Game of Thrones : Text Analysis of the George R.R Martin's book series *A Song of Ice and Fire* using SAS Text Miner

Brad Gross and Srividhya Naraharirao, Louisiana State University

LOUISIANA STATE UNIVERSITY

Background

A Song of Ice and Fire

- 60 million copies sold
- 45 languages
- HBO television show: 18.6 million viewers
- ***Presentation may contain spoilers**

Objective

Books contain a unique narrative structure of switching narrator point of view on a per chapter basis along with having dozens of characters, families, and locations.

Can we determine **speaker traits based upon text clusters and factor analysis**, **character qualities** based on common words used, **relationship strength** based on interactions?

Can we use text analytics with multiple models to attempt to predict which family's point of view the reader is viewing the world from?

Tools

SAS Enterprise Miner

Filter, Data Partition, Metadata, Regression, and Save Data

Used for filtering observations, data control, predictive modeling, and building data sets

SAS Text Miner

Text Import, Text Parsing, Text Filter, Text Profile, Text Cluster, Text Topic

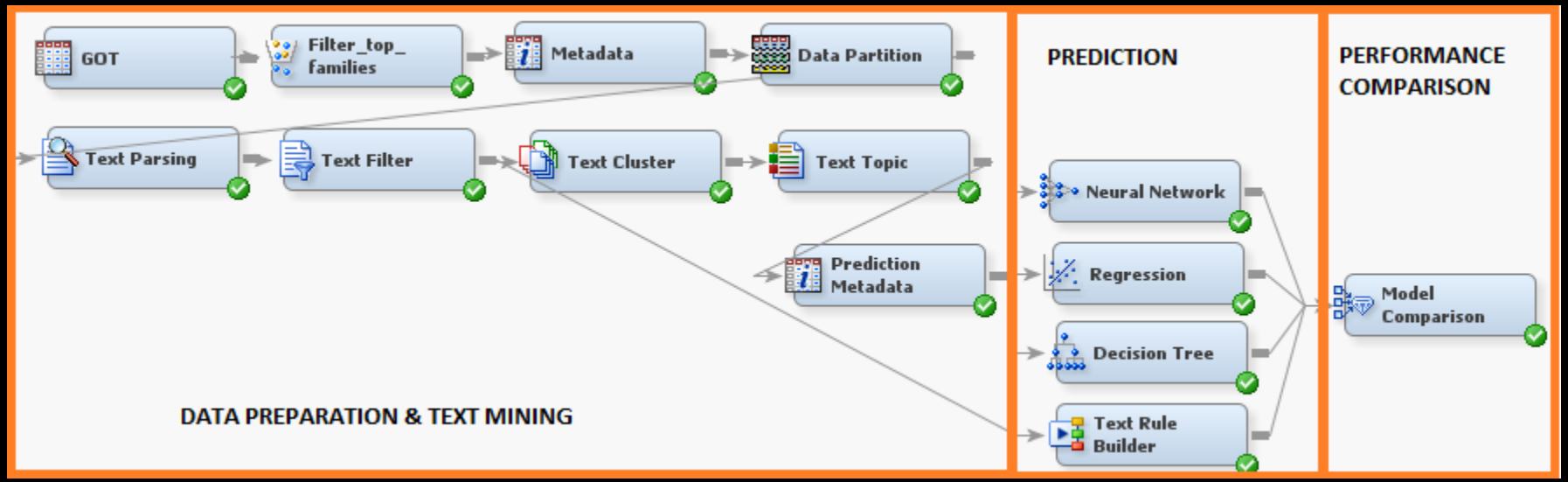
Used for reading in text, filtering terms, including/excluding parts of speech, pattern discovery

Corpus

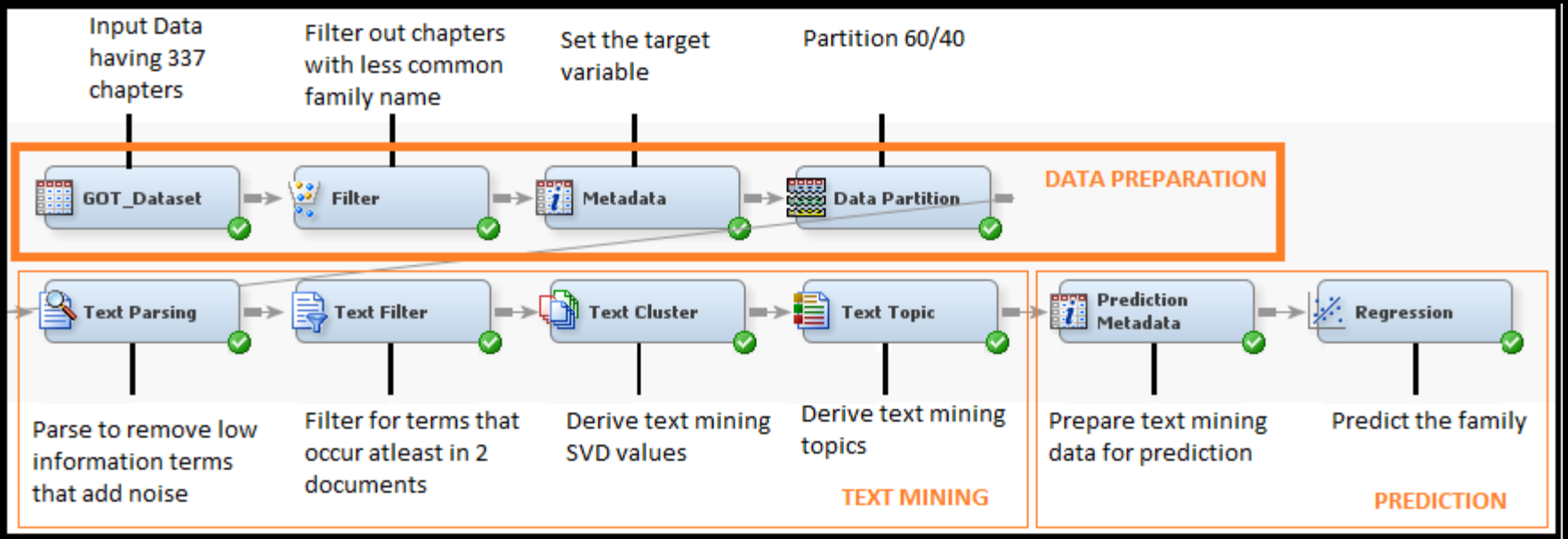
5 Books
337 Chapters
24 Character Perspectives
13 Families
134,840,898 characters of text

	TEXT	FIRST_NAME	LAST_NAME	BOOK	CHAPTER
1	THE PROPHET The prophet was drowning men on Gr...	aeron	greyjoy	4	1
2	THE DROWNED MAN Only when his arms and legs w...	aeron	greyjoy	4	2
3	THE CAPTAIN OF GUARDS The blood oranges are w...	areo	hotah	4	1
4	THE WATCHER Let us look upon this head," his princ...	areo	hotah	5	1
5	THE QUEENMAKER Beneath the burning sun of Dom...	arianne	martell	4	1
6	THE PRINCESS IN THE TOWER Hers was a gentle ...	arianne	martell	4	2
7	ARYA Arya's stitches were crooked again. She frowne...	arya	stark	1	1
8	ARYA Her father had been fighting with the council ag...	arya	stark	1	2
9	ARYA The one-eared black tom arched his back and ...	arya	stark	1	3
10	ARYA "High," Syrio Forel called out, slashing at her he...	arya	stark	1	4

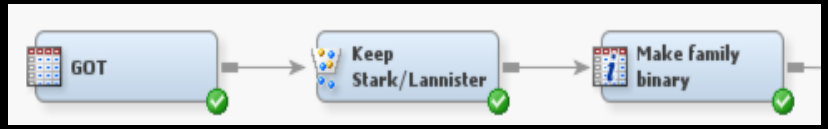
Process Flow

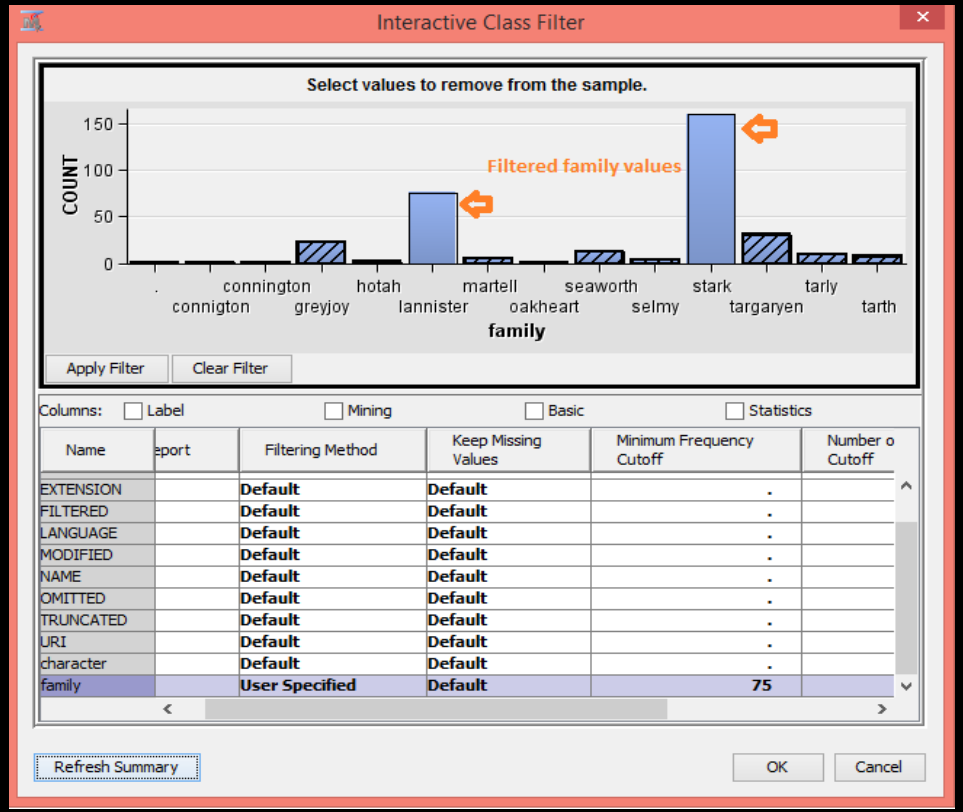


Step 1: Filter / Metadata / Partitioning



Filter





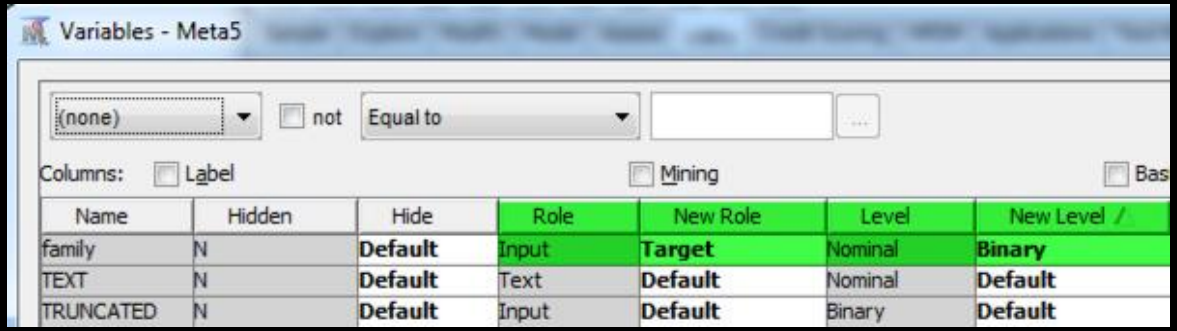
Bar graphs of all imported variables

User specified selection of which values to keep

Removal of missing values

Minimum Frequency/Number of Levels cutoffs

Metadata



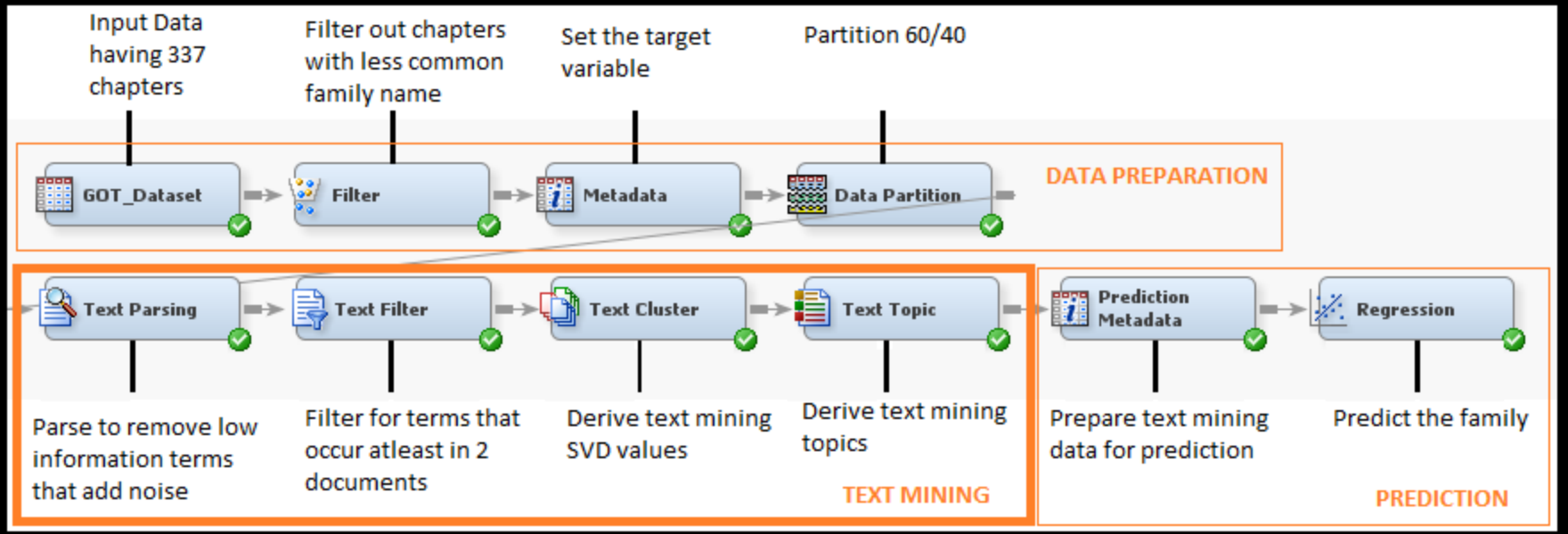
Configure and change metadata

Data Partition

Split data into training and validation sets

Train	
Variables	
Output Type	Data
Partitioning Method	Default
Random Seed	12345
Data Set Allocations	
Training	60.0
Validation	40.0
Test	0.0

Step 2: Text Mining



Text Parsing

First step in text mining analysis. Text parsing uses advanced natural language processing to represent documents as collections of terms

- Word stemming
- Exclude parts of speech
- Determine and exclude entity types
- Specified a stop list and multi-word list of terms that needs to be ignored from analysis including 18 castles, 491 people, 22 places, 24 words considered too specific, and 317 multi-word terms.

OUTPUT:

20,000 terms to be considered for further analysis

General	
Node ID	TextParsing4
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Parse	
Parse Variable	TEXT
Language	English
Detect	
Different Parts of Speech	No
Noun Groups	No
Multi-word Terms	
Find Entities	Standard
Custom Entities	
Ignore	
Ignore Parts of Speech	'Aux' 'Conj' 'Det' 'Interj' 'Parl
Ignore Types of Entities	'Location' 'Person' 'Prop_mis
Ignore Types of Attributes	'Num' 'Punct'
Synonyms	
Stem Terms	Yes
Synonyms	SASHELP.ENGSYNMS
Filter	
Start List	
Stop List	SASHELP.ENGSTOP
Select Languages	

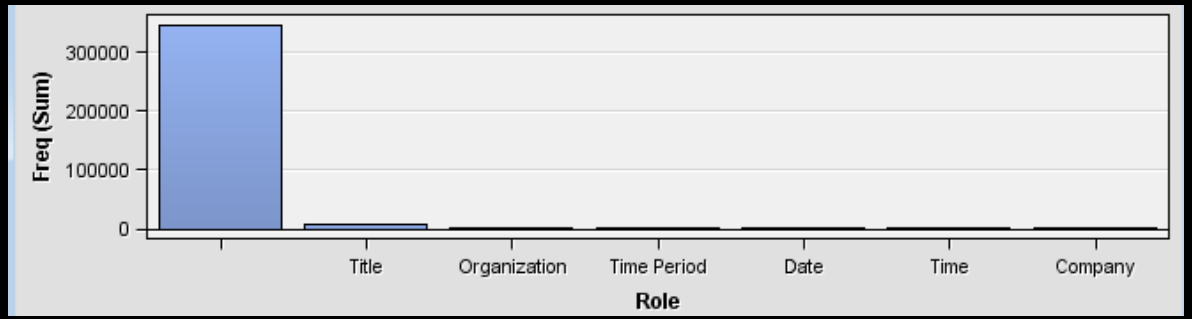
Text Parsing - Results

Corpus is parsed into terms:

- Role
- Attribute
  Alpha – All letters
- Frequency
- # of Docs
- Term will be kept

Term	Role ▼	Attribute	Freq	# Docs	Keep
lord ...	Title	Entity	3045	138	Y
king ...	Title	Entity	559	110	Y
lady ...	Title	Entity	520	104	Y
queen ...	Title	Entity	656	103	Y
sister ...	Title	Entity	403	100	Y
guard ...	Title	Entity	121	67	Y

Term ▲	Role	Attribute	Freq	# Docs	Keep
tyrell ...		Alpha	12	8	N
tyrells ...		Alpha	2	2	Y
tyrells ...	Organization	Entity	1	1	Y
tyrion ...		Alpha	86	28	N
tyrion lanniste...		Alpha	11	7	N



Text Filtering

Filter out terms that appeared in only one document.

Weight assigned to terms based on 'Inverse Document Frequency'

Terms occurring infrequently are given a higher score

.. Property	Value
General	
Node ID	TextFilter
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Spelling	
Check Spelling	No
Dictionary	
Weightings	
Frequency Weighting	Default
Term Weight	Inverse Document Frequency
Term Filters	
Minimum Number of Documents	2
Maximum Number of Terms	.
Import Synonyms	
Document Filters	
Search Expression	
Subset Documents	
Results	
Filter Viewer	
Spell-Checking Results	
Exported Synonyms	
Report	
Terms to View	All
Number of Terms to Display	20000
Status	

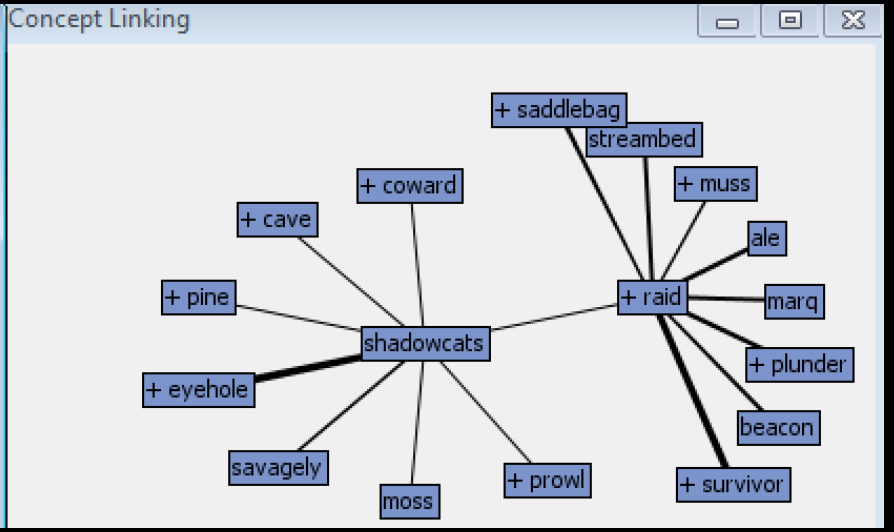
Text Filtering - Results

Terms are given a weight based on the inverse of the their frequency used .

Concept linking is a way to find and display the terms that are highly associated with the selected term in the Terms table. The selected term is surrounded by the terms that correlate the strongest with it.

Terms

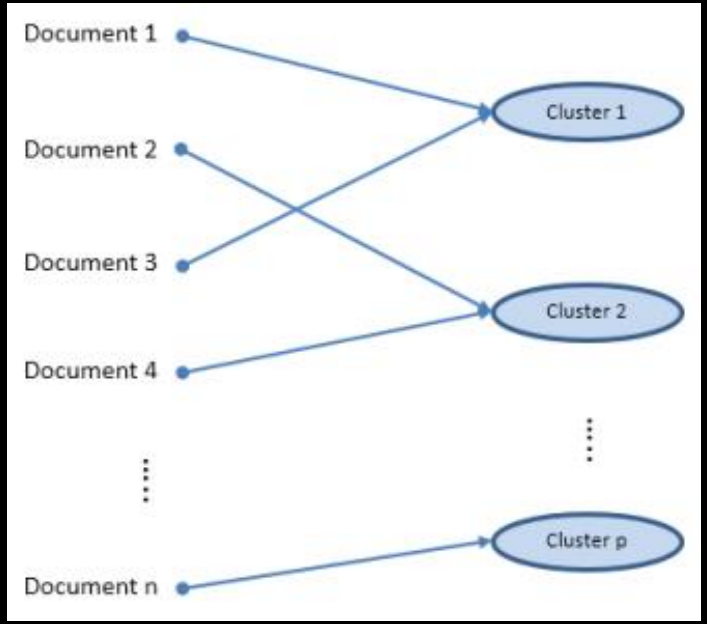
TERM	FREQ	# DOCS	KEEP	WEIGHT ▼
quilting	2	2	☑	7.14
eastward	2	2	☑	7.14
downwind	2	2	☑	7.14
shadowcats	2	2	☑	7.14
gunthor	2	2	☑	7.14
⊞ lizard	2	2	☑	7.14
trader	3	2	☑	7.14
sailor	3	2	☑	7.14
hearthfire	2	2	☑	7.14
⊞ half-crouch	2	2	☑	7.14



Text Clustering

First step in the knowledge extraction process. The following steps extract patterns from the data and match observations to the patterns.

Text Cluster node will discover themes and assign each document to one of these themes.



Text Clustering

Two clustering algorithms are available

- The **Expectation Maximization** algorithm clusters documents with a flat representation,
- The **Hierarchical clustering** algorithm groups clusters into a tree hierarchy

- Both approaches rely on the singular value decomposition (SVD) to transform the original weighted, term-document frequency matrix into a dense but low dimensional representation

General	
Node ID	TextCluster
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Transform	
SVD Resolution	Low
Max SVD Dimensions	10
Cluster	
Exact or Maximum Number	Exact
Number of Clusters	5
Cluster Algorithm	Expectation-Maximization
Descriptive Terms	15

Singular Value Decomposition

Dimensionality reduction technique

SVD resolution - Higher the number higher the risk of fitting to noise

Original Matrix

	document	error	invalid	message	file	format	unable	to	open	using	path	variable
1	d1	1	1	1	1	1	0	0	0	0	0	0
2	d2	1	0	2	1	0	1	1	1	1	1	0
3	d3	1	0	0	0	1	1	1	0	0	0	1

Form Dot Products

Weights

	U_2	
error	43	.30
invalid	.11	.13
message	.55	-.37
file	.33	-.12
format	.21	.55
unable	.31	.18
to	.31	.18
open	.22	-.25
using	.22	-.25
path	.22	-.25
variable	.09	.42

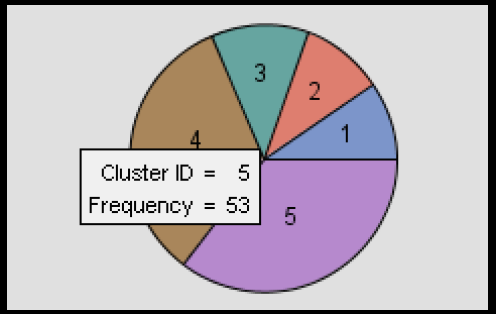
	document	SVD1	SVD2
1	d1	1.63	.49
2	d2	3.14	-.96
3	d3	1.35	1.64

Normalize

	document	SVD1	SVD2
1	d1	.96	.29
2	d2	.96	-.29
3	d3	.64	.77

Matrix for Mining

Text Clustering - Results



Cluster ID	Descriptive Terms
1	blackfish tully freys grace +hope +castle +king +knight king +ser +child +blood +hair +death +die
2	grace maester +death +friend +iron +dragon +castle +king +fight king +hope +hundred lord +few +child
3	khaleesi +dragon grace +gold +bear +horse +begin +death +land +ser +high +child three +blood +grow
4	grace king +gold +ser +king +dragon +smile +mean +father +knight +close +fear +land +friend +feel
5	three maester lord +foot +hundred +run +high +pass +castle +horse freys +great +black first +grow

Text Clusters Representations

Cluster 1

Cluster 3

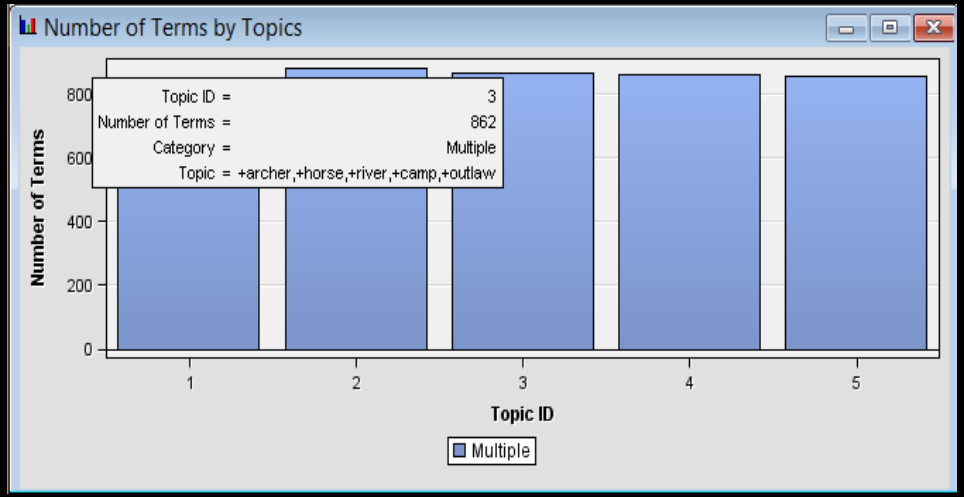
Cluster 2

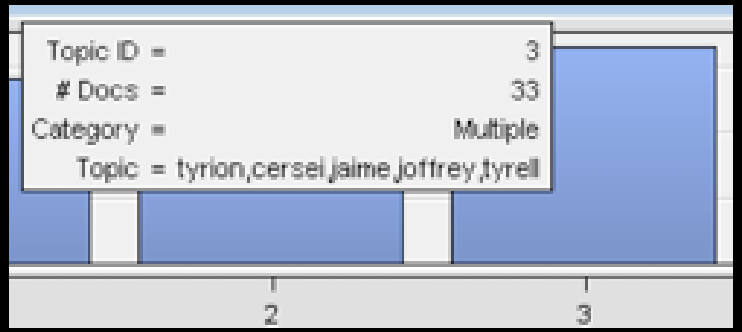
But all that went straight out of his head when he entered the Hand's solar to find Cersei, Ser Kevan, and Grand Maester Pycelle gathered about Lord Tywin and the king. Joffrey was almost bouncing, and Cersei was savoring a smug little smile, though Lord Tywin looked as grim as ever. I wonder if he could smile even if he wanted to. "What's happened?" Tyrion asked.

Text Topic

Text Topic node will derives themes/concepts from the terms that can be used instead of the terms

Each document is assigned to zero or more of those themes





We see topics clearly separating people/places:

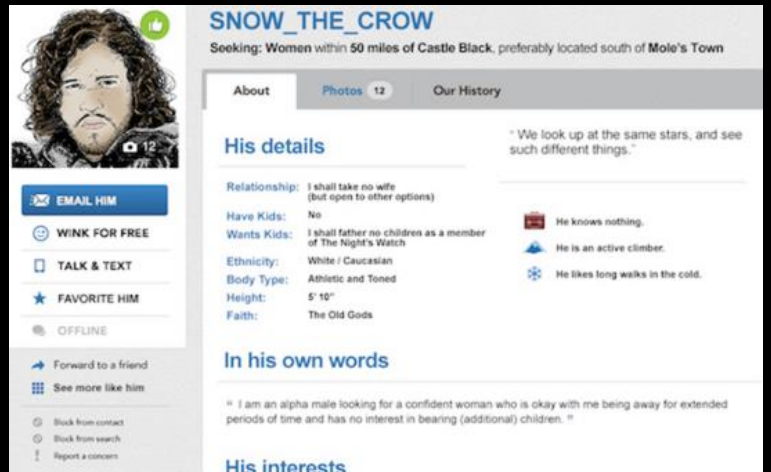
- Topic 1 - North
- Topic 2 - Cities
- Topic 3 - Rural areas
- Topic 5 - East

Document Cutoff	Term Cutoff	Topic	Number of Terms	# Docs
0.167	0.017	+wildling,+raven,+ranger,dolorous,ranger	827	21
0.245	0.017	+council,+ser,+wed,lord,+king	881	31
0.174	0.017	+archer,+horse,+river,+camp,+outlaw	862	24
0.333	0.017	n't,+man,+tree,+look,+back	859	22
0.163	0.017	+slave,cyvasse,+deck,+elephant,+dwarf	855	12

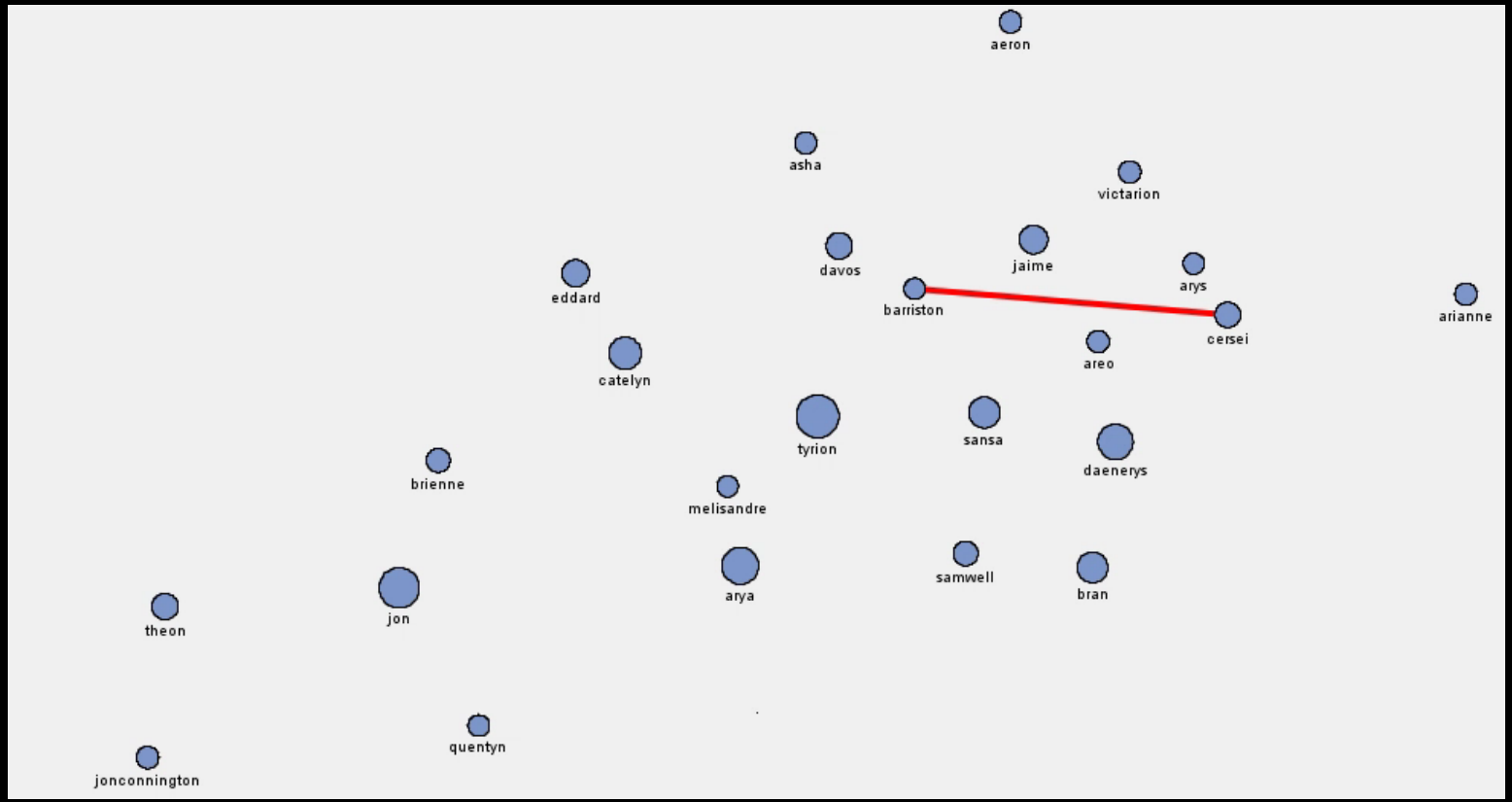
Text Profiling

The text profile node enables us to profile a target variable based on a set of terms from the document

Text profiling was leveraged to profile the Game of thrones characters, identify similarities and relationship strengths between the characters based on their interactions



Text Profiling results – Character relationship strengths



Text Profiling results – terms describing each character

Name ▲	Value	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6	Term 7	Term 8
character	aeron	sea	drown	priest/Ttl	wave	god	shout	hill	captain
character	areo						r/Ttl	skull	pool
character	arianne							know	sand
character	arya	n't	look	blind	tent	stone	run	wolf	road
character	arys	father	love	knight	prince/Ttl	woman	find	white	leave
character	asha	justin/Ttl	march	god	die	king	storm	day	horse
character	barriston	queen/Ttl	pit	pyramid	beast	brazen	city	king	dragon
character	bran	wolf	dream	n't	cr				
character	brienne	septon	dog	road	ra				
character	catelyn	lady	son	room	rain	north	lord	answer	south
character	cersei	queen/Ttl	hand	little	wed	lady/Ttl	ser	seven	wine
character	daenerys	dragon	blood	brother	slave	child	ride	wed	city
character	davos	ship	water						son
character	eddard	king	rain						seat
character	jaime	wench	sword						cousin
character	jon	ice	giant	wildling	watch	horn	raven	ranger	arrow
character	jonconnington	land	company	house	castle	golden	year	exile	camp
character	melisandre	wildling	fire	flame	bone	skull	boy	black	eye
character	quentyn	fr						run	three
character	samwell	g						fire	brother
character	sansa	n						knight	down
character	theon	uncle	castle	wall	dog	gate	warm	arrow	hall
character	tyrion	river	sister/Ttl	stone	slave	master	gold	sweet	pay
character	victarion	ship	iron	fleet	sea	woman	captain/Ttl	sail	hand

drowned men waded out to seize the wretch and hold him underwater. "Lord God who drowned for us," the priest prayed, in a voice as deep as the sea, "let Emmond your servant be reborn from

Deeper inside the pyramid, another four Brazen Beasts had been set to guard the iron doors outside the pit where

Jon had no wish to linger here. He started walking toward the wildling a dead giant whose head had been crushed by a stone. A raven was

Melisandre paid the naked steel no mind. If the wildling had meant her harm, she would have seen it in her flames.

Text Rule Builder

This node provides a text mining predictive modeling solution within SAS Text Miner

This derives a set of classification rules from the terms which are useful in describing and predicting the target variable

For eg. (Term A) &(Term B) ~(Term C) can be a rule to classify a target variable

The results of the model are highly interpretable

```
if yourFavoriteCharacter then
    death = prettySoon;
```

Text Rule Builder results

Rule	Rule #	Target Value	Precision	Recall	True Positive/Total	True positive/Total(Target level)
golden & wine & faith	1	LANNISTER	100.0%	35.56%	16/16	16/45
wit & sail	2	LANNISTER	100.0%	62.22%	12/12	28/45
jape & dwarf	3	LANNISTER	97.37%	82.22%	9/10	37/45
debt & enjoy	4	LANNISTER	97.67%	93.33%	5/5	42/45
grey	5	STARK	100.0%	83.33%	80/80	80/96
tree	6	STARK	100.0%	93.75%	10/10	90/96

The golden hand was the occasion for much admiring
knocked over a goblet of wine. Then his temper got the

was grey, and I could not see a foot past the nose
the trees were like long skinny arms reaching out

Statistics Label	Train	Validation
Average Squared Error	0.031691	0.034619
Divisor for ASE	282	190
Maximum Absolute Error	0.51476	0.51476
Sum of Frequencies	141	95
Root Average Squared Error	0.178021	0.186061
Sum of Squared Errors	8.936952	6.577572
Frequency of Classified Cases	141	95
Misclassification Rate	0.028369	0.231579
Number of Wrong Classifications	4	22

Training Hitrate: 97.2%

Validation Hitrate:76.85%

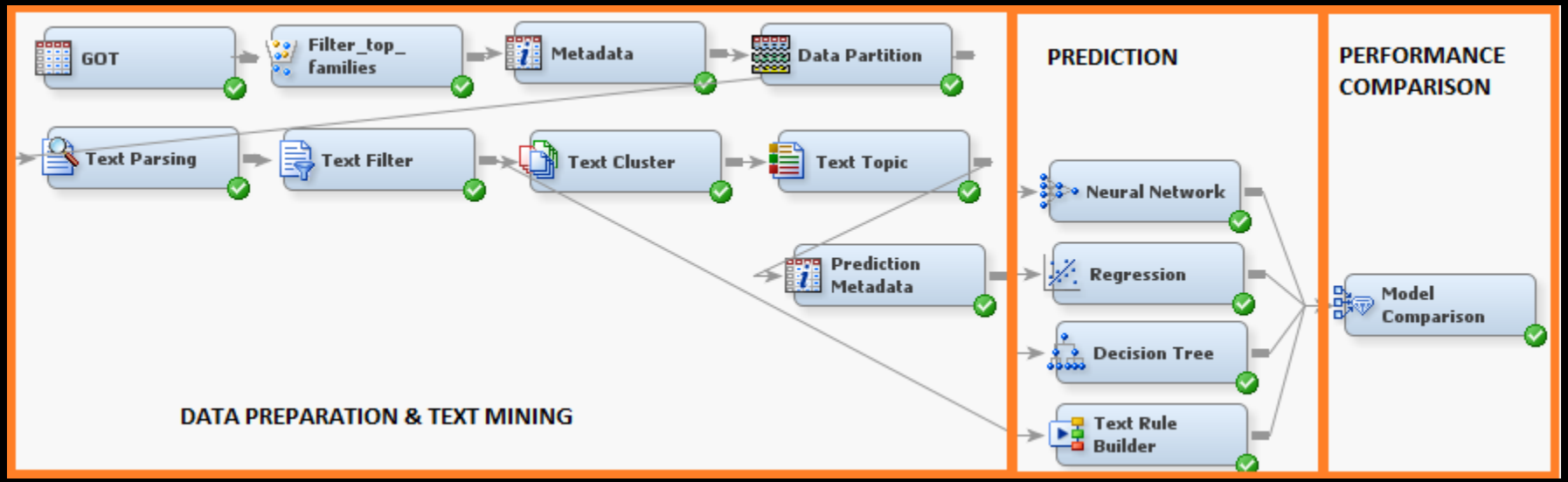
Not a great model for prediction, good for explanation

Comparison of predictive models

Once the predictor variables were obtained from the text mining process, various models were tested for their accuracy in predicting the character family by chapter

- Logistic Regression
- Decision Tree
- Neural Networks
- Text Rule Builder

Comparison of predictive models – process flow

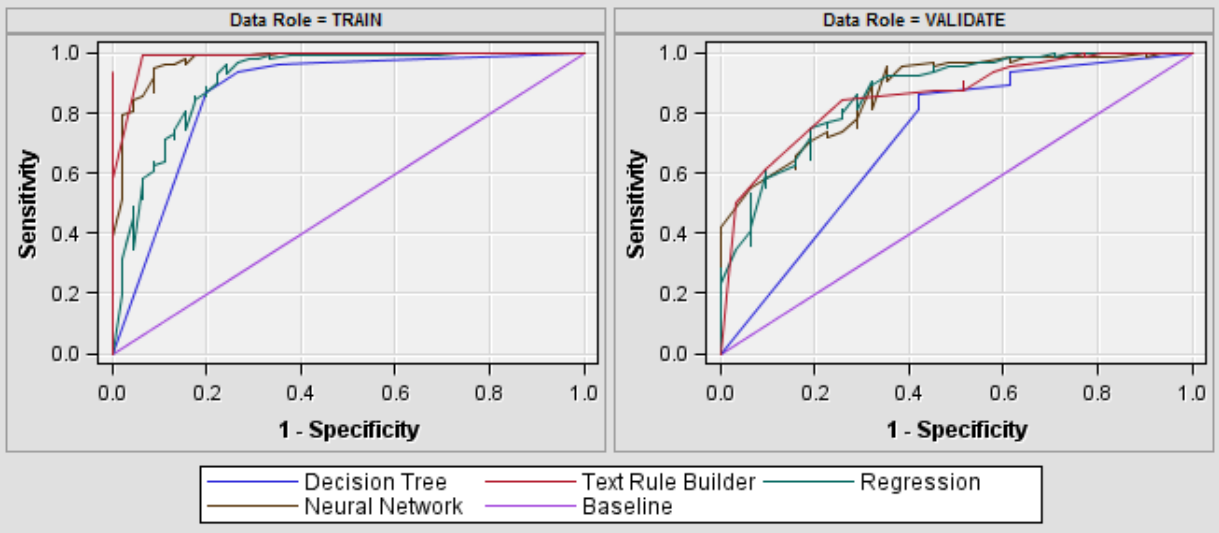


Model Comparison results

Model Description	Target Variable	Target Label	Selection Criterion: Train: Misclassification Rate	Valid: Misclassification Rate
Text Rule Builder	family		0.028369	0.231579
Neural Network	family		0.06383	0.168421
Regression	family		0.113475	0.189474
Decision Tree	family		0.12766	0.231579

Naive Rule:

96 Stark Chapters / 141 Total Chapters = 68%



Neural Nets vs Logistic Regression – Family Prediction

All predictors show as extremely significant One predictor shows as significant

Likelihood Ratio Test for Global Null Hypothesis: BETA=0

-2 Log Likelihood Intercept Only	-2 Log Likelihood Intercept & Covariates	Likelihood Ratio Chi-Square	DF	Pr > ChiSq
1052.614	88.972	963.6423	161	<.0001

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
TextCluster2_SVD1	5	2290.5706	<.0001
TextCluster2_SVD2	15	119.9281	<.0001
TextTopic_raw1	8	952.3889	<.0001
TextTopic_raw2	8	1497.9255	<.0001
TextTopic_raw3	16	28967.8810	<.0001
TextTopic_raw4	11	1625.6245	<.0001
TextTopic_raw5	11	34.8861	0.0003

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Standardized Estimate	Exp(Est)
Intercept	1	-3.7213	7.9386	0.22	0.6392		0.024
TextCluster_SVD1	1	7.1816	8.0853	0.79	0.3744	0.1867	999.000
TextCluster_SVD2	1	3.8720	8.2265	0.22	0.6379	0.6140	48.040
TextTopic2_raw1	1	-1.4273	8.3568	0.03	0.8644	-0.0657	0.240
TextTopic2_raw2	1	-7.6357	18.7575	0.17	0.6840	-0.3455	0.000
TextTopic2_raw3	1	-12.1276	8.5365	2.02	0.1554	-0.5266	0.000
TextTopic2_raw4	1	7.9615	10.5620	0.57	0.4510	0.2698	999.000
TextTopic2_raw5	1	-16.6764	7.7213	4.66	0.0308	-0.7322	0.000

Odds Ratio Estimates

Effect	Point Estimate
TextCluster_SVD1	999.000
TextCluster_SVD2	48.040
TextTopic2_raw1	0.240
TextTopic2_raw2	<0.001
TextTopic2_raw3	<0.001
TextTopic2_raw4	999.000
TextTopic2_raw5	<0.001

Neural Nets – Character Prediction

Predicting chapter by character

All text allowed

Prediction tends to get confused mainly amongst characters who interact often (Arya and Brienne) and characters who appear less often (Aeron)

Classification Table

Data Role=TRAIN Target Variable=character Target Label=' '

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
ARYA	ARYA	86.364	95.000	19	9.9476
BRIENNE	ARYA	9.091	50.000	2	1.0471
SANSA	ARYA	4.545	7.692	1	0.5236
ASHA	BRAN	11.111	50.000	1	0.5236
BRAN	BRAN	77.778	58.333	7	3.6649
THEON	BRAN	11.111	14.286	1	0.5236
ARYA	BRIENNE	100.000	5.000	1	0.5236
AERON	CATELYN	4.000	50.000	1	0.5236