Address Linkage Key with AnalyticsIQ
Milestone 3 Presentation

Project Owners :

Warren Smith

Jin Wang

Project Team :

Kat McElveen, Team Lead

Saritha Gudala, Technical Writer

Kat Greer, Technical Specialist

Byron Smith, Technical Specialist

IT7993 Capstone, Spring 2021

April 9, 2021

Project Overview

- Develop a tool/set of tools to compute the delivery point (DPBC) of a given address following the guidelines of USPS.
- The tool will be initially developed and tested for properly formatted addresses.
- As part of further improvements, it will be tested for improper address format and/or for the presence of certain types of delivery points such as high-rise buildings.

Project Progress Summary

- Project team continued weekly meetings (Wednesday and Friday) to assess the progress and discuss further plans.
- Development of secondary rules.
  - Team has self-assigned tasks according to Secondary DPBC rules
  - After the discussion with client, some of the secondary rules were eliminated which are out of scope and availability of sample data.
  - Team utilized Jupyter Notebooks for local testing and prototyping
  - User Defined Functions, Functions, and Test Cases were created for the applicable rules
  - Team used Git (via TortoiseGit and command line) to commit and push local edits, which were then merged into the master branch upon approval of Project Owner

Milestone 3 Final Deliverables Status Update

Complete:

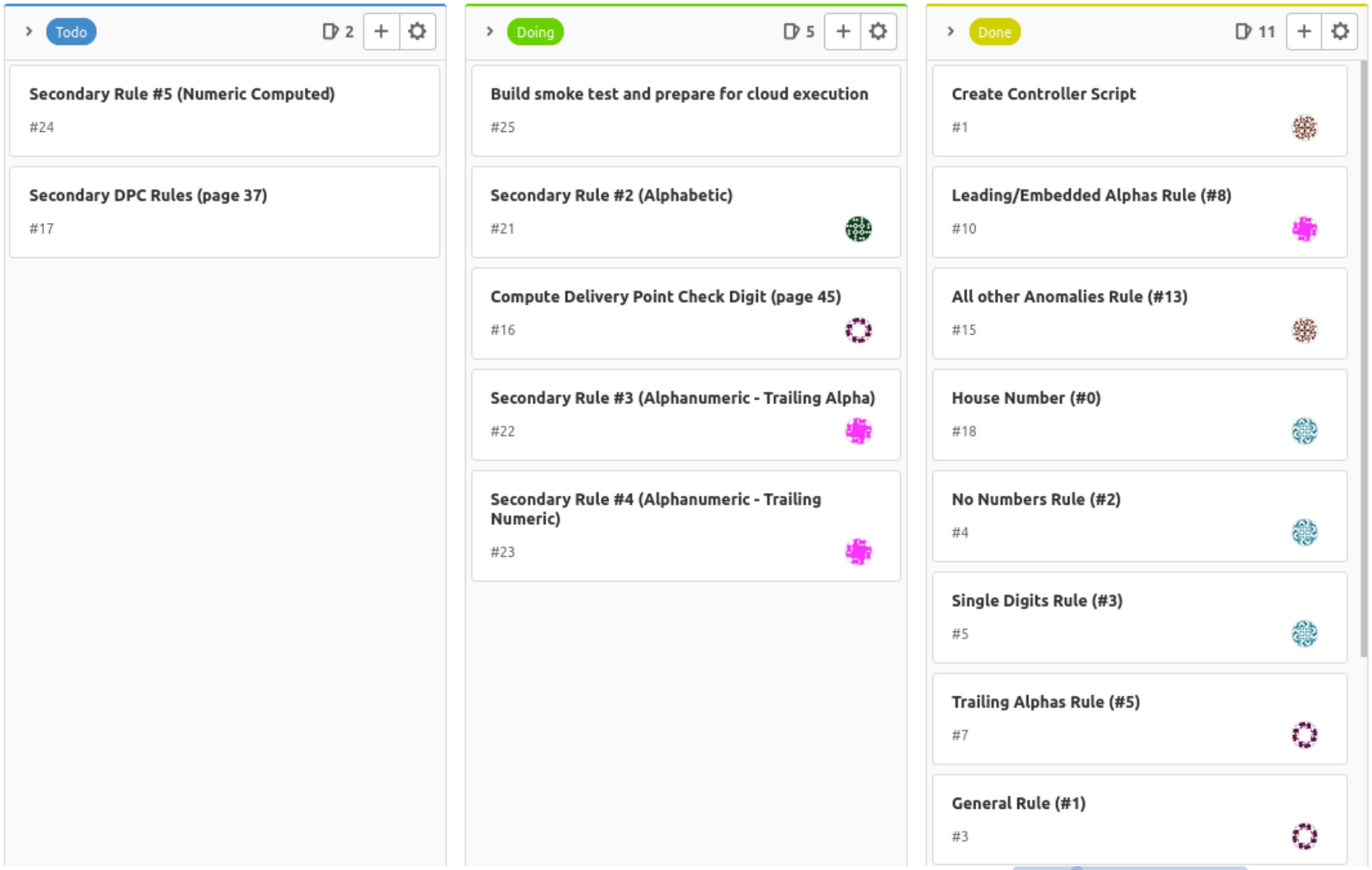
- Completed the implementation and testing of minimally viable product using primary rules
- Expanded house number extract (feeds into primary rules) to account for PO Boxes, Rural Routes, Highway Contract addresses
- Created functions and test cases for some of the secondary rules
- Created function to generate Delivery Point Check Digit

In Progress:

- Completion and testing of secondary rules
- Testing the calculation of the Delivery Point Check Digit

Changed:

Task Status Update from Gitlab



Delivery Point Barcode Rules (Secondary)

Rule 1: Numeric Simple Rule

- The Numeric Simple Rule applies to situation in which the secondary address value only contains numbers (0-9) excluding fractional values or special characters, and the numeric value in the hundreds or thousands place equal zero.
- The last two digits of the secondary number must become the DPC.
- Ex:

Secondary Value	DPC
1	01
2	02

Secondary Value	DPC
10001	01
10002	02

Delivery Point Barcode Rules (Secondary)

Rule 2: Alphabetic Rule

- The Alphabetic Rule is used when the secondary address value contains only alphabetic characters, excluding fractional values or special characters. Compute the DPC using only the rightmost alphabetic character.
- Each character of the alphabet is assigned a unique DPC based on a progressive substitution starting at 73 and continuing through 98 (e.g., A = 73, B = 74, Z =98)

Rule 3: Alphanumeric Rule – Trailing Alpha

- The Alphanumeric Numeric Rule/Trailing Alpha applies to alphanumeric secondary addresses in which the last character is an alphabetic character within the range A to Z. Form the DPC from the secondary address according to the following formula: DPC = MOD (X + (10 · Y))
- In this equation, “X” equals the conversion value of the rightmost alphabetic character from the alphanumeric conversion table, and “Y” equals the rightmost non-fractional numeric form value.

Delivery Point Barcode Rules (Secondary)

Rule 4: Alphanumeric Rule - Trailing Numeric

- The Alphanumeric Rule – Trailing Numeric applies to alphanumeric secondary addresses with trailing numbers. Derive the DPC from the secondary address according to the following
- formula: DPC = MOD ((X · 10) + Y)/100)

Rule 5:Numeric Computed Rule

- The Numeric Computed Rule applies to numeric secondary addresses when the value of the combination of digits in the hundreds and thousands places is greater than zero.

Note: See Rule 1, if the value in the hundreds and thousands place equals 0.

- Compute the DPC from the secondary address according to the following formula: DPC = 25 · (MOD (X/4)) + MOD (Y/25)

Compute Delivery Point Check Digit

- The delivery point check digit also known as correction character.
- This is a number that is added to the sum of the other digits in the delivery point barcode (DPBC) to yield a number that is a multiple of ten.

Example:

5-Digit ZIP Code	=	12345
ZIP + 4 Code	=	6789
Delivery Point Code	=	01
Sum of 1 + 2 + 3 + 4 + 5 + 6 +7 +8 + 9 + 0 + 1	=	46
Add check digit (4)		+ 4
Equals Multiple of 10		50

What is User Defined Function?

- Built-in functions are those that are already defined in **Python** libraries and we can call them directly. User defined functions are those that one can define to customize the function as per requirement and then call them wherever its needed.
- The Sample code in the next slide rounds of the delivery point check digit to multiple of 10.

Sample Code of User Defined Function?

In [9]:
```
def ceil_checksum(number, multiple):
    # passing default values
    if number is None:
        return None
    try:
        return math.ceil(float(number)/multiple)*multiple
    # to raise back  exception to the caller
    except ValueError as e:
        return None
```

In [10]:
```
@f.udf("int")
def spark_ceil_checksum(number, multiple):
    return ceil_checksum(number, multiple)
```

In [14]:
```
ceil_checksum(327.0,10)
```

Out[14]: 330

Challenges

- Choosing the correct tool (udf, case statement, etc.) to use in a situation. When creating a function to create a dpc2 for items meeting the requirements for the Alphabetic rule creating a udf wasn't working properly when used in the when .otherwise statement required to not overwrite the dpc2 values created by the numeric simple rule. To resolve this a case statement was used to create an intermediate df instead.

REFERENCES

1. USPS Publication 28 (Postal Addressing Standards)
2. USPS CASS Technical Guide
3. https://regexr.com/

THANK YOU