

# **A Model for Age and Gender Profiling of Social Media Accounts Based on Post Contents Documentation**

*Release*

**Cheng, Fernandez, Quindoza, Tan**

August 15, 2017



<b>1</b>	<b>thesis</b>	<b>1</b>
1.1	Driver50 module . . . . .	1
1.2	addEngPOS module . . . . .	2
1.3	batchprocessing module . . . . .	2
1.4	combinepos module . . . . .	2
1.5	features package . . . . .	3
1.6	model package . . . . .	8
1.7	pipelinewraps package . . . . .	10
1.8	prepareedstheis module. . . . .	14
1.9	utility package . . . . .	15
	<b>Python Module Index</b>	<b>17</b>
	<b>Index</b>	<b>19</b>



## 1.1 Driver50 module

`Driver50.clean ( x )`

**Parameters** **x** -- data to be cleaned

**Returns** cleaned email and links from the data

`Driver50.dimensionReduction ( X, y, source, mindf, maxdf, data=None )`

perform dimension reduction

**Parameters**

- **X** -- text data
- **y** -- classes (gender and age)
- **source** -- twitter, facebook, or merged
- **mindf** -- lower threshold for term frequency filter
- **maxdf** -- upper threshold for term frequency filter
- **data** -- features of the data

`Driver50.evaluate ( age_data, gen_data, both_data, model )`

Evaluates the age and gender profiling performance of the model (various model structures)

**Parameters**

- **age\_data** -- data feature selected based on age
- **gen\_data** -- data feature selected based on gender
- **both\_data** -- data feature selected based on age and gender
- **model** -- classifier to be used

**Returns**

`Driver50.execute ( )`

entire process to execute. Includes feature extraction, dimension reduction, and evaluation

`Driver50.getSpecificFeatures ( data, features )`

filters the features

**Parameters**

- **data** -- features of the data

- **features** -- specific features to be retrieved

**Returns** specified features

`Driver50.get_Data_from_CSV ( source, mindf, maxdf, fs, param=None )`

- Parameters**
- **source** -- twitter, facebook, or merged
  - **mindf** -- lower threshold for term frequency filter
  - **maxdf** -- upper threshold for term frequency filter
  - **fs** -- feature selection method
  - **param** -- parameter used by the feature selection

**Returns** age, gender, and both (combined structure) data

`Driver50.writeToExcel ( book, sheet, classifier, features, row )`

## 1.2 addEngPOS module

**class** `addEngPOS.ConnectionFactory`

Bases: `object`

**getConnectionThesis** ( )

`addEngPOS.add_english_pos ( )`

adds the english pos to the d :return:

## 1.3 batchprocessing module

`batchprocessing.getPosts ( )`

`batchprocessing.getPostsFromFile ( filepath )`

`batchprocessing.updateEngPOS ( ids, texts )`

`batchprocessing.updatePosts ( ids, posts )`

`batchprocessing.writePostsToFile ( posts, filepath )`

## 1.4 combinepos module

`combinepos.combinePOS ( )`

populate the texts' combined POS

## 1.5 features package

### 1.5.1 Submodules

#### 1.5.2 features.CharacterFeatures module

**class** features.CharacterFeatures.CharacterFeatures

Bases: object

Returns the character features of a text

**getNumberOfRepeatedPunctuationMarks** ( *text* )

Parameters **text** -- text to be processed

Returns total number of instances of consecutive punctuation marks

**getNumberOfRepetitiveAlphaCharacters** ( *text* )

Parameters **text** -- text to be processed

Returns total number of instances that alpha characters are repeated more than twice consecutively

**getNumberOfSpecialChars** ( *text* )

Parameters **text** -- text to be processed

Returns total number of special characters besides punctuation marks

**getNumberOfWhiteSpaces** ( *text* )

Parameters **text** -- text to be processed

Returns total number of white spaces

**getTotalNumberOfCharacters** ( *text* )

Parameters **text** -- text to be processed

Returns total number of characters

**getTotalNumberOfDigitalNumbers** ( *text* )

Parameters **text** -- text to be processed

Returns total number of digital numbers

**getTotalNumberOfLetters** ( *text* )

Parameters **text** -- text to be processed

Returns total number of letters

**getTotalNumberOfUppercase** ( *text* )

Parameters **text** -- text to be processed

Returns total number of uppercase letters

### 1.5.3 features.Context module

```
class features.Context.Context
    Bases: object
    Returns the contextual features (words after 'my') in a text

    process ( s )
        Parameters s -- text to be processed
        Returns    text containing the contextual features
```

### 1.5.4 features.EmojisEmoticons module

```
class features.EmojisEmoticons.EmojisEmoticons
    Bases: object

    getEmojiTFIDF ( data )

    getLabels ( )
```

### 1.5.5 features.Feature module

```
class features.Feature.Feature ( X, y, source, data=None )
    Bases: object
    Applies dimension reduction to data

    applyExtraction ( selection )
        applies feature selection
        Parameters
            • selection -- feature extraction technique
            • type -- Gender, Age, or Both

        Returns    feature extracted data

    applySelection ( selection, type )
        applies feature selection
        Parameters
            • selection -- feature selection technique
            • type -- Gender, Age, or Both

        Returns    feature selected data

    getFeatures ( selection, mode )
        applies feature selection or extraction
        Parameters
            • selection -- feature selection or extraction technique
            • mode -- Gender, Age, or Both

        Returns    feature selected or extracted data

    useLasso ( mode )
        applies LASSO feature selection
```



**Parameters** **selection** -- feature selection or extraction technique

**Returns** feature selected data

### 1.5.6 features.FeatureExtract module

**class** features.FeatureExtract.**FeatureExtract** ( *source, mindf, maxdf* )

Bases: object

Extracts features from the text and post time

**clean** ( *x* )

cleans the data

**Parameters** **x** -- text data

**Returns** cleaned text

**fit\_transform** ( *X* )

**Parameters** **X** -- text data

**Returns** dataframe containing features extracted

**get\_liwc** ( )

reads the LIWC csv files

**Returns** dataframe containing the LIWC results

**transform** ( *X* )

The transform is only done after fitting the data, useful for TFIDF features

**Parameters** **X** -- text data

**Returns** dataframe containing features extracted

### 1.5.7 features.FunctionWordCount module

**class** features.FunctionWordCount.**FunctionWordCount**

Bases: object

**FUNCTIONWORDS\_FILENAME** = 'features/functionwords.txt'

**getAdpositionCount** ( *text* )

**getAllFunctionWordCount** ( *text* )

**getArticleCount** ( *text* )

**getAuxillaryCount** ( *text* )

**getConjunctionCount** ( *text* )

**getInterjectionCount** ( *text* )

**getProSentenceCount** ( *text* )

**getPronounCount** ( *text* )

### 1.5.8 features.Links module

```
class features.Links.Links
    Bases: object

    get_keywords ( link )

    get_links ( text )

    get_list_keywords ( text )

    get_title ( link )
```

### 1.5.9 features.POSFeature module

```
class features.POSFeature.POSFeature
    Bases: object

    ADJECTIVE = 'JJ'

    UNKNOWN = 'UNK'

    VERB = 'VB'

    getCombinedPOSTag ( post )

    getEnglishPOS ( text )
        jvmPath = jpye.getDefaultJVMPath() jpye.startJVM(jvmPath, "-Djava.class.path=dependencies/
        NormAPI.jar;dependencies/RBPOST.jar") rbpost = JPackage("rbpost").RBPOST
        result = rbpost.tokenizer_Text(text) tokenizedText = result.split(" ") jpye.shutdownJVM()

    getPOSCount ( text )

    populateMappingDictionary ( )
```

### 1.5.10 features.POSSequencePattern module

```
class features.POSSequencePattern.POSSequencePattern ( documentList )
    Bases: object

    MAX_LENGTH = 7

    candidateGen ( fList )

    computeFairSCP ( key, count )

    minePOSPatterns ( minsup, minadherence )

    retrievePOSTags_docFrequency ( )
```

### 1.5.11 features.Structure module

```
class features.Structure.Structure
    Bases: object
```

```
ABBREVIATIONS_FILENAME = '../features/abbreviations.txt'

getAvgNCharacterPerParagraph ( text )

getAvgNSentencePerParagraph ( text )

getAvgNWordPerParagraph ( text )

getAvgNWordPerSentence ( text )

getNParagraphs ( text )

getNSentenceBegLower ( text )

getNSentenceBegUpper ( text )

getNSentences ( text )

getParagraphs ( text )
```

### 1.5.12 features.TFIDF module

```
class features.TFIDF.TFIDF ( mindf, maxdf )
    Bases: object
    Processes the TFIDF of text

    getFeatureNames ( )
        Returns labels of the features

    get_testing_TFIDF ( test )
        Parameters documentList -- testing text data
        Returns      tfidf of the text

    get_training_TFIDF ( documentList )
        Parameters documentList -- training text data
        Returns      tfidf of the text
```

### 1.5.13 features.WordCount module

```
class features.WordCount.WordCount
    Bases: object

    ABBREVIATIONS_FILENAME = 'features/abbreviations.txt'

    getAveLengthWords ( text )

    getDictOfWordsMappedToOccurrence ( text )

    getEntropy ( text )

    getHapaxDislegomena ( text )

    getHapaxLegomena ( text )
```

```
getHonoresR ( text )
getLolHmmCount ( text )
getNDifferentWords ( text )
getNWordsBegCapital ( text )
getNWordsWithRepLetters ( text )
getOccurrenceArray ( text )
getRatioOfHapaxDislegomena ( text )
getRatioOfHapaxLegomena ( text )
getRatioOfNetAbbrev ( text )
getRatioOfShortWords ( text )
getRatioOfUniqueWords ( text )
getSichelsS ( text )
getSimpsonsD ( text )
getTotalNumberOfWords ( text )
getWordLengthFreqDist ( text )
getYulesK ( text )
```

#### 1.5.14 Module contents

### 1.6 model package

#### 1.6.1 Submodules

##### 1.6.2 model.Document module

```
class model.Document.Document ( content, posSequence )
    Bases: object
```

##### 1.6.3 model.Post module

```
class model.Post.Post ( id, content, epos, fpos )
    Bases: object
```

##### 1.6.4 model.RootModel module

```
class model.RootModel.RootModel ( data, type, modelType, k=10 )
    Bases: object
```

This class represents the parallel and combined structure. Its results can be fed to the StackModel for the stacked model structure.

**evaluateKfold** ( *train\_predictions=None, test\_predictions=None* )

**Parameters**

- **train\_predictions** -- predictions of the model for the training data
- **test\_predictions** -- predictions of the model for the testing data

**Returns** returns the metrics for both training data and testing data

**getPredictions** ( )

**Returns** the predictions of the model for training and testing data

**getTestingUser** ( *ind* )

**Parameters** *ind* -- k-fold index

**Returns** users for the testing data for the ith k-fold

**getTestingX** ( *ind* )

**Parameters** *ind* -- k-fold index

**Returns** testing data for the ith k-fold

**getTestingy** ( *ind* )

**Parameters** *ind* -- k-fold index

**Returns** testing results for the ith k-fold

**getTrainingUser** ( *ind* )

**Parameters** *ind* -- k-fold index

**Returns** users for the training data for the ith k-fold

**getTrainingX** ( *ind* )

**Parameters** *ind* -- k-fold index

**Returns** training data for the ith k-fold

**getTrainingy** ( *ind* )

**Parameters** *ind* -- k-fold index

**Returns** training results for the ith k-fold

### 1.6.5 model.StackModel module

**class** model.StackModel.**StackModel** ( *root, modelType, data, type, k=10* )

Bases: object

This class represents the stacked structure.

**evaluateKfold** ( *train\_predictions=None, test\_predictions=None* )

**Parameters**

- **train\_predictions** -- predictions of the model for the training data
- **test\_predictions** -- predictions of the model for the testing data

**Returns** returns the metrics for both training data and testing data

**getPredictions ( )**

Returns the predictions of the model for training and testing data

**getTestingUser ( *ind* )**

Parameters **ind** -- k-fold index

Returns users for the testing data for the ith k-fold

**getTestingX ( *ind* )**

Parameters **ind** -- k-fold index

Returns testing data for the ith k-fold

**getTestingy ( *ind* )**

Parameters **ind** -- k-fold index

Returns testing results for the ith k-fold

**getTrainingUser ( *ind* )**

Parameters **ind** -- k-fold index

Returns users for the training data for the ith k-fold

**getTrainingX ( *ind* )**

Parameters **ind** -- k-fold index

Returns training data for the ith k-fold

**getTrainingy ( *ind* )**

Parameters **ind** -- k-fold index

Returns training results for the ith k-fold

## 1.6.6 Module contents

# 1.7 pipelinewraps package

## 1.7.1 Submodules

## 1.7.2 pipelinewraps.AgeRangeWrap module

**class** pipelinewraps.AgeRangeWrap.**AgeRangeWrap**

Bases: sklearn.base.TransformerMixin

Transforms the age to numerical labels TransformerMixin gives it the standard fit and transform functions to transform the data

**fit** ( *X*, *y=None*, **\*\*fit\_params** )

**transform** ( *X*, **\*\*transform\_params** )

```
pipelinewraps.AgeRangeWrap.enrange ( x )
```

**Parameters** *x* -- age of the user

**Returns** age range group

```
pipelinewraps.AgeRangeWrap.getClasses ( )
```

**Returns** array of the age ranges

### 1.7.3 pipelinewraps.CharacterWrap module

```
class pipelinewraps.CharacterWrap.CharacterWrap
```

Bases: sklearn.base.TransformerMixin

Processes all character features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

```
fit ( X, y=None, **fit_params )
```

```
transform ( X, y=None, **transform_params )
```

### 1.7.4 pipelinewraps.ContextualWrap module

```
class pipelinewraps.ContextualWrap.ContextualWrap ( target=None )
```

Bases: sklearn.base.TransformerMixin

Processes all contextual features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

```
fit ( X, *args, **kwargs )
```

```
transform ( X, y=None, **transform_params )
```

### 1.7.5 pipelinewraps.EmojiWrap module

```
class pipelinewraps.EmojiWrap.EmojiWrap ( target=None )
```

Bases: sklearn.base.TransformerMixin

Processes all emoji features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

```
fit ( X, *args, **kwargs )
```

```
transform ( X, y=None, **transform_params )
```

### 1.7.6 pipelinewraps.ExtractionWrap module

```
class pipelinewraps.ExtractionWrap.ExtractionWrap ( extraction, target=None )
```

Bases: sklearn.base.TransformerMixin

Performs feature extraction

```
fit ( X, *args, **kwargs )
```

```
transform ( X, y=None, **transform_params )
```

### 1.7.7 pipelinewraps.FunctionWrap module

**class** pipelinewraps.FunctionWrap.**FunctionWrap**

Bases: sklearn.base.TransformerMixin

Processes all function word features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

**fit** ( *X*, *y=None*, **\*\*fit\_params** )

**transform** ( *X*, *y=None*, **\*\*transform\_params** )

### 1.7.8 pipelinewraps.GenderWrap module

**class** pipelinewraps.GenderWrap.**GenderWrap**

Bases: sklearn.base.TransformerMixin

Transforms the gender to numerical labels TransformerMixin gives it the standard fit and transform functions to transform the data

**fit** ( *X*, *y=None*, **\*\*fit\_params** )

**transform** ( *X*, **\*\*transform\_params** )

pipelinewraps.GenderWrap.**enrange** ( *x* )

**Parameters** *x* -- gender of the user

**Returns** 0 for F and 1 for M

pipelinewraps.GenderWrap.**getClasses** ( )

**Returns** returns the gender classes

### 1.7.9 pipelinewraps.ItemSelector module

**class** pipelinewraps.ItemSelector.**ItemSelector** ( *key* )

Bases: sklearn.base.BaseEstimator, sklearn.base.TransformerMixin

**fit** ( *x*, *y=None* )

**transform** ( *data\_dict* )

### 1.7.10 pipelinewraps.LinkWrap module

**class** pipelinewraps.LinkWrap.**LinkWrap** ( *target=None* )

Bases: sklearn.base.TransformerMixin

Processes all link features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

**fit** ( *X*, *\*args*, **\*\*kwargs** )

**transform** ( *X*, *y=None*, **\*\*transform\_params** )



### 1.7.11 pipelinewraps.POSSeqWrap module

```
class pipelinewraps.POSSeqWrap.POSSeqWrap
    Bases: sklearn.base.TransformerMixin
    Processes all POS features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

    fit ( X, y=None, **fit_params )

    transform ( X, y=None, **transform_params )

pipelinewraps.POSSeqWrap.dfToDocument ( df )
```

### 1.7.12 pipelinewraps.PostTimeWrap module

```
class pipelinewraps.PostTimeWrap.PostTimeWrap
    Bases: sklearn.base.TransformerMixin
    Processes all word features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

    fit ( X, y=None, **fit_params )

    transform ( X, **transform_params )

pipelinewraps.PostTimeWrap.enrange ( x )
    Parameters x -- exact hour posted
    Returns    time group

pipelinewraps.PostTimeWrap.getClasses ( )
    Returns returns the post time classes
```

### 1.7.13 pipelinewraps.SelectionWrap module

```
class pipelinewraps.SelectionWrap.SelectionWrap ( selection, target=None )
    Bases: sklearn.base.TransformerMixin
    Performs feature selection

    fit ( X, y, *args, **kwargs )

    transform ( X, y=None, **transform_params )
```

### 1.7.14 pipelinewraps.StackAgeRangeWrap module

```
class pipelinewraps.StackAgeRangeWrap.StackAgeRangeWrap
    Bases: sklearn.base.TransformerMixin
    Transforms the age multiclass to multilabel binary TransformerMixin gives it the standard fit and transform functions to transform the data

    fit ( X, y=None, **fit_params )

    transform ( X, **transform_params )
```

```
pipelinewraps.StackAgeRangeWrap.getClasses ( )
```

Returns array of the age ranges

### 1.7.15 pipelinewraps.StackGenderWrap module

```
class pipelinewraps.StackGenderWrap.StackGenderWrap
```

Bases: sklearn.base.TransformerMixin

Transforms the gender multiclass to multilabel binary TransformerMixin gives it the standard fit and transform functions to transform the data

```
fit ( X, y=None, **fit_params )
```

```
transform ( X, **transform_params )
```

```
pipelinewraps.StackGenderWrap.getClasses ( )
```

Returns returns the gender classes

### 1.7.16 pipelinewraps.StructureWrap module

```
class pipelinewraps.StructureWrap.StructureWrap
```

Bases: sklearn.base.TransformerMixin

Processes all structure features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

```
fit ( X, y=None, **fit_params )
```

```
transform ( X, y=None, **transform_params )
```

### 1.7.17 pipelinewraps.WordWrap module

```
class pipelinewraps.WordWrap.WordWrap
```

Bases: sklearn.base.TransformerMixin

Processes all word features of the data. TransformerMixin gives it the standard fit and transform functions to transform the data

```
fit ( X, y=None, **fit_params )
```

```
transform ( X, y=None, **transform_params )
```

### 1.7.18 Module contents

## 1.8 preparedstthesis module

```
class preparedstthesis.ConnectionFactory
```

Bases: object

```
getConnectionThesis ( )
```

```
preparedstthesis.addposts ( )
```

```
preparedsthesis.addusers ( limit=None )
```

## 1.9 utility package

### 1.9.1 Submodules

#### 1.9.2 utility.DataCleaner module

```
class utility.DataCleaner.DataCleaner
    Bases: object

    URL = 'URL'

    USERNAME = 'USERNAME'

    clean_data ( post_content )

    clean_email ( post_content )
```

#### 1.9.3 utility.LanguageDetector module

```
class utility.LanguageDetector.Language
    Bases: object

    ENGLISH = 0

    FILIPINO = 1

    TAGLISH = 2

    UNKNOWN = -1

    getLanguage ( code )

class utility.LanguageDetector.LanguageDetector
    Bases: object

    englishOrTagalog ( string )

    getLanguage ( text )

    getLanguageDetailed ( text )
```

#### 1.9.4 utility.PostCleaner module

```
class utility.PostCleaner.PostCleaner
    Bases: object

    changeEmojisToText ( postContent )

    changeForeignToText ( postContent )

    changeLinkToText ( postContent )
```

**fixAcronymSpaces** ( *postContent* )

**getEmojis** ( *postContent* )

**insertSpace** ( *postContent* )

**normalizeUnicode** ( *postContent* )

**removeEmojis** ( *postContent* )

### 1.9.5 Module contents

- [\*Index\*](#)
- [\*Module Index\*](#)
- [\*Search Page\*](#)

## a

addEngPOS, 2

## b

batchprocessing, 2

## c

combinepos, 2

## d

Driver50, 1

## f

features, 8

- features.CharacterFeatures, 3
- features.Context, 4
- features.EmojisEmoticons, 4
- features.Feature, 4
- features.FeatureExtract, 5
- features.FunctionWordCount, 5
- features.Links, 6
- features.POSFeature, 6
- features.POSSequencePattern, 6
- features.Structure, 6
- features.TFIDF, 7
- features.WordCount, 7

## m

model, 10

- model.Document, 8
- model.Post, 8
- model.RootModel, 8
- model.StackModel, 9

## p

pipelinewraps, 14

- pipelinewraps.AgeRangeWrap, 10
- pipelinewraps.CharacterWrap, 11
- pipelinewraps.ContextualWrap, 11
- pipelinewraps.EmojiWrap, 11
- pipelinewraps.ExtractionWrap, 11
- pipelinewraps.FunctionWrap, 12
- pipelinewraps.GenderWrap, 12
- pipelinewraps.ItemSelector, 12
- pipelinewraps.LinkWrap, 12
- pipelinewraps.POSSeqWrap, 13
- pipelinewraps.PostTimeWrap, 13
- pipelinewraps.SelectionWrap, 13
- pipelinewraps.StackAgeRangeWrap, 13
- pipelinewraps.StackGenderWrap, 14
- pipelinewraps.StructureWrap, 14
- pipelinewraps.WordWrap, 14

prepareedsthesi, 14

## u

utility, 16

- utility.DataCleaner, 15
- utility.LanguageDetector, 15
- utility.PostCleaner, 15



**A**

ABBREVIATIONS\_FILENAME (features.Structure.Structure attribute), 7  
ABBREVIATIONS\_FILENAME (features.WordCount.WordCount attribute), 7  
add\_english\_pos() (in module addEngPOS), 2  
addEngPOS (module), 2  
addposts() (in module preparedsthesi), 14  
addusers() (in module preparedsthesi), 15  
ADJECTIVE (features.POSFeature.POSFeature attribute), 6  
AgeRangeWrap (class in pipelinewraps.AgeRangeWrap), 10  
applyExtraction() (features.Feature.Feature method), 4  
applySelection() (features.Feature.Feature method), 4

**B**

batchprocessing (module), 2

**C**

candidateGen() (features.POSSequencePattern.POSSequencePattern method), 6  
changeEmojisToText() (utility.PostCleaner.PostCleaner method), 15  
changeForeignToText() (utility.PostCleaner.PostCleaner method), 15  
changeLinkToText() (utility.PostCleaner.PostCleaner method), 15  
CharacterFeatures (class in features.CharacterFeatures), 3  
CharacterWrap (class in pipelinewraps.CharacterWrap), 11  
clean() (features.FeatureExtract.FeatureExtract method), 5  
clean() (in module Driver50), 1

clean\_data() (utility.DataCleaner.DataCleaner method), 15  
clean\_email() (utility.DataCleaner.DataCleaner method), 15  
combinepos (module), 2  
combinePOS() (in module combinepos), 2  
computeFairSCP() (features.POSSequencePattern.POSSequencePattern method), 6  
ConnectionFactory (class in addEngPOS), 2  
ConnectionFactory (class in preparedsthesi), 14  
Context (class in features.Context), 4  
ContextualWrap (class in pipelinewraps.ContextualWrap), 11

**D**

DataCleaner (class in utility.DataCleaner), 15  
dfToDocument() (in module pipelinewraps.POSSeqWrap), 13  
dimensionReduction() (in module Driver50), 1  
Document (class in model.Document), 8  
Driver50 (module), 1

**E**

EmojisEmoticons (class in features.EmojisEmoticons), 4  
EmojiWrap (class in pipelinewraps.EmojiWrap), 11  
ENGLISH (utility.LanguageDetector.Language attribute), 15  
englishOrTagalog() (utility.LanguageDetector.LanguageDetector method), 15  
enrange() (in module pipelinewraps.AgeRangeWrap), 11  
enrange() (in module pipelinewraps.GenderWrap), 12  
enrange() (in module pipelinewraps.PostTimeWrap), 13  
evaluate() (in module Driver50), 1

evaluateKfold() (model.RootModel.RootModel method), 9  
evaluateKfold() (model.StackModel.StackModel method), 9  
execute() (in module Driver50), 1  
ExtractionWrap (class in pipelinewraps.ExtractionWrap), 11

## F

Feature (class in features.Feature), 4  
FeatureExtract (class in features.FeatureExtract), 5  
features (module), 8  
features.CharacterFeatures (module), 3  
features.Context (module), 4  
features.EmojisEmoticons (module), 4  
features.Feature (module), 4  
features.FeatureExtract (module), 5  
features.FunctionWordCount (module), 5  
features.Links (module), 6  
features.POSFeature (module), 6  
features.POSSequencePattern (module), 6  
features.Structure (module), 6  
features.TFIDF (module), 7  
features.WordCount (module), 7  
FILIPINO (utility.LanguageDetector.Language attribute), 15  
fit() (pipelinewraps.AgeRangeWrap.AgeRangeWrap method), 10  
fit() (pipelinewraps.CharacterWrap.CharacterWrap method), 11  
fit() (pipelinewraps.ContextualWrap.ContextualWrap method), 11  
fit() (pipelinewraps.EmojiWrap.EmojiWrap method), 11  
fit() (pipelinewraps.ExtractionWrap.ExtractionWrap method), 11  
fit() (pipelinewraps.FunctionWrap.FunctionWrap method), 12  
fit() (pipelinewraps.GenderWrap.GenderWrap method), 12  
fit() (pipelinewraps.ItemSelector.ItemSelector method), 12  
fit() (pipelinewraps.LinkWrap.LinkWrap method), 12  
fit() (pipelinewraps.POSSeqWrap.POSSeqWrap method), 13  
fit() (pipelinewraps.PostTimeWrap.PostTimeWrap method), 13  
fit() (pipelinewraps.SelectionWrap.SelectionWrap method), 13  
fit() (pipelinewraps.StackAgeRangeWrap.StackAgeRangeWrap method), 13

fit() (pipelinewraps.StackGenderWrap.StackGenderWrap method), 14  
fit() (pipelinewraps.StructureWrap.StructureWrap method), 14  
fit() (pipelinewraps.WordWrap.WordWrap method), 14  
fit\_transform() (features.FeatureExtract.FeatureExtract method), 5  
fixAcronymSpaces() (utility.PostCleaner.PostCleaner method), 16  
FunctionWordCount (class in features.FunctionWordCount), 5  
FUNCTIONWORDS\_FILENAME (features.FunctionWordCount.FunctionWordCount attribute), 5  
FunctionWrap (class in pipelinewraps.FunctionWrap), 12

## G

GenderWrap (class in pipelinewraps.GenderWrap), 12  
get\_Data\_from\_CSV() (in module Driver50), 2  
get\_keywords() (features.Links.Links method), 6  
get\_links() (features.Links.Links method), 6  
get\_list\_keywords() (features.Links.Links method), 6  
get\_liwc() (features.FeatureExtract.FeatureExtract method), 5  
get\_testing\_TFIDF() (features.TFIDF.TFIDF method), 7  
get\_title() (features.Links.Links method), 6  
get\_training\_TFIDF() (features.TFIDF.TFIDF method), 7  
getAdpositionCount() (features.FunctionWordCount.FunctionWordCount method), 5  
getAllFunctionWordCount() (features.FunctionWordCount.FunctionWordCount method), 5  
getArticleCount() (features.FunctionWordCount.FunctionWordCount method), 5  
getAuxillaryCount() (features.FunctionWordCount.FunctionWordCount method), 5  
getAveLengthWords() (features.WordCount.WordCount method), 7  
getAvgNCharacterPerParagraph() (features.Structure.Structure method), 7  
getAvgNSentencePerParagraph() (features.Structure.Structure method), 7  
getAvgNWordPerParagraph() (features.Structure.Structure method), 7  
getAvgNWordPerSentence() (features.Structure.Structure method), 7  
getClasses() (in module pipelinewrap-



- s.AgeRangeWrap), 11
- getClasses() (in module pipelinewraps.GenderWrap), 12
- getClasses() (in module pipelinewraps.PostTimeWrap), 13
- getClasses() (in module pipelinewraps.StackAgeRangeWrap), 14
- getClasses() (in module pipelinewraps.StackGenderWrap), 14
- getCombinedPOSTag() (features.POSFeature.POSFeature method), 6
- getConjunctionCount() (features.FunctionWordCount.FunctionWordCount method), 5
- getConnectionThesis() (addEngPOS.ConnectionFactory method), 2
- getConnectionThesis() (prepareedstthesis.ConnectionFactory method), 14
- getDictOfWordsMappedToOccurrence() (features.WordCount.WordCount method), 7
- getEmojis() (utility.PostCleaner.PostCleaner method), 16
- getEmojiTFIDF() (features.EmojisEmoticons.EmojisEmoticons method), 4
- getEnglishPOS() (features.POSFeature.POSFeature method), 6
- getEntropy() (features.WordCount.WordCount method), 7
- getFeatureNames() (features.TFIDF.TFIDF method), 7
- getFeatures() (features.Feature.Feature method), 4
- getHapaxDislegomena() (features.WordCount.WordCount method), 7
- getHapaxLegomena() (features.WordCount.WordCount method), 7
- getHonoresR() (features.WordCount.WordCount method), 8
- getInterjectionCount() (features.FunctionWordCount.FunctionWordCount method), 5
- getLabels() (features.EmojisEmoticons.EmojisEmoticons method), 4
- getLanguage() (utility.LanguageDetector.LanguageDetector method), 15
- getLanguage() (utility.LanguageDetector.LanguageDetector method), 15
- getLanguageDetailed() (utility.LanguageDetector.LanguageDetector method), 15
- getLolHmmCount() (features.WordCount.WordCount method), 8
- getNDifferentWords() (features.WordCount.WordCount method), 8
- getNParagraphs() (features.Structure.Structure method), 7
- getNSentenceBegLower() (features.Structure.Structure method), 7
- getNSentenceBegUpper() (features.Structure.Structure method), 7
- getNSentences() (features.Structure.Structure method), 7
- getNumberOfRepeatedPunctuationMarks() (features.CharacterFeatures.CharacterFeatures method), 3
- getNumberOfRepetitiveAlphaCharacters() (features.CharacterFeatures.CharacterFeatures method), 3
- getNumberOfSpecialChars() (features.CharacterFeatures.CharacterFeatures method), 3
- getNumberOfWhiteSpaces() (features.CharacterFeatures.CharacterFeatures method), 3
- getNWordsBegCapital() (features.WordCount.WordCount method), 8
- getNWordsWithRepLetters() (features.WordCount.WordCount method), 8
- getOccurrenceArray() (features.WordCount.WordCount method), 8
- getParagraphs() (features.Structure.Structure method), 7
- getPOSCount() (features.POSFeature.POSFeature method), 6
- getPosts() (in module batchprocessing), 2
- getPostsFromFile() (in module batchprocessing), 2
- getPredictions() (model.RootModel.RootModel method), 9
- getPredictions() (model.StackModel.StackModel method), 10
- getPronounCount() (features.FunctionWordCount.FunctionWordCount method), 5
- getProSentenceCount() (features.FunctionWordCount.FunctionWordCount method), 5
- getRatioOfHapaxDislegomena() (features.WordCount.WordCount method), 8
- getRatioOfHapaxLegomena() (features.WordCount.WordCount method), 8
- getRatioOfNetAbbrev() (features.WordCount.WordCount method), 8
- getRatioOfShortWords() (features.WordCount.WordCount method), 8
- getRatioOfUniqueWords() (features.WordCount.WordCount method), 8
- getSichelsS() (features.WordCount.WordCount method), 8
- getSimpsonsD() (features.WordCount.WordCount method), 8
- getSpecificFeatures() (in module Driver50), 1
- getTestingUser() (model.RootModel.RootModel method), 9

getTestingUser() (model.StackModel.StackModel method), 10  
getTestingX() (model.RootModel.RootModel method), 9  
getTestingX() (model.StackModel.StackModel method), 10  
getTestingy() (model.RootModel.RootModel method), 9  
getTestingy() (model.StackModel.StackModel method), 10  
getTotalNumberOfCharacters() (features.CharacterFeatures.CharacterFeatures method), 3  
getTotalNumberOfDigitalNumbers() (features.CharacterFeatures.CharacterFeatures method), 3  
getTotalNumberOfLetters() (features.CharacterFeatures.CharacterFeatures method), 3  
getTotalNumberOfUppercase() (features.CharacterFeatures.CharacterFeatures method), 3  
getTotalNumberOfWords() (features.WordCount.WordCount method), 8  
getTrainingUser() (model.RootModel.RootModel method), 9  
getTrainingUser() (model.StackModel.StackModel method), 10  
getTrainingX() (model.RootModel.RootModel method), 9  
getTrainingX() (model.StackModel.StackModel method), 10  
getTrainingy() (model.RootModel.RootModel method), 9  
getTrainingy() (model.StackModel.StackModel method), 10  
getWordLengthFreqDist() (features.WordCount.WordCount method), 8  
getYulesK() (features.WordCount.WordCount method), 8

## I

insertSpace() (utility.PostCleaner.PostCleaner method), 16  
ItemSelector (class in pipelinewraps.ItemSelector), 12

## L

Language (class in utility.LanguageDetector), 15  
LanguageDetector (class in utility.LanguageDetector), 15  
Links (class in features.Links), 6  
LinkWrap (class in pipelinewraps.LinkWrap), 12

## M

MAX\_LENGTH (features.POSSequencePattern.POSSequencePattern attribute), 6  
minePOSPatterns() (features.POSSequencePattern.POSSequencePattern method), 6  
model (module), 10  
model.Document (module), 8  
model.Post (module), 8  
model.RootModel (module), 8  
model.StackModel (module), 9

## N

normalizeUnicode() (utility.PostCleaner.PostCleaner method), 16

## P

pipelinewraps (module), 14  
pipelinewraps.AgeRangeWrap (module), 10  
pipelinewraps.CharacterWrap (module), 11  
pipelinewraps.ContextualWrap (module), 11  
pipelinewraps.EmojiWrap (module), 11  
pipelinewraps.ExtractionWrap (module), 11  
pipelinewraps.FunctionWrap (module), 12  
pipelinewraps.GenderWrap (module), 12  
pipelinewraps.ItemSelector (module), 12  
pipelinewraps.LinkWrap (module), 12  
pipelinewraps.POSSeqWrap (module), 13  
pipelinewraps.PostTimeWrap (module), 13  
pipelinewraps.SelectionWrap (module), 13  
pipelinewraps.StackAgeRangeWrap (module), 13  
pipelinewraps.StackGenderWrap (module), 14  
pipelinewraps.StructureWrap (module), 14  
pipelinewraps.WordWrap (module), 14  
populateMappingDictionary() (features.POSFeature.POSFeature method), 6  
POSFeature (class in features.POSFeature), 6  
POSSequencePattern (class in features.POSSequencePattern), 6  
POSSeqWrap (class in pipelinewraps.POSSeqWrap), 13  
Post (class in model.Post), 8  
PostCleaner (class in utility.PostCleaner), 15  
PostTimeWrap (class in pipelinewraps.PostTimeWrap), 13  
prepreedsthesis (module), 14  
process() (features.Context.Context method), 4

## R

removeEmojis() (utility.PostCleaner.PostCleaner method), 16  
retrievePOSTags\_docFrequency() (features.POSSe-

quencePattern.POSSequencePattern  
method), 6  
RootModel (class in model.RootModel), 8

## S

SelectionWrap (class in pipelinewraps.SelectionWrap), 13  
StackAgeRangeWrap (class in pipelinewraps.StackAgeRangeWrap), 13  
StackGenderWrap (class in pipelinewraps.StackGenderWrap), 14  
StackModel (class in model.StackModel), 9  
Structure (class in features.Structure), 6  
StructureWrap (class in pipelinewraps.StructureWrap), 14

## T

TAGLISH (utility.LanguageDetector.Language attribute), 15  
TFIDF (class in features.TFIDF), 7  
transform() (features.FeatureExtract.FeatureExtract method), 5  
transform() (pipelinewraps.AgeRangeWrap.AgeRangeWrap method), 10  
transform() (pipelinewraps.CharacterWrap.CharacterWrap method), 11  
transform() (pipelinewraps.ContextualWrap.ContextualWrap method), 11  
transform() (pipelinewraps.EmojiWrap.EmojiWrap method), 11  
transform() (pipelinewraps.ExtractionWrap.ExtractionWrap method), 11  
transform() (pipelinewraps.FunctionWrap.FunctionWrap method), 12  
transform() (pipelinewraps.GenderWrap.GenderWrap method), 12  
transform() (pipelinewraps.ItemSelector.ItemSelector method), 12  
transform() (pipelinewraps.LinkWrap.LinkWrap method), 12  
transform() (pipelinewraps.POSSeqWrap.POSSeqWrap method), 13  
transform() (pipelinewraps.PostTimeWrap.PostTimeWrap method), 13  
transform() (pipelinewraps.SelectionWrap.SelectionWrap method), 13  
transform() (pipelinewraps.StackAgeRangeWrap.StackAgeRangeWrap method), 13  
transform() (pipelinewraps.StackGenderWrap.StackGenderWrap method), 14  
transform() (pipelinewraps.StructureWrap.Struc-

tureWrap method), 14  
transform() (pipelinewraps.WordWrap.WordWrap method), 14

## U

UNKNOWN (features.POSFeature.POSFeature attribute), 6  
UNKNOWN (utility.LanguageDetector.Language attribute), 15  
updateEngPOS() (in module batchprocessing), 2  
updatePosts() (in module batchprocessing), 2  
URL (utility.DataCleaner.DataCleaner attribute), 15  
useLasso() (features.Feature.Feature method), 4  
USERNAME (utility.DataCleaner.DataCleaner attribute), 15  
utility (module), 16  
utility.DataCleaner (module), 15  
utility.LanguageDetector (module), 15  
utility.PostCleaner (module), 15

## V

VERB (features.POSFeature.POSFeature attribute), 6

## W

WordCount (class in features.WordCount), 7  
WordWrap (class in pipelinewraps.WordWrap), 14  
writePostsToFile() (in module batchprocessing), 2  
writeToExcel() (in module Driver50), 2

