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

Release

Cheng, Fernandez, Quindoza, Tan

August 17, 2017

Table of Contents

CHAPTER 1

thesis

1.1 Driver module

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 docs package

1.5.1 Module contents

1.6 features package

1.6.1 Submodules

1.6.2 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 ofcharacters

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

1.6.3 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

1.6.4 features.EmojisEmoticons module

class features.EmojisEmoticons.**EmojisEmoticons**

Bases: object

getEmojiTFIDF (*data*)

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

Returns TFIDF of the extracted emojis

getLabels ( )

1.6.5 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

1.6.6 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

1.6.7 features.FunctionWordCount module

class features.FunctionWordCount.**FunctionWordCount**

Bases: object

FUNCTIONWORDS_FILENAME = 'features/functionwords.txt'

getAdpositionCount (*text*)

Parameters **text** -- string to be counted for adposition words

Returns total number of adposition words

getAllFunctionWordCount (*text*)

Parameters **text** -- string to be counted for all function words

Returns total number of all function words

getArticleCount (*text*)

Parameters **text** -- string to be counted for articles

Returns total number of articles

getAuxillaryCount (*text*)

Parameters **text** -- string to be counted for auxillary words

Returns total number of auxillary words

getConjunctionCount (*text*)

Parameters **text** -- string to be counted for conjunctions

Returns total number of conjunctions

getInterjectionCount (*text*)

Parameters **text** -- string to be counted for interjections

Returns total number of interjections

getProSentenceCount (*text*)

Parameters **text** -- string to be counted for pro-sentence words

Returns total number of pro-sentence words

getPronounCount (*text*)

Parameters **text** -- string to be counted for pronouns

Returns total number of pronouns

1.6.8 features.Links module

class features.Links.**Links**

Bases: object

get_keywords (*link*)

get_links (*text*)

get_list_keywords (*text*)

get_title (*link*)

1.6.9 features.POSFeature module

class features.POSFeature.**POSFeature**

Bases: object

getCombinedPOSTag (*post*)

This method combines the resulting English and Filipino POS tags from the two separate POS Tagger. :param post: the document to be processed :return: returns a string of combined POS tags joined by "-"

getEnglishPOS (*text*)

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

Returns returns a string of the resulting POS tags joined by "-"

populateMappingDictionary ( )
This method populates the dictionary to include the list of mapped Filipino POS to its equivalent English POS by reading the contents of the mapping.txt file

1.6.10 features.POSSequencePattern module

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

MAX_LENGTH = 7

candidateGen (*fList*)

Param fList: list of POS sequence generated previously

Returns dictionary of POS sequences newly generated, all initialized with a value of 0

computeFairSCP (*key*, *count*)

Parameters
- **key** -- pos sequence
- **count** -- document frequency of pos sequence

Returns symmetrical conditional probability of given pos sequence

minePOSPatterns (*minsup*, *minadherence*)

Parameters
- **minsup** -- user-supplied minimum support
- **minadherence** -- user-supplied minimum adherence

Returns list of pos patterns that satisfy the thresholds

retrievePOSTags_docFrequecy ( )
creates a dictionary with the POS tags document frequency

1.6.11 features.Structure module

class features.Structure.**Structure**
Bases: object

ABBREVIATIONS_FILENAME = 'features/abbreviations.txt'

getAvgNCharacterPerParagraph (*text*)

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

Returns returns a float of the average number of characters per paragraphs detected in the text

getAvgNSentencePerParagraph (*text*)

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

Returns returns a float of the average number of sentences per paragraphs detected in the text

getAvgNWordPerParagraph (*text*)

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

Returns returns a float on the average number of words per paragraphs detected in the text

getAvgNWordPerSentence (*text*)

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

Returns returns a float of the average number of words per sentences detected in the text

getNParagraphs (*text*)

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

Returns returns an integer of the number of detected paragraphs in the text.

getNSentenceBegLower (*text*)

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

Returns returns an integer on the number of sentences beginning with an lowercase.

getNSentenceBegUpper (*text*)

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

Returns returns an integer on the number of sentences beginning with an uppercase.

getNSentences (*text*)

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

Returns returns an integer of the number of sentences detected in the text

getParagraphs (*text*)

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

Returns returns a list containing the detected paragraphs

1.6.12 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

1.6.13 features.WordCount module

class features.WordCount.**WordCount**

Bases: object

ABBREVIATIONS_FILENAME = 'features/abbreviations.txt'

getAveLengthWords (*text*)

Parameters **text** -- string to be used

Returns average length of words

getDictOfWordsMappedToOccurrence (*text*)

Parameters **text** -- string to be used

Returns dictionary of words mapped to occurrence

getEntropy (*text*)

Parameters **text** -- string to be used

Returns entropy

getHapaxDislegomena (*text*)

Parameters **text** -- string to be used

Returns hapax dislegomena

getHapaxLegomena (*text*)

Parameters **text** -- string to be used

Returns hapax legomena

getHonoresR (*text*)

Parameters **text** -- string to be used

Returns honores r

getLolHmmCount (*text*)

Parameters **text** -- string to be used

Returns number of lol's and hmm's with the use of regex

getNDifferentWords (*text*)

Parameters **text** -- string to be used

Returns number of unique words

getNWordsBegCapital (*text*)

Parameters **text** -- string to be used

Returns number of words beginning with a capital letter

getNWordsWithRepLetters (*text*)

Parameters **text** -- string to be used

Returns number of words with repeating letters

getOccurrenceArray (*text*)

Parameters **text** -- string to be used

Returns array of word occurrences

getRatioOfHapaxDislegomena (*text*)

Parameters **text** -- string to be used

Returns ratio of hapax dislegomena to total number of words

getRatioOfHapaxLegomena (*text*)

Parameters **text** -- string to be used

Returns ratio of hapax legomena to total number of words

getRatioOfNetAbbrev (*text*)

Parameters **text** -- string to be used

Returns ratio of net abbriavtions to toal number of words

getRatioOfShortWords (*text*)

Parameters **text** -- string to be used

Returns ratio of words with less than 3 characters to total number of words

getRatioOfUniqueWords (*text*)

Parameters **text** -- string to be used

Returns ratio of unique words to total number of words

getSichelsS (*text*)

Parameters **text** -- string to be used

Returns sichels s

getSimpsonsD (*text*)

Parameters **text** -- string to be used

Returns simpsons d

getTotalNumberOfWords (*text*)

Parameters **text** -- string to be used

Returns total number of words

getWordLengthFreqDist (*text*)

Parameters **text** -- string to be used

Returns an array with the word length frequency distribution from length 1 to 20

getYulesK (*text*)

Parameters **text** -- string to be used

Returns yules k measure

1.6.14 Module contents

1.7 model package

1.7.1 Submodules

1.7.2 model.Document module

class `model.Document.`**`Document`** (*content*, *posSequence*)

Bases: `object`

1.7.3 model.Post module

class `model.Post.`**`Post`** (*id*, *content*, *epos*, *fpos*)

Bases: `object`

1.7.4 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

1.7.5 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

1.7.6 Module contents

1.8 pipelinewraps package

1.8.1 Submodules

1.8.2 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

1.8.3 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*)

1.8.4 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*)

1.8.5 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*)

1.8.6 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*)

1.8.7 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*)

1.8.8 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

1.8.9 pipelinewraps.ItemSelector module

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

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

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

transform (*data_dict*)

1.8.10 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*)

1.8.11 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*)

1.8.12 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

1.8.13 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*)

1.8.14 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

1.8.15 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

1.8.16 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*)

1.8.17 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*)

1.8.18 Module contents

1.9 prepareedsthesis module

class prepareedsthesis.**ConnectionFactory**
Bases: object

getConnectionThesis ( )

prepareedsthesis.**addposts** ( )

prepareedsthesis.**addusers** (*limit=None*)

1.10 utility package

1.10.1 Submodules

1.10.2 utility.DataCleaner module

class utility.DataCleaner.**DataCleaner**
Bases: object

URL = 'URL'

USERNAME = 'USERNAME'

clean_data (*post_content*)

clean_email (*post_content*)

1.10.3 utility.LanguageDetector module

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

ENGLISH = 0

FILIPINO = 1

TAGLISH = 2

UNKNOWN = -1

getLanguage (*code*)

Parameters **code** -- integer assigned to represent a language

Returns the meaning of the codes

class utility.LanguageDetector.**LanguageDetector**

Bases: object

englishOrTagalog (*string*)

Parameters **string** -- string to be identified as either English or Tagalog

Returns strings of "en" (English) or "tl" (Tagalog)

getLanguage (*text*)

Parameters **text** -- string to be language detected

Returns "ENGLISH", "FILIPINO" or "TAGALOG", else "UNKNOWN"

getLanguageDetailed (*text*)

Parameters **text** -- string to be language identified

Returns detailed probabilities of the languages detected, else "UNKNOWN"

1.10.4 utility.PostCleaner module

class utility.PostCleaner.**PostCleaner**

Bases: object

changeEmojisToText (*postContent*)

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

Returns returns a string where the detected emojis are replaced into the label "EMOJI"

changeForeignToText (*postContent*)

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

Returns returns a string where the detected foreign languages are replaced into the label "FOREIGN"

changeLinkToText (*postContent*)

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

Returns returns a string where the links are replaced into the label "URL"

fixAcronymSpaces (*postContent*)

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

Returns returns a string with fixed acronym spaces

getEmojis (*postContent*)

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

Returns returns a list of emojis detected in the text

insertSpace (*postContent*)

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

Returns returns a string formatted so that emojis that are sticked together will have a space in between them for easier processing later on

normalizeUnicode (*postContent*)

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

Returns returns a text with the normalize unicode string

removeEmojis (*postContent*)

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

Returns returns a string without the emojis

1.10.5 Module contents

- *Index*
- *Module Index*
- *Search Page*

a

b

c

d

f

m

p

u

A

B

C

D

E

F

G

I

L

M

N

P

R

S

T

U

W