Project: **Customer Request Learning Team-Suite (CRLT)**

Authored on: 01/092020, 02/09/2020, 03/09/2020, 04/09/2020, 05/09/2020

Team: AOEC

Team members:

1. K.S.Venkatram and 2. Aakkash K V

Hackathon: Hack Vista, Bengaluru

Dev Post URL https://devpost.com/software/customer-request-learning-team-suite-crlt

Summary

An Automotive Parts Manufacturer and related Service Centre framework needs to group similar service or remedy requests about parts into a new classification, so that the classification can be used to send out standard or pre-defined resolution.

The issue being that the Automotive Parts Manufacturer focuses on manufacturing multiple categories of parts, where the SMART Manufacturing concept with sensor enabled control systems improve production volume and achieve high levels of quality.

However the staff allocated to receiving customer requests need to have sufficient product knowledge and also need to remain trend-savvy to resolve queries, requests or issues. Some of this is time consuming and number intensive.

The current economic downturn has affected the company and it needs to control costs.

The cost cutting drive is on, where the calling is for a reduction in the money spent on acquiring markets, ensuring sales & enabling of associated customer services.

Reduction in manpower is a part of this cost cutting strategy.

The financial reports highlight there is a decrease in market share, so the company needs to manage production volume and also respond effectively for all service or remedy requests but with new reduced time-spent-per-request-or-call “rules & regulations”.

The multiple parts being manufactured

The Automotive Parts Manufacturer focuses on the following part manufacturing:

Focus 1: Tyres

Focus 2: Mechanical parts

Focus 3: Body Shell Assembly

Focus 4: Interior Trim

Focus 5: Seats

Focus 6: Small parts like rotors in electric motors for injection pumps

Focus 7: Modules like the top and bottom parts of the fuel tank

Focus 8: Exhaust systems and parts like the muffler and exhaust pipe

Focus 9: Electronic parts like populated printed circuit boards and their components

The problem on hand is to reduce the time spent on each request or call for the parts, where customers could need customer service or remedial steps.

What it does (Solution and Approach)

The Customer Request Learning Tool (CRLT) reviews customer requests to find that service or remedy requests coming to the Centre

1. Can have repetitive text in the description or problem details

2. Can have some standard steps that the customer can take to find resolution for some questions or problems

3. Can have responses / resolutions that do not need human intervention

Inference

1. The CRLT experts infer that buckets of service or remedy requests can be created through a machine learning algorithm based on the past service requests.

2. For each bucket, a set of corresponding standard resolutions can be associated to reduce time spent per request or call.

3. Whenever a new service or remedy request comes to the Service Centre, a machine learning algorithm would identify the bucket to which the new request should be tagged to, where this should help come up with a standard resolution and then help report the same across to the customer.

Methodology

In the solution,

1. The service or remedy requests in the repository are clustered using a combination of (a) **Text-analytics** of “text fields” with select

descriptions, (b) **the time estimated or taken to resolve each request and** (c) **a categorization variable** that categorizes the nature of request

2. The Text-analytics technique is based on **Word2Vector**

3. The clustering technique is based on **DBSCAN**

4. The **Cosine similarity algorithm** is used to classify service or remedy requests to fit within one of the buckets created (where this is based on text categorization)

How we built it

We at AOEC are developing the idea using the Python & Anaconda framework and different libraries for data analysis, array processing, Natural language processing, Text-analytics & clustering, visualizing of clusters, request or remedy description similarity

The details of the libraries follow:

Specific libraries to load data, perform computation and display output are

(a) Pandas – Data acquisition library

(b) numpy – Array processing library

(c) nltk.data and nltk.corpus – Natural language processing library

(d) gensim and gensim.models – for text analytics and clustering, where the Word2Vector function is used

(e) gensim.models.keyedvectors – to import keyed vectors

(f) matplotlib – for visualizing clusters

(g) sklearn.cluster – to import DBSCAN for clustering

(h) sklearn.metrics.pairwise – to import cosine-similarity to find out request description similarity

Code snippets in the proof of concept (step wise)

(1) To import libraries and functions

(2) To load data

(3) For filtering of requests based on groups for “parts and requests categorization” (where there are 9 Focus groups and one Transaction Hub category, it is noted that the Transaction Hub category can be exploded further when the proof of concept is developed into a complete application)

(4) Text analytics to create the training data for the machine learning algorithm

(5) Running of the clustering function

(6) Assigning of a new service or remedy request to a correct bucket based on the cosine-similarity function