OVERVIEW

The Phone Number Plugin (PNP) takes in a dataset column that is assumed to be converted to a list of strings. The PNP first removes any leading and trailing spaces in the strings. The PNP then removes any null (None) values or any string values denoting a null value ('NA', etc.) found in the list. Then, the PNP returns a score for each value based on how well the PNP is confident that that value is a phone number. After scores have been assigned, any outlier scores (anything not within 2 standard deviations from the mean) are removed before taking and returning the final average of the scores (the final confidence score for the entire column).

FUNCTION DEFINITIONS

DEPENDENCIES: re

get_confidence_score()

The *get_confidence_score()* function takes in a *string* called *col_name* and a list of *strings* called *col_vals*. *col_vals* is presumed to be a dataset column that has been converted to a list of *strings*. *col_name* is presumed to be the dataset column name. The function first checks to see if '*phone'* is detected within the column name. If so, the *col_name_check* is set to *True* and *False* otherwise. The function then creates an empty *double* list called *scores* to contain the confidence score for each value in *col_vals*. The function then modifies *col_vals* by calling on the *remove_lead_trail_space()* then the *remove_null()* function. After, the function iterates through each value in *col_vals* and calls on the *get_elem_score()* function to append each score returned from that function to the *scores* list. Afterwards, the *remove_outliers()* function is called on the *scores* list. Finally, the average (which is *double* type) of the *scores* list is returned (this is the final confidence score for the entire column and is capped at 100.0) or 0.0 is returned if *scores* is empty.

get_elem_score()

The *get_elem_score()* function takes in a bool called *col_name_check* and a *string* called *elem*. *col_name_check* is presumed to be *True* if *'phone'* is detected in the column name and *False* otherwise. *elem* is presumed to be a value from a list of *strings*. If *col_name_check* is *True*, then a score boost of x1.1 is set. If not, the boost is set to x1.0. The function then uses regular expressions in each if-statement to determine what confidence score (which is *double* type) to return. *See the PHONE NUMBER CONVERSIONS section for more information on the confidence score.*

For whatever non-zero confidence score is chosen, 5 points will be deducted per unmatched character detected at both the beginning and end of the string. For example, "**&**(555)555-5555**&**" will give a confidence score of 90 instead of 100 because of the two additional ampersands detected.

For whatever non-zero confidence score is chosen, a score boost will be applied. This is set to *x1.1* if the column name was detected to have *'phone'* in it and set to *x1.0* if not.

*Note: Points will not be detected for typos within the middle of the format [555-5**k**5-5555] or any typos that are missing from a format [555-55-5555 (missing a third number in the second column)]. This is because regular expressions cannot find a match with these types of mistakes.*

remove_null()

The *remove_null()* function takes in a list of *strings* called *col_vals*. *col_vals* is presumed to be a dataset column that has been converted to a list of *strings*. The function uses list comprehension to return a list where any *None* values have been removed and where any

strings denoting null values have been removed. The list of strings that denote values are noted as: *['NA', 'N/A', 'na', 'n/a', 'Na', 'N/a', '']*.

remove_lead_trail_space()

The *remove_lead_trail_space()* function takes in a list of *strings* called *col_vals*. *col_vals* is presumed to be a dataset column that has been converted to a list of *strings*. The function uses list comprehension to return a list where all the string elements have leading and trailing space removed.

remove_outliers()

The *remove_outliers()* function takes in a list of *doubles* called *scores*. *scores* is presumed to be the list of confidence scores of each value in a dataset column. *scores* is immediately returned if the *scores* list is empty. The function first creates an empty *set* to contain outliers found. The function then calculates the average and the standard deviation of *scores*. Then the function iterates through each score in *scores* and determines whether the score is within 2 standard deviations from the average. If the score is not found within that range, that score is added to the outlier list. Because 95% of values in any distribution (*Statistic Found Here*) fall within that 2 standard deviation range, generally only extreme outliers will be removed. Finally, list comprehension is used to remove any found outliers from the *scores* list then that modified list is returned.

PHONE NUMBER CONVERSIONS

From the phone number formats in this article, https://blog.insycle.com/phone-number-formatting-crm, these phone number formats were chosen:

Phone Number Format	Regex Form	Confidence Score
(555)555-5555	\(\d{3}\)\s?\d{3}-\d{4}	100.00
(555) 555-5555	\(\d{3}\)\s?\d{3}-\d{4}	100.00
+555-555-5555 **(also matches 80)**	(\+1?\s?\d{3}[\.-]\d{3}[\.-]\d{4})|(\+1\d{10})	90.00
+1 555.555.5555 **(also matches 80)**	(\+1?\s?\d{3}[\.-]\d{3}[\.-]\d{4})|(\+1\d{10})	90.00
+1 555-555-5555 **(also matches 80)**	(\+1?\s?\d{3}[\.-]\d{3}[\.-]\d{4})|(\+1\d{10})	90.00
+15555555555 **(also matches 20)**	(\+1?\s?\d{3}[\.-]\d{3}[\.-]\d{4})|(\+1\d{10})	90.00
555.555.5555	(1-\d{3}-\d{3}-\d{4})|(\d{3}[\.-]\d{3}[\.-]\d{4})	80.00
1-555-555-5555	(1-\d{3}-\d{3}-\d{4})|(\d{3}[\.-]\d{3}[\.-]\d{4})	80.00
555-555-5555	(1-\d{3}-\d{3}-\d{4})|(\d{3}[\.-]\d{3}[\.-]\d{4})	80.00
555 555 5555	\d{3}\s\d{3}\s\d{4}	65.00
5555555555	\d{10}	20.00

Generally, there is one regex expression per confidence score category.

Though several of the formats in the 90 category match those in a below category, this should **not skew results**. This is because the if statements are set up in a way that any match with the 90 category is caught before matches within the 80 or 20 category.

If none of the above formats are matched, a confidence score of 0.0 is returned.
If formats are matched with additional unmatched characters at the beginning and end of the string, 5 points are deducted per additional unmatched character from the original score.

The confidence scores for each format was chosen due to the specificity of the format. For example, the *(555) 555-5555* format generally does not denote anything except for phone

numbers (hence the *100.0* confidence scores) whereas a pure 10-digit number (*5555555555*) could denote a phone number but also could very well represent a generic ID (hence the *20.0* confidence score). *These confidence score pairings are subject to change either to team majority vote or mentor discretion.*

IMPLICATIONS FOR FUTURE

In the future, we would like to handle international phone numbers instead of just U.S. phone numbers. This means adding more regular expression formats or using an external API to check if an item is a valid phone number or not.

In the future, we would like to catch typos that occur within the middle of a format or typos that have missing characters from a format. This means not using regular expressions and using another method of string similarity comparison.

Since this plugin uses the same format as other plugins that use regular expression, we would like to have one .txt file containing all data types' regular expressions and one Python file that handles all regular expression cases instead of having separate Python files for each data type.