471 1.120792    0
##
## Rsquared
##                     Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## Decesion_Tree 0.02547729 0.04844762 0.07141795 0.05826309 0.07465600 0.07789405
## knn           0.05249723 0.05444881 0.05640039 0.06119403 0.06554243 0.07468447
## SVM           0.04221511 0.04546900 0.04872289 0.05355168 0.05921997 0.06971704
## Random_Forest 0.03728251 0.05761053 0.07793854 0.06457916 0.07822748 0.07851641
##               NA's
## Decesion_Tree    0
## knn              0
## SVM              0
## Random_Forest    0
```



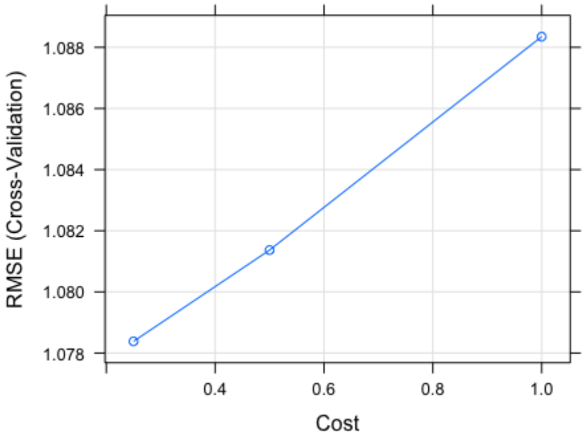
R Squared Results for Models

Reviewing SVM Model

- The computational cost goes up in order to identify the range of RMSE’s that could potentially occur by using this model.

- Said a different way, the cost to cross validate increases substantially when moving from the second cross validation to the third cross validation.

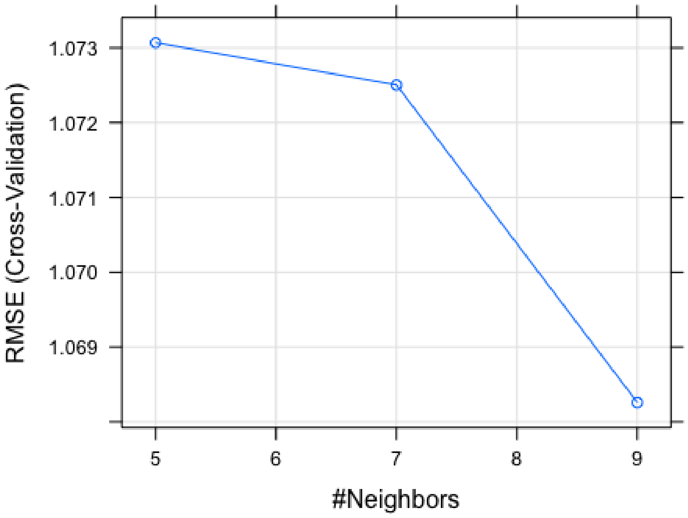
```
## Support Vector Machines with Radial Basis Function Kernel
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 452, 450, 450
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared    MAE
##   0.25  1.078382  0.05355168  0.8322313
##   0.50  1.081368  0.05031733  0.8314339
##   1.00  1.088349  0.04645726  0.8357987
##
## Tuning parameter 'sigma' was held constant at a value of 0.06688856
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were sigma = 0.06688856 and C = 0.25.
```



Reviewing KNN Model

- Looking at knn for cross validating 3 points, the RMSE goes down as we introduce more number of neighbors.

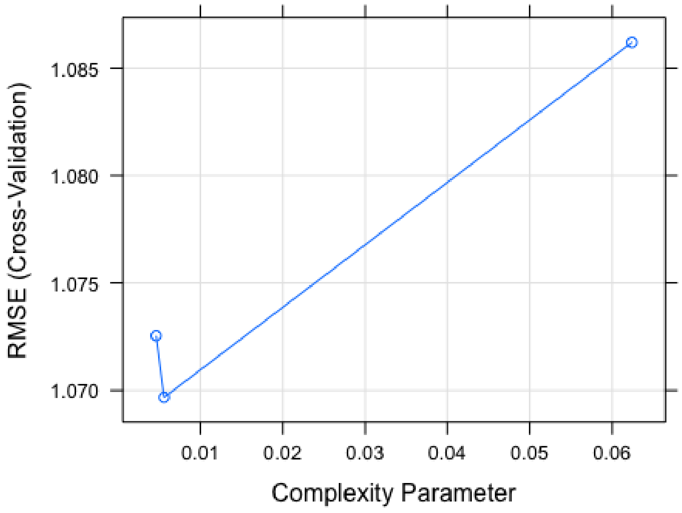
```
## k-Nearest Neighbors
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 451, 450, 451
## Resampling results across tuning parameters:
##
##   k  RMSE      Rsquared    MAE
##   5  1.073068  0.05694859  0.8441783
##   7  1.072502  0.05724067  0.8442520
##   9  1.068258  0.06119403  0.8411651
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 9.
```



Reviewing Decision Tree Model

- The complexity of the decision tree dramatically goes up during cross validation.
- From the first to the second point, the increase is only marginal.
- When going to the third cross validation point, the jump is more than 6 times.

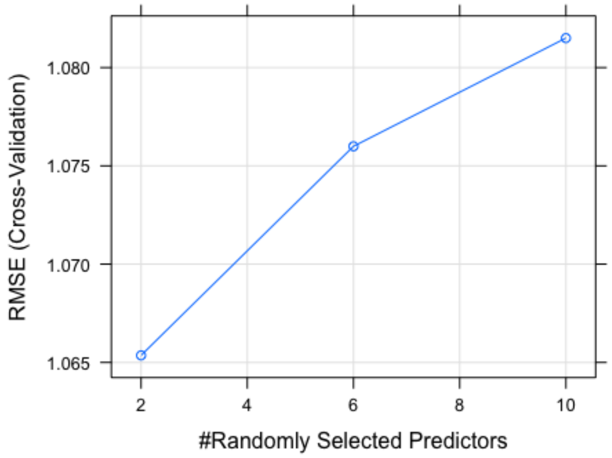
```
## CART
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 450, 451, 451
## Resampling results across tuning parameters:
##
##   cp           RMSE      Rsquared    MAE
##   0.004607214  1.072544  0.05716714  0.8373405
##   0.005582405  1.069676  0.05826309  0.8377500
##   0.062422683  1.086199  0.03998550  0.8614039
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.005582405.
```



Reviewing Random Forest

- This model takes the longest to run and, in order to capture a range of 3 RMSE’s points, 10 randomly selected predictors had to be introduced.

```
## Random Forest
##
## 676 samples
##   3 predictor
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 451, 450, 451
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared    MAE
##    2    1.065359  0.06457916  0.8399497
##    6    1.075996  0.05756937  0.8404841
##   10    1.081501  0.05397236  0.8421437
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 2.
```



Overall R-Squared and RMSE

Before jumping into predicting IMDb scores, the average Rsquares and RMSE's are as followes:

```
##     svm_rsq    knn_rsq   tree_rsq     rf_rsq
## 1 0.05355168 0.06119403 0.05826309 0.06457916

##   svm_rmse knn_rmse tree_rmse  rf_rmse
## 1 1.078382 1.068258  1.069676 1.065359
```

Pridicting IMDB Scores

- Something interesting happened here, regardless of how low all of the Rsquares were and how how the RMSE were, knn was able to achieve a RMSE of 0.07 much lower than what the cross validation chose.
- The hypothesis to why K Nearest Neighbor is able to achieve these results is because most of the data is around an IMDb score of 6. knn captures similarity by looking at the distance

or closeness to each data point. Meaning that the Rsquared is a great validation method to understand if the input variable can explain the change in the target variable (IMDb Score).

- Due to the data already being so close to 6, a user would be better off just guessing the score.

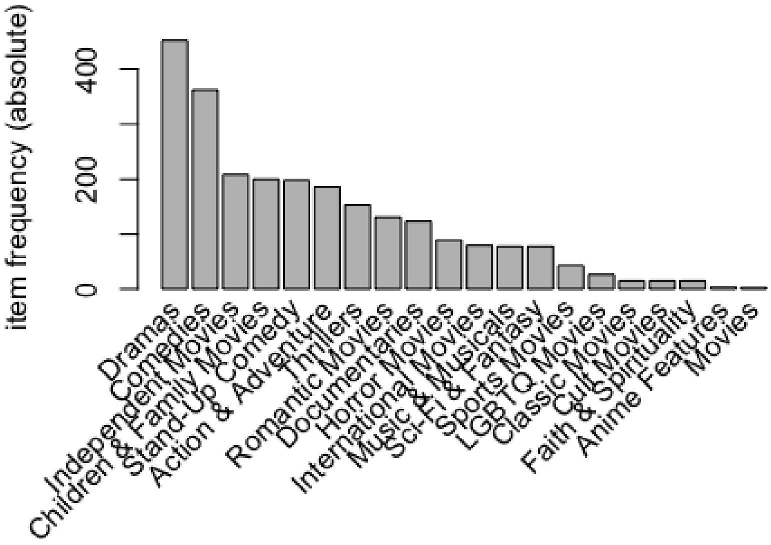
```
##   Imdb.Score     tree      svm      knn random_f
## 1        5.7 6.104967 6.389784 6.210769 6.178737
## 2        5.9 6.104967 6.009974 5.959016 6.037264
## 3        6.0 6.995833 6.859163 6.426531 7.179161
## 4        5.8 6.104967 6.009974 5.959016 6.037264
## 5        6.1 6.104967 6.009974 5.959016 6.037264
## 6        6.0 6.104967 6.389784 6.210769 6.178737

##   results svm_rmse   knn_rmse tree_rmse  rf_rmse
## 1   Train 1.078382 1.06825781  1.069676 1.065359
## 2    Test 1.046301 0.07241446  1.051801 1.044411
```

Association Rule Mining

Association rule mining required the data to be transaction type structure. Meaning that each movie will be treated as a separate transaction with, at most, 3 different genres associated to them. The goal for using this method is to identify the next movie to watch by simply looking to genre.

- The most watched type of movie is either Drama or Comedies.
- After those two genres, the movies step down dramatically from around 350 to 200 titles.

```
##      items
## [1]  {Comedies}
## [2]  {Dramas,Thrillers}
## [3]  {Horror Movies,Independent Movies,Thrillers}
## [4]  {Dramas,Faith & Spirituality,Romantic Movies}
## [5]  {Action & Adventure,Sci-Fi & Fantasy}
## [6]  {Dramas,Thrillers}
## [7]  {Action & Adventure}
## [8]  {Dramas,LGBTQ Movies}
## [9]  {Independent Movies,Sci-Fi & Fantasy,Thrillers}
## [10] {Comedies,Dramas,Independent Movies}
## [11] {Dramas,Independent Movies}
## [12] {Action & Adventure}
## [13] {Action & Adventure,Dramas}
## [14] {Dramas,Independent Movies,Romantic Movies}
## [15] {Documentaries}
## [16] {Children & Family Movies}

## [17] {Comedies,Romantic Movies}

## [18] {Children & Family Movies,Dramas,Romantic Movies}

## [19] {Children & Family Movies,Dramas,Romantic Movies}

## [20] {Children & Family Movies,Dramas,Romantic Movies}
```



The data is now in a transaction format and the next step includes looking at *support* and *confidence*: - *support* is an indication of how frequently an item appear in the data - *confidence* indicates the number of times the if-then statements are found true.

- Setting the minimum support to 0.001 pulls as many items into the dataset, but not all items to avoid bringing transactions that did not have as many associations.

- The minimum confidence is set to 0.8 in order to bring in rules that are significant in making actionable decisions.

Looking to some summary information about the rules illuminates some information about the data:

- The number of rules generated: 8.
- The distribution of rules by length: most rules are 3 items long.
- The summary of quality measures: interesting to see ranges of support, lift, and confidence.
- The information on the data mined: total data mined and minimum parameters.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[20 item(s), 1340 transaction(s)] done [0.00s].

## sorting and recoding items ... [20 item(s)] done [0.00s].

## creating transaction tree ... done [0.00s].

## checking subsets of size 1 2 3 done [0.00s].

## writing ... [8 rule(s)] done [0.00s].

## creating S4 object  ... done [0.00s].


## set of 8 rules

##

## rule length distribution (lhs + rhs):sizes

## 2 3

## 1 7

##

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.

##    2.00    3.00    3.00    2.88    3.00    3.00

##

## summary of quality measures:

##     support         confidence      coverage            lift          count

##  Min.   :0.0015   Min.   :0.80   Min.   :0.0015   Min.   : 2.4   Min.   : 2.0

##  1st Qu.:0.0015   1st Qu.:0.82   1st Qu.:0.0015   1st Qu.: 2.4   1st Qu.: 2.0

##  Median :0.0026   Median :0.93   Median :0.0030   Median : 2.8   Median : 3.5

##  Mean   :0.0039   Mean   :0.91   Mean   :0.0046   Mean   : 4.4   Mean   : 5.2

##  3rd Qu.:0.0050   3rd Qu.:1.00   3rd Qu.:0.0060   3rd Qu.: 4.4   3rd Qu.: 6.8

##  Max.   :0.0090   Max.   :1.00   Max.   :0.0112   Max.   :10.9   Max.   :12.0
```

```
##

## mining info:

##              data ntransactions support confidence

##  GenreTransactions          1340   0.001        0.8
```

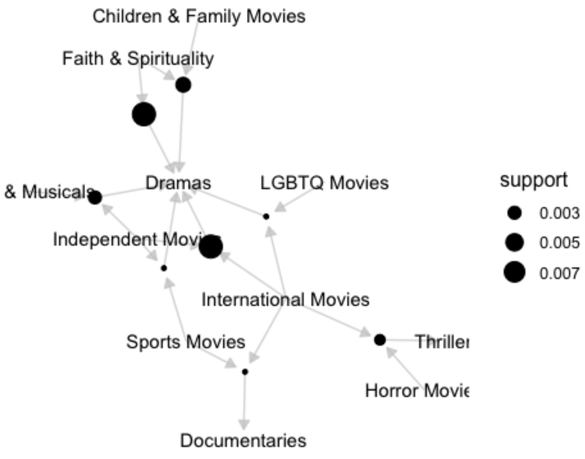
Exploring Metrics to Evaluate - Confidence

- The chart below shows a general sense of what the rules look like when sorting by confidence.
- The most confident predictors are when an individual watches an international or independent film that is either about Sports or LGBTQ has a confidence of 1 and in only one case 0.86. The movies suggested to watch next is a drama.
- In the case of the movie currently being watched with international and sports as the genre tags, the next movie recommended would be a documentary.
- If a person watches a horror movie and from an international production country, the next movie suggestion should be a thriller.

```
##     lhs                          rhs             support confidence coverage lift count

## [1] {International Movies,

##      LGBTQ Movies}            => {Dramas}         0.0015       1.00   0.0015  3.0     2

## [2] {International Movies,

##      Sports Movies}           => {Documentaries}  0.0015       1.00   0.0015 10.9     2

## [3] {Independent Movies,
##     Sports Movies}          => {Dramas}        0.0015      1.00   0.0015  3.0     2
## [4] {Horror Movies,
##     International Movies}    => {Thrillers}     0.0022      1.00   0.0022  8.8     3
## [5] {Independent Movies,
##     International Movies}    => {Dramas}        0.0090      0.86   0.0104  2.5    12
## [6] {Children & Family Movies,
##     Faith & Spirituality}    => {Dramas}        0.0037      0.83   0.0045  2.5     5
## [7] {Faith & Spirituality}    => {Dramas}        0.0090      0.80   0.0112  2.4    12
## [8] {Independent Movies,
##     Music & Musicals}        => {Dramas}        0.0030      0.80   0.0037  2.4     4
```

Plotting the Association Rule

Most movies lead back to Dramas, Documentaries, or Thrillers, with a high support for Drama.



Conclusion

In terms of accurately grouping movies as they arrive in the recommender system, clustering and classification both struggled due primarily to the low frequency of words in the description with an average of 14 words per movie. Classification did end up slightly more successful than clustering with about a 54 percent success rate but still did not perform well with movies that had lower parental ratings. Pulling a summary of the movie from other websites with more information about the film might make classification and clustering viable methods for initially grouping the data in the future. Additionally, getting more movies with lower ratings into the system could help the reliability of clustering and classification as movies with a R rating greatly outnumbered the other movies.

Overall, the models were not able to provide value in predicting IMDb score. More attributes might help in having the precision to to predict a score but simply using Genre, Rating, if the movie was international and Duration Bins, does not help explain the score. K-Nearest Neighbors appeared to have an incredibly low Root Mean Square error but is overshadowed by the cross validation reducing the ability to adopt the model.

This association rule mining technique proved to be valuable in helping identify the next movie to watch. Although Drama was the most watched genre, what was interesting is the models technique identified for international sport movies that a Netflix watcher should jump to a Documentary next. This is outside the norm of what is expected in just normal behavior. In addition, the lift was substantially higher than the other movies with confidence levels over 0.80. That suggestion could have a noticeable impact on helping Netflix watchers discover what they might like to watch next.

References

Makhija, S. (2021, July). Netflix Movies and TV Shows 2021. Kaggle. https://www.kaggle.com/satpreetmakhija/netflix-movies-and-tv-shows-2021

Course Reflection

The Applied Machine Learning course is the next step up in data science through the addition of various algorithms and data analysis techniques required for machine learning. Relying on R programming as the primary tool, the course highlights methods to evaluate data and understand the type of analysis necessary for the specific data. The final project for this class necessitated the use of multiple algorithms to find the best solution for creating a recommendation system and those same principles apply to many other real-world applications. In learning from the project, the following are important aspects to keep in mind when evaluating and understanding data:

- Ensure the data analysis has a logical flow from the start of evaluating and exploring a dataset to the end. Like many of the projects in this portfolio, having an idea in mind for the end solution elevates the exploration and analysis of data. This project started without an end goal in mind and ended up adding hours' worth of work for the team trying to capture the data story.
- Breaking down the problem into smaller steps is an essential strategy for tackling larger problems. Knowing the breakdown for each problem helps in focusing efforts on stepping blocks to achieve the desired goal.
- Along the same lines as the previous point, the intention for this project centered on a Netflix-like recommendation system but the data was not adequate to hit that lofty goal. Being able to pivot to a new goal after the exploration phase is a viable option and having an end result not meeting the desired end state is perfectly fine if there are follow-up actions the organization can take to improve the analysis.

This course fulfills the requirement for four out of seven of the portfolio learning objectives through finding the data story and using different machine learning techniques.

- **Describe a broad overview of the major practice areas in data science:**

  The Applied Machine Learning course demonstrates major practice areas in data science through the emphasis placed on telling the data story. Taking the results produced through machine learning and showing how they fall into the broader goals of the organization is an imperative skill for success in data science.

- **Collect and organize data:**

  The project specifically introduced challenges in collecting and organizing the data. Although the data came from Kaggle, the data needed extensive cleaning and processing in preparation for analysis. This issue remains a key aspect for Data Scientists as data quality is important for machine learning algorithms.

- **Identify patterns in data via visualization, statistical analysis, and data mining:**

  Exploration through data visualizations, analysis, and data mining was a critical aspect for this project and drove the understanding of the data. Using statistical analysis helped for pulling together the data story and was critical in reaching the conclusion.

- **Develop alternative strategies based on the data:**

  The exploration phase for this project shed same heavy light on what was and was not possible to do with the data. After dropping the clustering technique and switching more to a regression modeling aspect, the recommendation system improved drastically.

Portfolio Conclusion

The reflections contained within this portfolio report exemplify the significant areas of knowledge gained throughout the Applied Data Science program at Syracuse University. Each of the classes outlined above hold a piece of the axiomatic data science puzzle and represent the foundational blocks for communicating both the statistical processes involved in data analysis and the model development required to reach the predictive and prescriptive elements successful organizations need. Beyond the tools learned in this program, the lessons and strategies developed for achieving solutions derived from data analysis show in the careful development of projects integral for permanent success in the field.

The projects outlined above represent major practice areas in the field of Data Science and highlight the skills obtained in using widely available tools in conjunction with statistical principles. Collection and organization of data using the data warehouse and database principles drive the subsequent analysis and visualization of data through displaying the necessary strategies for handling organizational data. Data consisting of text, pictures, or any other data type and can form as structured, semi-structured, or unstructured and requires knowledge for effectively managing the flow in and out of enterprise systems. Without these steps, businesses would struggle in achieving functional decision-making and would be unable to take advantage of key insights into business processes.

The theory behind the methods within data science augment the application of data collection, exploration, visualization, analysis, and modeling for enterprise-level operations and is essential to successful decision-making. Each party involved in organizational processes represents an important part of achieving long-term success and communication is integral in serving as the connection point between teams within an organization. From a data science aspect, effectively communicating with database managers, business intelligence analysts, decision-makers, and any other professionals within an organization elevates the management and analysis to create a solid culture of data with a foundation of trust. This trust is critical as organizations have a duty to protect the privacy of individuals while ensuring the ethical principles behind data analysis remain intact.

Portfolio References

Applied Data Science Master's Degree. (2022). Syracuse University. Retrieved February 12, 2022, from **https://ischool.syr.edu/academics/applied-data-science-masters-degree**

Block, Gregory. (2021, June 30). *Applied Machine Learning for Data Analytics.* Syracuse University, IST 707.

Harper, Chad. (2021, January 4). *Database Administration Concepts and Database Management.* Syracuse University, IST 659.

Khan, Humayun. (2022, January 3). *Data Warehouse.* Syracuse University, IST 722.

Larche, Michael. (2021, September 29). *Natural Language Processing.* Syracuse University, IST 664.

Stinnett, John. (2022, January 3). *Project Portfolio Milestone Requirement.* Syracuse University, IST 782.

The Team Data Science Process lifecycle. (2022, February 11). Microsoft. Retrieved February 12, 2022, from **https://docs.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle**