Property and Casualty Risk - Product Group

December 14, 2019

0.1 Project Abstract

At any Property and Casulaty insurance carrier, one of the biggest challenges is deciding which insurance products are right for their clients' needs. Using data on the carrier's clients, the goal of this research is to explore different machine learning techniques to predict the type of products the company should promote and sell to clients to secure new business and also deploy an optimized model in a Production environment so that other business applications and systems can use it. The research uses a simulated dataset similar to the real world dataset with a total of 43 features to conduct this analysis and identify an optimized model that can predict the product group which can then be utilized by the internal business unit to make informed decision.

This research demonstrates a preliminary work to build a base model that can predict the product group and optimize it based on initial observations to create a final model that can be implemented by utilizing the client purchase history and client census information. The final model will offer actionable insights and enable the company to make informed decision on coverage/product selection for different customer segment with help of data analysis.

For prediction, the research explored both Supervised and Unsupervised learning methods although the dataset available is labeled. In Supervised learning, different classification techniques were explored including LogisticRegression classifier, Naive Bayes classifier, Decision Tree classifier, an ensemble method (Random Forest), KNN classifier and in Unsupervised learning one clustering technique specifically KMeans was explored. The results were evaluated on accuracy primarily, F1 score and Silhouette coefficient for Unsupervised learning.

The results indicate that the tree and ensemble methods exhibited high accuracy compared to other methods. The Decision Tree classifier exhibited high accuracy consistently on different variants of the dataset but from a operationalization point of view, the research recommends implementing a random forest classifier because it offered better performance on larger dataset.

0.2 Introduction

In Property and Casualty insurance, underwriting excellence has been a prime focus for a very long time but achieving underwriting excellence and improvement could be a massive task. Loss ratio and expense ratio are primary indicators of financial performance of a insurance carrier and there is significant variability on these indicators among different insurance carriers. Over the past few years, the Property and Casualty carriers have witnessed great deal of success and setbacks in improving underwriting processes specifically on coverage selection. According to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019), underwriting excellence and improvement focusses on five major building blocks namely portfolio steering, pricing adequacy, risk selection, capacity optimization, and coverage design. According to (Chester, A. & Ebert,S. & Kauderer,S. &

McNeill,C., 2019), As technical advancements in the big data space, data analytics and artificial intelligence/machine learning advance and new applications emerge, the aforementioned building blocks evolve and become more data-driven. According to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019), Organizations are starting to adapt these technologies while also focusing on important enablers such as distribution, culture, digital, and strategy. According to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019), Collectively, these building blocks and enablers are the foundation of underwriting excellence.

According to (Boodhun, N. & Jayabalan, M., 2018), Risk profiles of insured's business are analyzed end-to-end by the underwriting team in Property and Casualty insurance business. The role of the underwriter is to ensure that the risks being insured as part of the insured's business are thoroughly reviewed, careful coverage selections are made and premiums are booked accurately ensuring profitability for the insurer. Risk selection is a popular term among Property and Casualty insurance carriers, which refers to choosing the right coverages and exclusion that are appropriate for the insured business and the estimated level of potential risks. The coverage or risk selection can be achieved using predictive analytics, appropriate tools and technologies and historical data that is available. The conventional methods of risk selection practiced over several years have proven to be time consuming and inefficient. It had delayed quoting and issuance for insureds and resulted in lost business because the insured is unable to wait until the correct underwriting decisions are made. Predictive analytics has been used for several years in the claims space to detect severity and probability recovery but there has not been extensive research in this area for coverage or risk selection (in our case Product Group selection). According to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019), it is easier said than done due to the fact underwriting has traditionally been slow to change but the perils it faces have been changing quickly. The objective of this research is to employ predictive analytics to determine the right coverage for the insured using historical data and recommend optimized model that can classify the correct product group for the risks being covered.

0.3 Literature Review

Over several years, Property and Casualty insurers have been trying different methods and processes to sell their products efficiently but the underwriting process involves a sequence of steps before a insurance policy is quoted and issued. According to ("Fundamentals of Underwriting", 2019), the underwriting process involves gathering information about the insured and their businesses and assessing the potential risks to those businesses. The line and staff underwriters then assess the nature of business being insured and decide on the risk selection and coordinate the management-level underwriting activities. Although there are no standard responsibilities between line and staff underwriters, every insurance carrier determines what responsibilities fall under each of these roles. There are two key responsibilities of either the line or staff underwriter depending on the insurer they are associated with and they are - Risk Selection and Selecting Insureds (particularly avoiding adverse selection). These two activities have a direct impact on the profitability goals of the insurer in terms of premium being booked, the overall book of business and potential for high severity claims.

According to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019), underwriting excellence is of prime importance to a Property and Casualty insurer's performance and that underwriting excellence encompasses risk selection which again according to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019) should be blended with human judgement and data-driven analytics. According to (Chester, A. & Ebert,S. & Kauderer,S. & McNeill,C., 2019), data-driven analytics can

immensely supplement human judgement in doing risk selections and would enable underwriters make more informed risk selections based on the business in overcapitalized markets.

According to ("Fundamentals of Underwriting", 2019) and (Berger,L. et al, 1991, p.1-420), underwriting account selection is another major responsibility that rests on the line or staff underwriters and they should be particularly cautious about avoiding adverse selections. Adverse selection is situation where the insurer does not have full knowledge of the insured's businesses and the risks involved and end up underwriting policies for those high risk businesses with a potential for high risk claims. Adverse selection and risk selection are two related activities and these can be performed effectively and potential negative impact to profitability for the insurer could be avoided if the risks or the product groups as in our problem are determined correctly using predictive analytics, which is the goal of this research.

0.4 Data Description

The objective of the research is to predict the product group for different customers and the associated risks. The research explores different classification models for such a setting to predict the classes using a set of features given the basic assumption that the examples are independent and identically distributed. The target variable seems to have distinct classes and it is important to compare the performance of different machine learning algorithms consistently.

The client purchase history dataset contains 1314922 records with 36 attributes and the client census data dataset contains 514976 records with 8 attributes. The client purchase history contains information about the insured business, type of risks, premium and other policy related information while the client census dataset contains mostly demographic information about the insured and the associated business. The target variable appears to have missing values and so are some of the attributes in the datasets. The missing values will be imputed using an appropriate imputation startegy during the data pre-processing phase. The datasets contains both categorical and numerical attributes, there are about 21 categorical and 22 numerical attributes. Based on the initial review of the dataset some of the attributes seem to have overlapping relationship with other attributes in the dataset and need to be studied further using correlation analysis. There are few attributes like the client key, duns number etc. that seem to have low significance to model training and evaluation because they are more informational and do not explain variability, such attributes should be reviewed and dropped if they are not needed during the data pre-processing phase. The two datasets can be merged on the client key or client id to create a combined dataset to use in the modeling process.

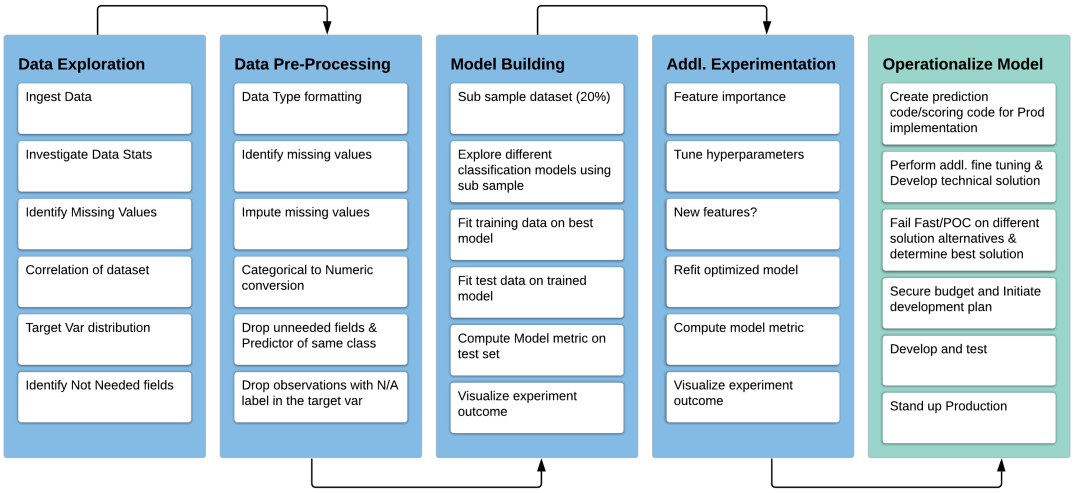
Considering the size of the datasets, it would be a good approach to evaluate different models using a stratified sub sample (20%) of the data and based on the outcome, a model can be trained and evaluated with the full dataset. First steps would be cleaning and pre-processing of the data before training the model and the initial training/evaluation would be done using basic parameter setting and upon observing the outcome of this initial model, it would be ideal to conduct further experimentation to tune the hyper-parameters if the required accuracy ($> 80\%$) is not achieved through any of the models and if needed engineer additional features from the provided input data to improve the accuracy of the final model.

Data can be found at: https://github.com/ananthajanakiraman/Applied-Data-Science-Project (File Names: client_census_data.csv, client_purchase_history.csv)

0.5 Model Pipeline

Here is a visual representation of the perceived model pipeline including the different phases of data preparation and data analysis. The visualization provides a brief description of the sub-tasks that should be conducted in each of the individual phases leading up to operationalization of the final optimized model in Production. The visualization serves as a blue print for the forthcoming sections.

[1]:



0.6 Data Load

The first step to the modeling process is to ingest the two datasets - client purchase history and client census data that are available as csv files in the working directory using a well known pandas function in Python.

0.6.1 Merge dataset

The loaded datasets from the above step can be merged into one dataset using the client key attribute so that there will be one combined dataset to work with for the modeling process.

```
Shape of client purchase history dataset:  (1314922, 36)
Shape of client census dataset:  (514976, 8)
Shape of final merged dataset:  (1314922, 43)
```

0.7 Data Exploration

The Exploratory Data Analysis (EDA) will include both univariate and bivariate analyses. The exploratory data analysis would enable a researcher to visualize and make sense of different distributions that are illustrated by the features or attributes on the dataset. A bivariate analysis represents the relationship between two variables and permits the researcher to see at a glance the degree and pattern of relation between the different predictor variables and the target

attribute and perform deeper analysis. It is very important to analyze how the independent variables impact the target variable and affect the accuracy of the model. As part of this research, visual analytics on the dataset was conducted and meaningful insights were derived from the dataset. The data will be visualized using graphs, plots and bars because the visual medium helps in researching the entire dataset without having to manually investigate for individual anomalies, understand the data better and also the relationships between the variables. The visualization will help decide what type of models would explain the data better.

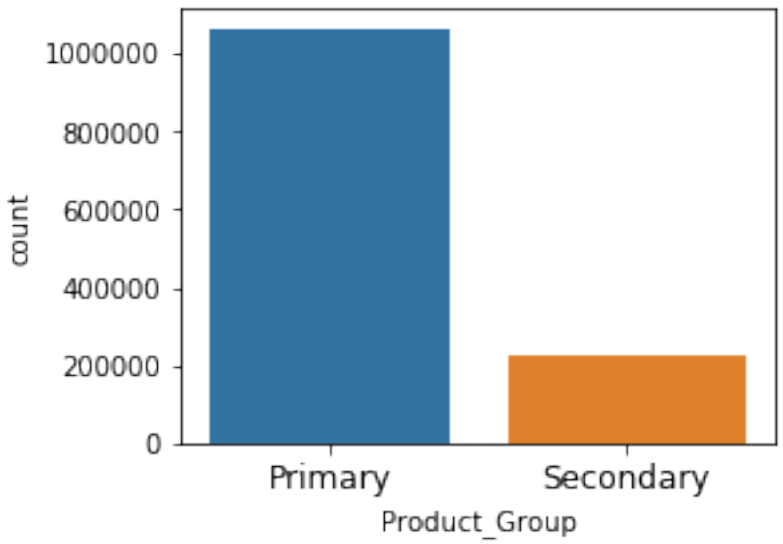
0.7.1 Preliminary Data Statistics

Following are some of the important data metrics extracted from the dataset that are interesting to me.

```
# Total count in client purchase history  - 1314922
# Total count in client census dataset    - 514976
# Total count of missing Product_Group    - 27871
# Total count of missing Industry Segment - 607814
# Total count of missing Fee Revenue      - 1018400
# Total count of missing DUNS Number      - 606951
# Total count of missing Client Offices   - 614126
```

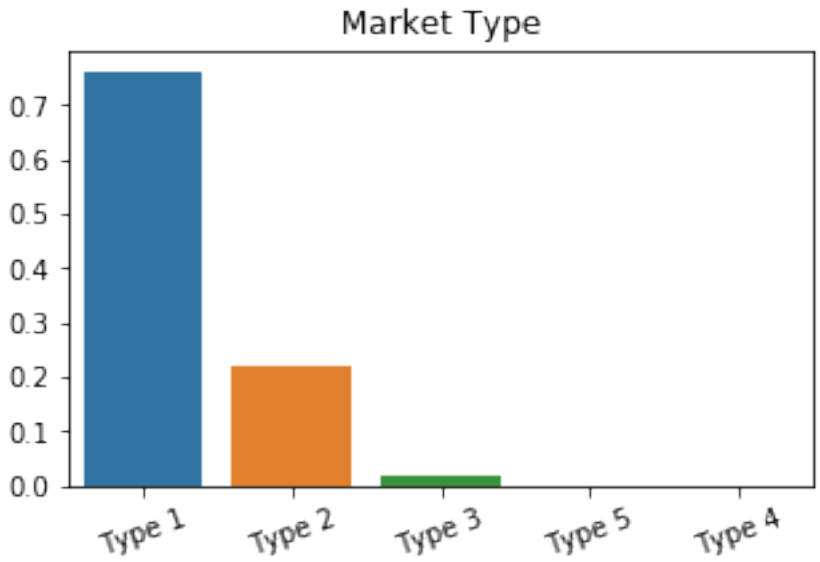
0.7.2 Target Variable (Class) Frequency

Based on class frequency visualization, 80% of the observations are classified as the ``Primary'' Product Group which is significant observation. The ``Primary'' product group seems to be encompassing generic coverages for different risk categories while ``Secondary'' seems to cover more specialized risk exposures.



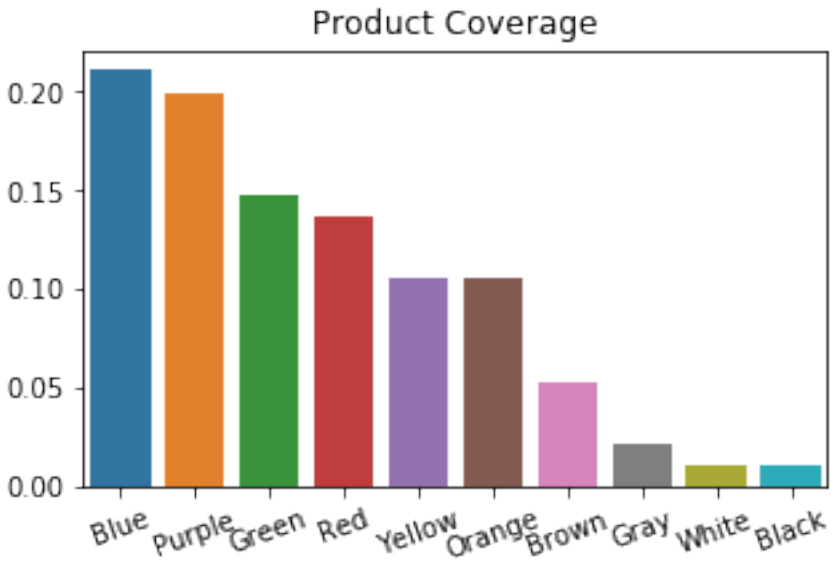
0.7.3 Market Type Distribution

The Market Type distribution visualization below proves that this particular attribute will have significant effect on the final prediction. There are several observation that are categorized as ``Type-1'' which may mean that those observations could potentially be small businesses because of the number of employees and overall income while the other categories could represent middle markets and other specialized business units.



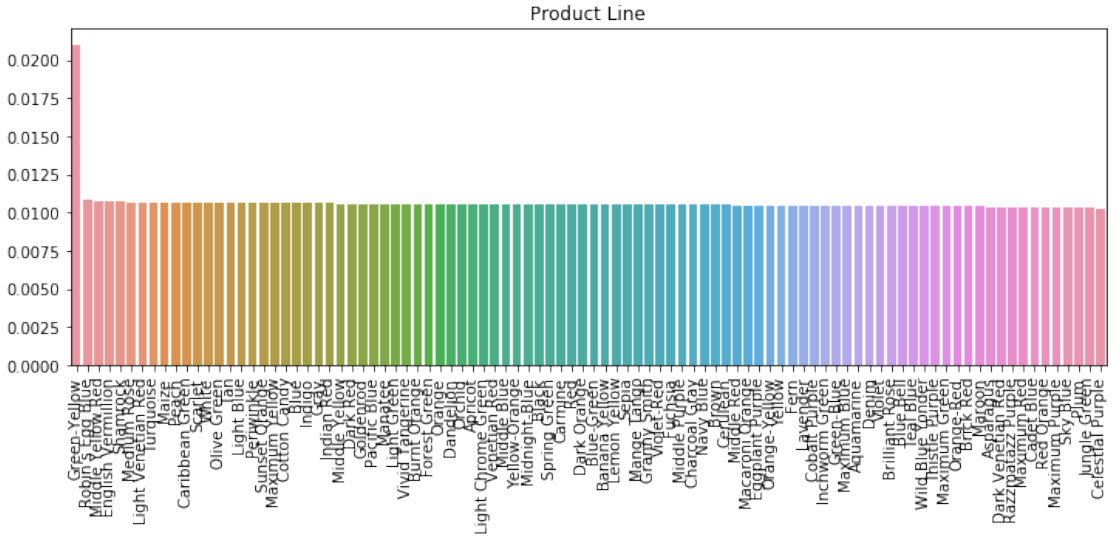
0.7.4 Product Coverage Distribution

The Product Coverage distribution below shows that there are at least 10 different product coverage categories and there are large number of observations that fall under the Blue and Purple coverage categories, the Green, Red, Yellow and Orange coverage categories are more or less evenly distributed across the observations and there is a low percentage of other specialized coverages like the Brown, Gray, White and Black. The Blue and Purple coverages seem to be generic coverages that get added on most risks being covered while the other coverages seem to be more specific to the type of business and industry segment being insured.



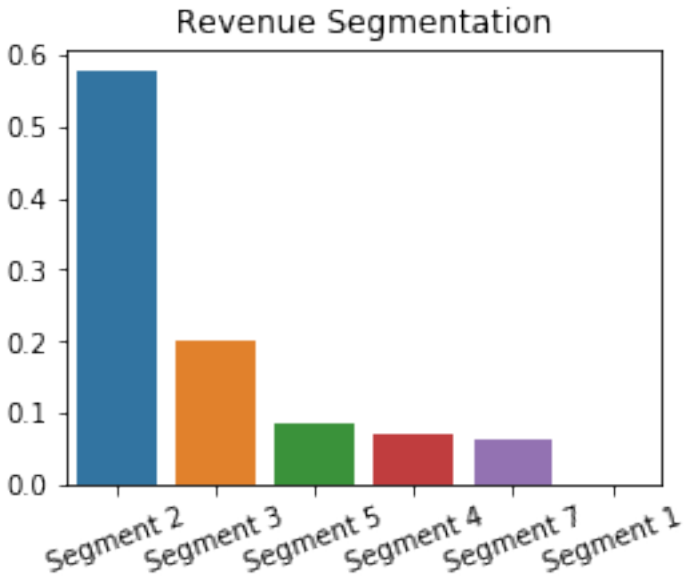
0.7.5 Product Line Distribution

In the visualization below, Product line distribution shows that the carrier offers a comprehensive product line designed specifically to cover the particular insurance needs of small and mid-sized businesses. The distribution seems to be evenly distributed across observations except for one specific category - ``Green-Yellow'' that encompasses more number of related products targeting a specific industry segment/business unit.



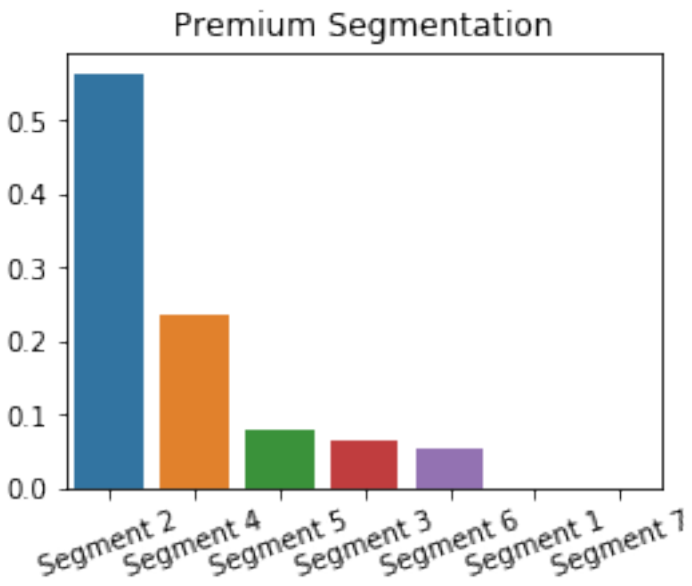
0.7.6 Revenue Segmentation Distribution

From the Revenue Segmentation Distribution visualization below, it is obvious that the Segment-2 is significantly higher than the other categories primarily because client's businesses associated with this category are small sized business. This segment categorization is not direct indication of premium being booked or profitability of the carrier but more about the revenue categories that the client's businesses fall under.



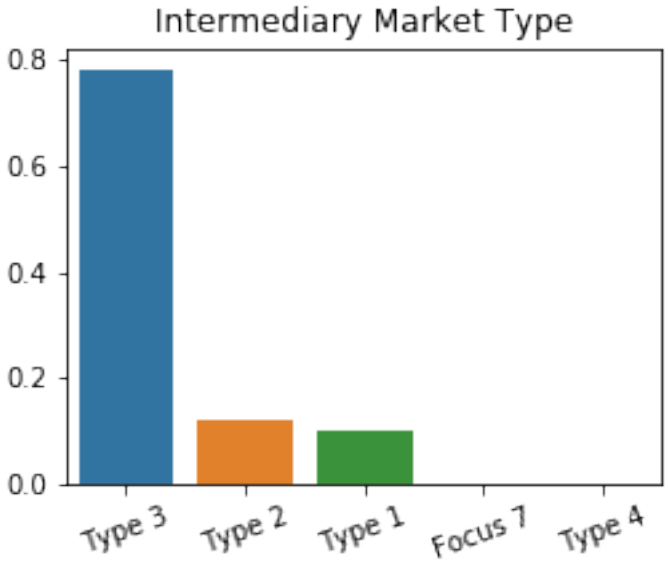
0.7.7 Premium Segmentation Distribution

The Premium Segmentation Distribution plot below indicates that majority of the insureds fall in the Segment-2 category which complements the earlier visualization of client's revenue segmentation. This again is not a representation of booked premium in the sense that it does not indicate that significant chunk of booked premium is from the Segment-2 category.



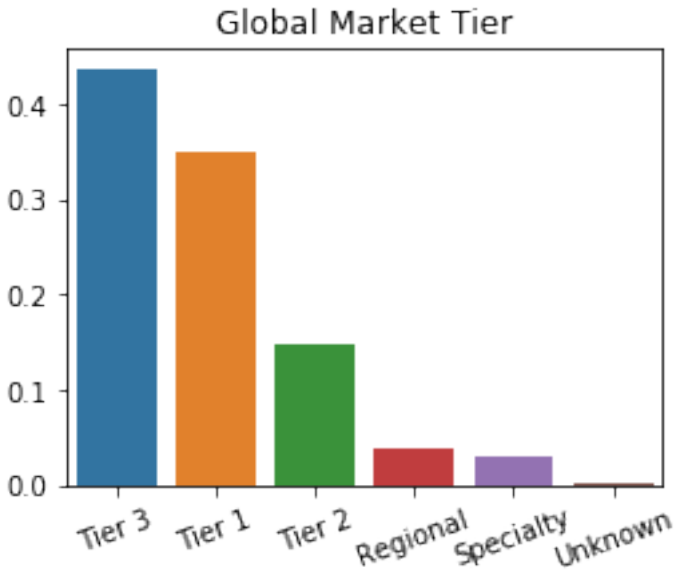
0.7.8 Intermediary Market Type Dist

The visualization below for Market Type Distribution indicates that majority of the observations fall under the Type-3 market category and this category could be indicative of the fact that the intermediaries are related to servicing policies and risks associated with small sized businesses. A more in depth look at the data would probably provide more information on the type businesses associated with this specific intermediary category. Type-2 and Type-1 intermediary categories are evenly distributed on relatively low number of observations and the other two categories are associated with very few observations.



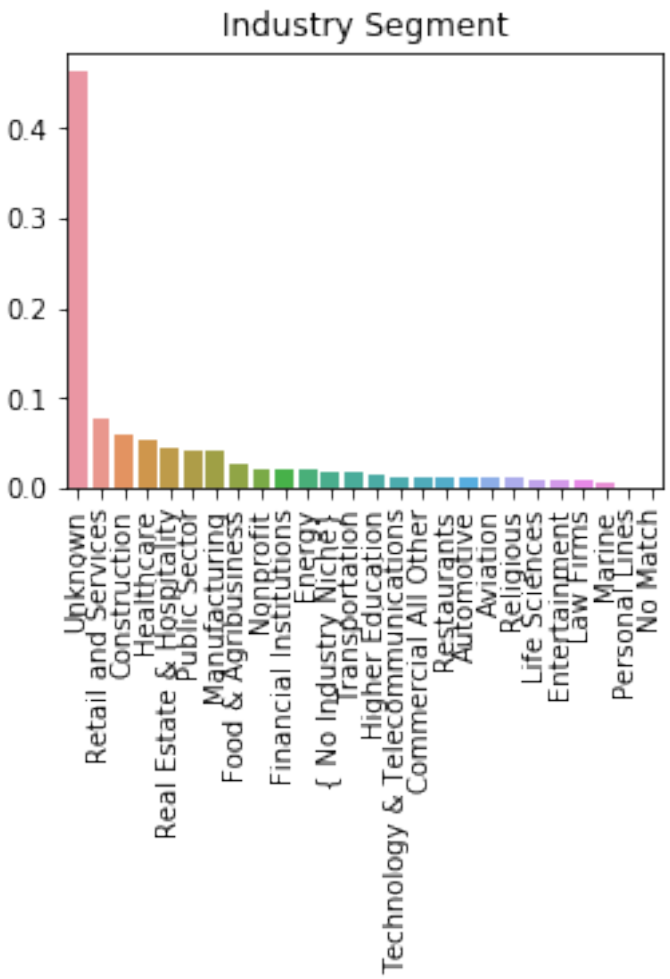
0.7.9 Global Market Tier Distribution

The Global Market Tier Distribution indicates that the most of the observations fall under Tier-3 or Tier-1 category. Tier-2 category is relatively low compared to the Tier-1 and Tier-3 categories. Only a few observations have this attribute categorized as Regional or Specialty. Tier-1 and Tier-3 both include small and mid-sized businesses and it would be very interesting to see how this particular attribute affects the model. The obsevations with unknown value for this attribute will be retained and included in the final pre-processed dataset.



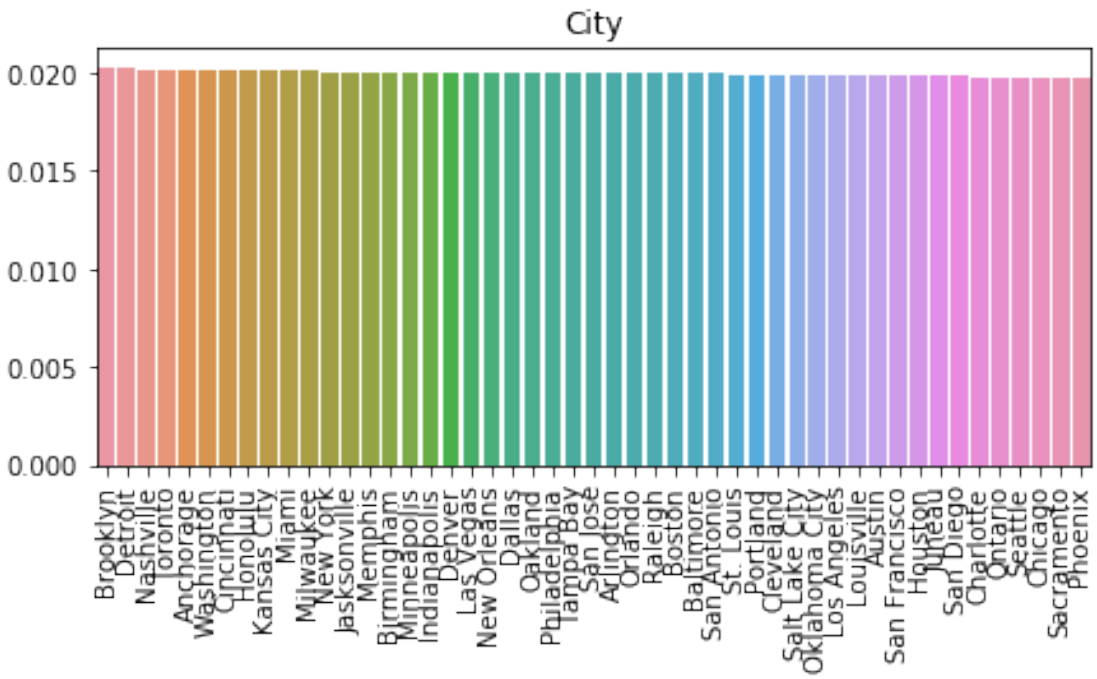
0.7.10 Industry Segment Distribution

The visualization below indicates that the Management Niche attribute in the dataset is not significant because of the number of observations that are marked as ``Unknown''. But, based on the initial analysis, industry segment appears in different form in the dataset. There are several attributes that offer insight into the client's industry segment namely sic_level_2, sic_level_2_name, sic_level_8, sic_level_8_name, management_niche, industrynicheconvention_desc, NAICSlevel1_Name, NAICSLevel1_Code. Most of the above features will be dropped except sic_level_8 and NAICSLevel1_Code that are related to industry segmentation and add value to model training and evaluation.



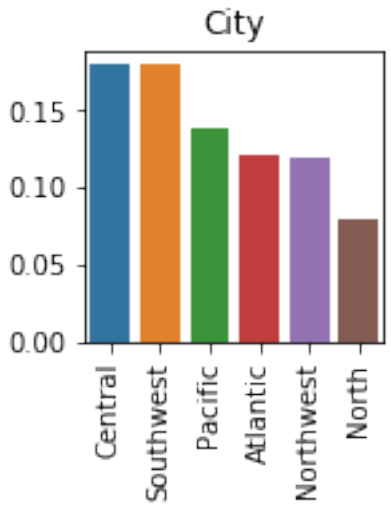
0.7.11 City Distribution

There are no anomalies in the City Distribution below, the observations are spread across different cities evenly and the distribution represents the nation wide presence of the carrier's products and uniformity in the underwriting service level across different cities. It also represents the fact that the insurer is preferred by different businesses for insuring certain types of risks and adopts a well defined pricing strategy across different states. This observation in combination with the market type, market tier and industry segment distribution indicates that the insurer has underwriting expertise in insuring Type-3 markets across the nation.



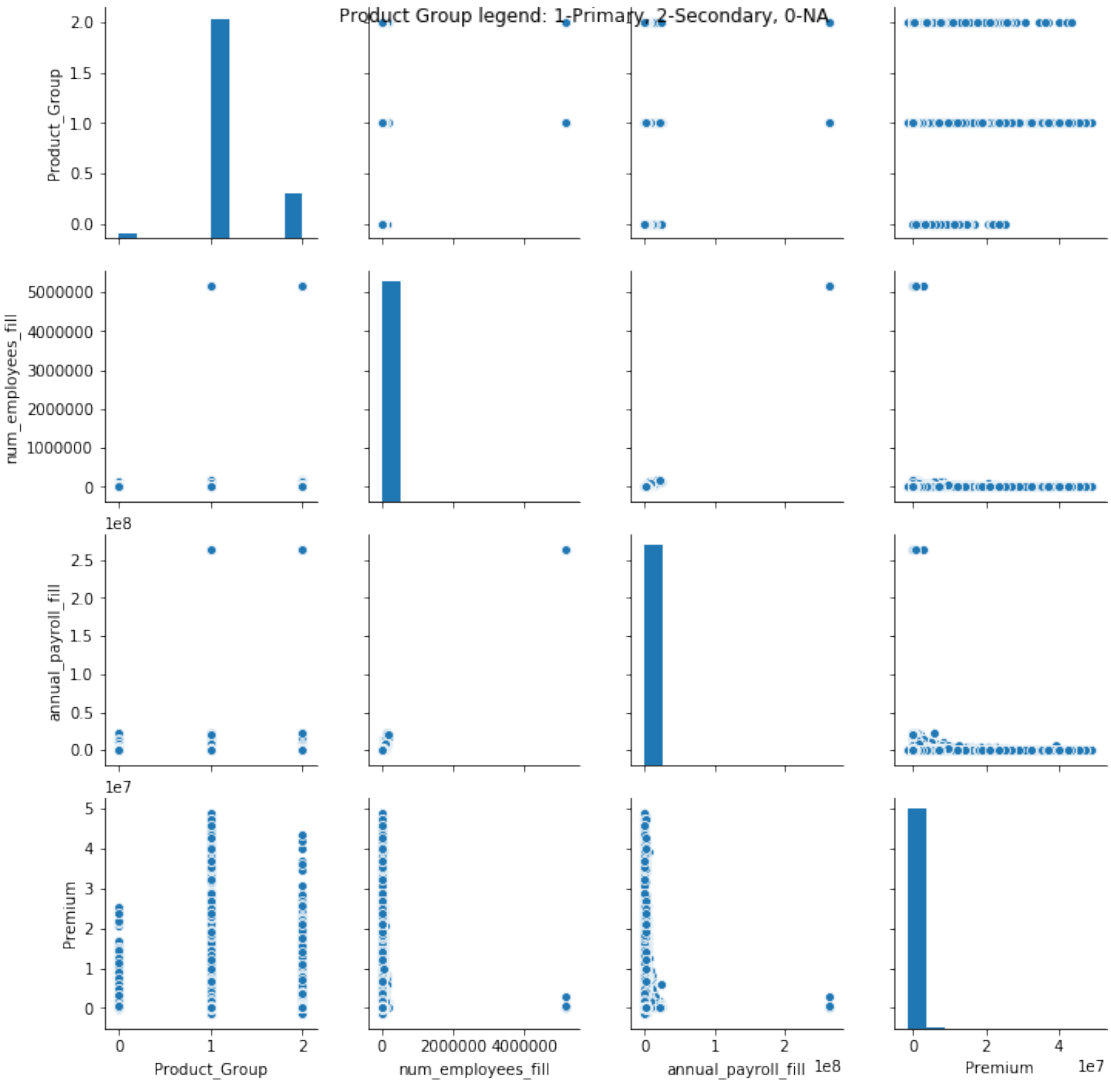
0.7.12 Region Distribution

As in the City distribution, there are no significant anomalies in the Region Distribution below, the observations are more or less spread evenly across different regions and the distribution represents the nation wide presence of the carrier's products. The Central and Southwest regions have a slightly higher significance compared to other regions because of the business units and market types that is targeted by the carrier. This distribution again goes to show that the carrier has gained underwriting reputation in insuring manufacturing businesses that potentially fall under the Type-3 market category or small sized businesses.



0.7.13 Bivariate Distribution - Employee Count, Annual Payroll, Premium vs Product Group

Below are bivariate relationship plots of employee count, annual payroll and premium plotted against product group to identify the degree and pattern of relationship between these variables. The attributes employee count, annual payroll and premium were chosen for this plot because these feature explain significant variability in the model and it would be interesting to understand how these features relate to the Product Group. The plots clearly indicate that there is a strong relationship between Primary product group and these key features. Significant number of observations fall under the Primary product group consistent with our earlier findings.



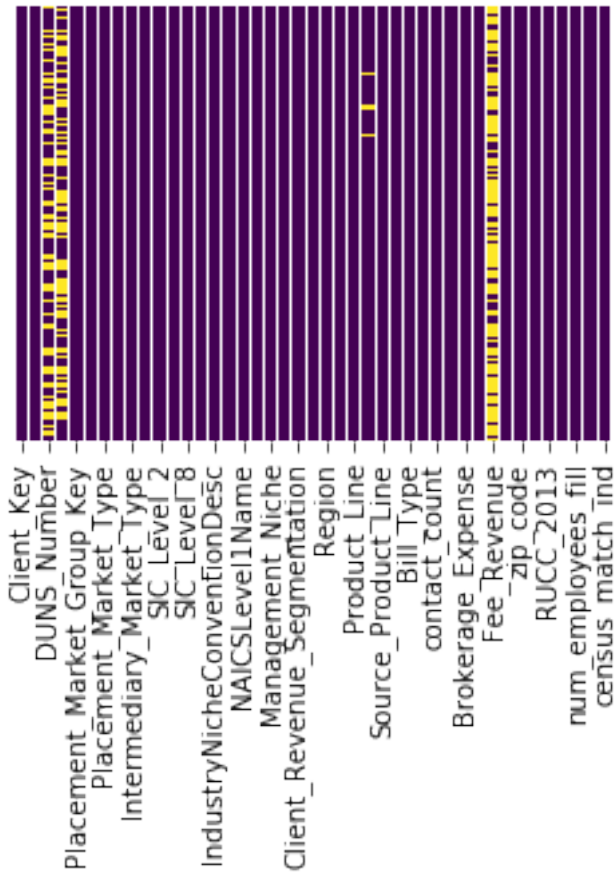
0.7.14 Null values Heatmap

From the visualization below, some of the predictor variables like Fee_Revenue, Client_Offices, DUNS_Number are missing values and the percentage of missing values ranges between 45-80%. The intuition is that these variables do not hold any statistical significance for the prediction problem and they should be dropped. There is no need for for imputing the missing values on these variables.

The target variable (Product Group) has a very small percentage of missing values and those observations can be dropped from the final dataset. The missing values on the target variable requires manual involvement and input from the business units to include them in the supervised learning process. There is a potential for these missing values to turn into a new class which may force the problem to be updated to a multiclass classification problem. For now, the observations with missing values on the target variable will be ignored and proceed with the

assumption that this is a binomial classification problem.

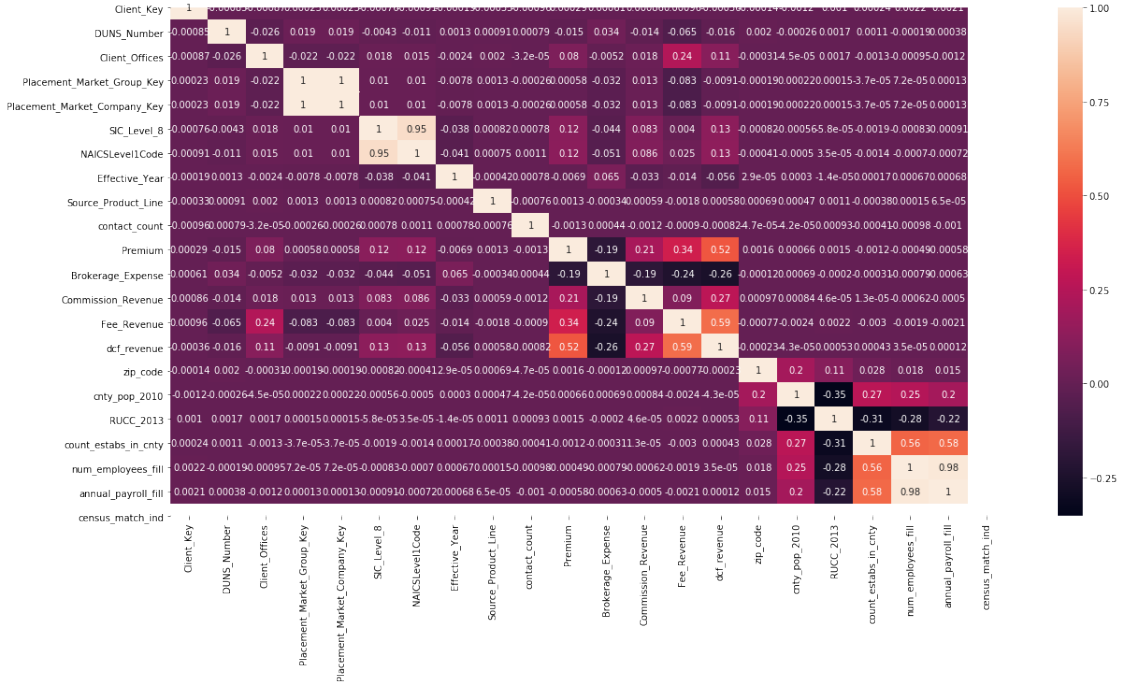
[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1a254f6f50>



0.7.15 Predictor Variables Correlation Heat Map

The correlation matrix below indicates that there are some variables with very strong positive and negative correlation which should technically be excluded from the model training - for example, zip code & cnty population, some of very high correlation features should be dropped during data pre-processing and others retained to determine the importance of those features. There is a potential risk of overfitting but the model could be re-trained after observing the importance of those high correlation features.

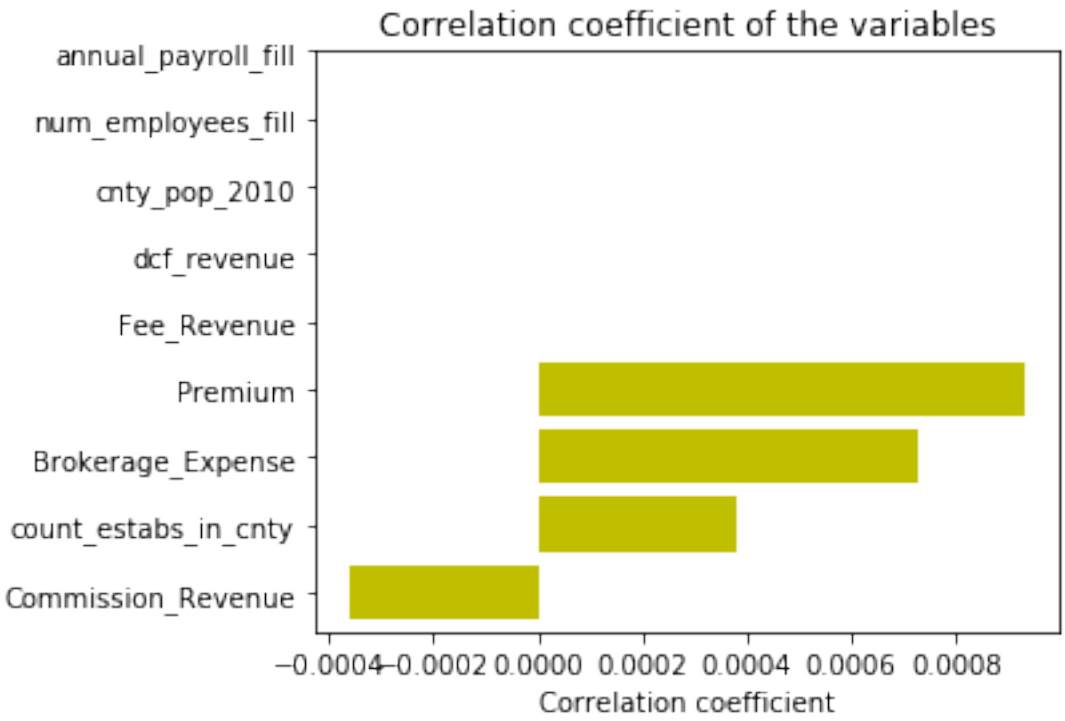
[19]: <matplotlib.axes._subplots.AxesSubplot at 0x1a25c9aed0>



0.7.16 Correlation of Numeric Attributes in Dataset

The below visualization represents the correlation between the numeric predictor variables in the dataset. As observed in the previous correlation heat map, there are certain numeric variables that exhibit very high postive and negative correlation which would significantly impact the model accuracy and they will be dropped from the model training and evaluation during data pre-processing phase.

['Premium', 'Brokerage_Expense', 'Commission_Revenue', 'Fee_Revenue', 'dcf_revenue', 'cnty_pop_2010', 'count_estabs_in_cnty', 'num_employees_fill', 'annual_payroll_fill']



0.7.17 Check Missing values

The table below indicates the ratio of missing values on the total dataset. The table lists only those attributes that have at least 1 missing value. From the list below - Fee Revenue, Client Offices and DUNS Number have the very high number of missing values and the ratio is insignificant on the remaining attributes. The attributes with large number of missing values have been determined during EDA are of low significance and will be dropped from model training and evaluation. The observations with missing values on the target variable will be dropped from the final pre-processed dataset. The other numeric variables that are missing values and part of the feature space will be imputed to the mean value.

[21]:

	columns	data_missing_ratio
34	Fee_Revenue	0.774495
3	Client_Offices	0.467044
2	DUNS_Number	0.461587
25	Product_Group	0.021196
37	cnty_pop_2010	0.002401
38	RUCC_2013	0.002401
1	Global_Market_Tier	0.001919
40	num_employees_fill	0.001345
41	annual_payroll_fill	0.001154
35	dcf_revenue	0.001005

4 Placement_Market_Group_Key 0.000002

0.7.18 Outlier Detection

The dataset contains 39 outlier records that have unrealistic values for number of employees, zip code, fee revenue and several cases missing values on several key features. These outlier records have the potential to skew the model and impact the accuracy. These outlier records will be dropped from the final dataset.

0.8 Data Pre-Processing and Feature Engineering

Data pre-processing is also referred to as the data preparation or data cleaning phase and in this phase unneeded features and outliers will be removed, records with missing values will be dropped, semantic adjustments will be made, categorical features will be transformed to numerical features and missing values will be imputed to generate a final target dataset that can be consumed by the model. This step will also ensure any inconsistencies in the data are corrected and new features are dervied if needed. The merged dataset (client_purchase_history.csv and client_census_data.csv) will be used in this phase and the above listed data pre-processing steps will be executed to produce the final target dataset.

0.8.1 Drop unneeded features

As stated and identified during the EDA phase, the following features will be dropped from the dataset because these features do not add any value to the model and some of the features have high correlation with other attributes and impact the independence assumption. The features that would be dropped are: Client_Key, Client_Key, DUNS_Number, Placement_Market_Group_Key, Placement_Market_Company_Key, SIC_Level_2, SIC_Level_2_Name, SIC_Level_8_Name, IndustryNicheConventionDesc, NAICSLevel1Name, Source_Product_Line, Brokerage_Expense, Commission_Revenue, Fee_Revenue, cnty_pop_2010, RUCC_2013, dcf_revenue, census_match_ind, PolicyID, Bill_Type, Source_System_Name, Management_Niche, Intermediary_Detail, Client_Offices

0.8.2 Semantic adjustments to features

The attribute Satellite in the dataset will be renamed to city because it represents the city in which the client operates the business that is being insured. Although this semantic adjustment does not have any relation to the model performance and accuracy, it is sometimes necessary to make such adjustments to improve readability and user understanding.

0.8.3 Drop outlier records

In this step, the outlier records identified with zip code value as 99999 will be dropped from the dataset because as discussed before, these observations skew the model significantly and affect the performance. The outlier records can also

force the model to overfit the data which will cause the model to perform poorly on the test dataset.

0.8.4 Impute missing value for Payroll & Employee count with mean

In this step, the missing values on the some of the essential features will be imputed to the mean of the values for those features. The operation will also make sure that the mean value is calculated after ignoring the null or NA values for features. This imputation strategy is the ideal imputation strategy for the identified numeric attributes because of their distribution across all observations.

0.8.5 Drop records with missing values

As a first sub-step records with missing values in Product Group and Global Market Tier attributes will be dropped. Product Group is the target variable and it cannot have missing values. If those observations need to be retained then they need to be labeled with help of business stakeholders before including them in the final dataset. As a second sub-step, the target variable will be encoded to numeric value for model training and evaluation.

0.8.6 Convert features to string for Categorical Encoding

In this step, categorical attributes including global_market_tier, placement_market_type, placement_market_focus, intermediary_market_type, client_premium_segmentation, client_revenue_segmentation, city, region, product_coverage, product_line should be transformed using an encoding strategy before they can be included in the model training and evaluation. There is a element of risk in encoding these features because of very high cardinality. As part of this research, a label encoder has been used to transform the features and the transformed features will be retained in the final dataset for training and evaluation.

0.9 Model Experimentation using Sub-Sampled Data

This is a preliminary step to understand how the different modeling techniques compare using a sub-sampled dataset from the final pre-processed dataset obtained from the data pre-processing phase before conducting training and evaluation on the full dataset. Different supervised learning models will be explored using basic parameter setting without any fine tuning and based on the outcome of this preliminary experimentation, the hyper parameter settings would be refined for optimal performance on the full dataset.

0.9.1 Split the dataset

As a first step, the numeric encoded target variable will be isolated from the pre-processed final dataset and dataset will be split randomly using a well known dataset split function in Python (train_test_split)

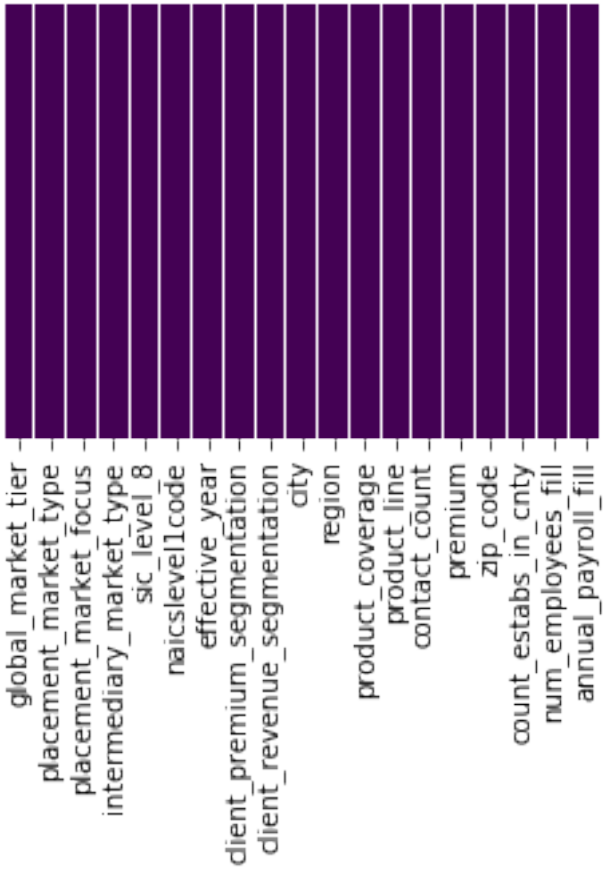
0.9.2 Create Sub Sample for Model Comparison

In this step, the randomly split train and test dataset will be sub-sampled at the rate of 0.2 and stratified to ensure that there is more or less the same distribution of target classes in the split datasets.

0.9.3 Heatmap - No Nulls

This step is a re-plot of the missing values heat map to ensure there are no missing values in the final split datasets to be used in the model exploration process. This step may look unnecessary but it is a good modeling practice to ensure all missing values are accounted for before conducting training and evaluation to avoid any unintended results.

[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1a25636a50>

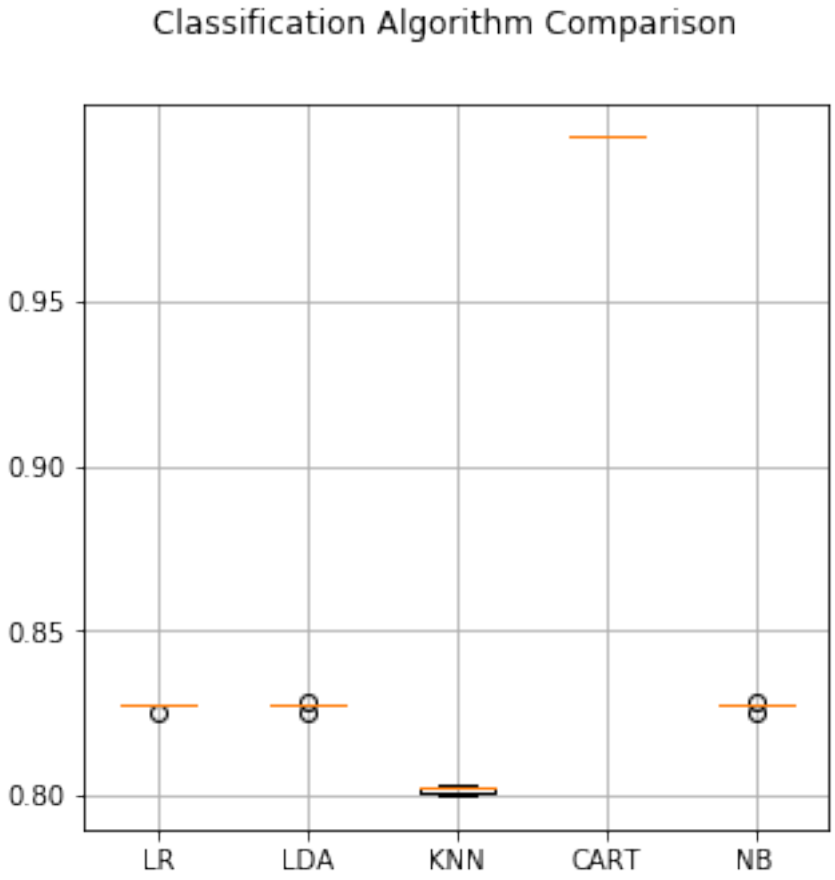


0.9.4 Model Comparison

Multiple supervised learning techniques/algorithms will be explored in this step using the sub-sampled dataset including Decision Tree, LDA, Gaussian NB, KNN, Logistic Regression with basic parameter setting and cross validation will be

employed to identify the best estimator that produces the best accuracy score. The results of cross validation on trained models using the sub sampled dataset indicate that the tree based method (Decision Tree Classifier) performs better than the other modeling techniques. In the full dataset model exploration, an ensemble method will also be explored and compared with the results of other modeling techniques. SVM classifier although could produce the same level accuracy as the tree based method may not be ideal for Production setting due to the performance issues associated with it on training large datasets.

```
LR: 0.827392 (0.000953)
LDA: 0.827475 (0.001015)
KNN: 0.801753 (0.001421)
CART: 1.000000 (0.000000)
NB: 0.827475 (0.001015)
```



0.10 Model Training & Evaluation using Pre-Processed Data

The final cleaned up pre-processed dataset contains 19 features and a target variable. This step encompasses a model preparation pipeline that transforms categorical to numerical data and scales the data as the initial step and fits

the pre-processed data on this pipeline. The supervised learning modeling techniques discussed in the previous section will be trained and evaluated on the full dataset too and in addition one ensemble method (Random Forest would be explored) and one clustering method (KMeans) will be explored. This exploration on the final pre-processed dataset will help in deciding the ideal modeling technique for Production implementation both in terms of model accuracy and performance. The research also explored the dataset as it is with the original dimensions/features and reduced dimensions using Factor Analysis. The results of the experimentation and training & evaluation are discussed in the ``Conclusions'' section.

Factor analysis is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. It is possible that variations in observed variables mainly reflect the variations in unobserved (underlying) variables. Factor analysis searches for such joint variations in response to unobserved latent variables. The observed variables are modelled as linear combinations of the potential factors, plus ``error'' terms. Factor analysis aims to find independent latent variables. (Wikipedia, 2019).

0.10.1 Supervised Learning Methods Exploration

Split the data using Sklearn As a first step, the numeric encoded target variable will be isolated from the pre-processed final dataset and dataset will be split randomly using a well known dataset split function in Python (train_test_split) into train and test. The split training dataset will be further split into 80% training set and 20% validation set that will be used for model training and evaluation.

Train Decision Tree Classifier According to (Pedregosa et al, 2011), Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of the target variable by learning simple decision rules inferred from the data features. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and fitter the model. As with other classifiers, DecisionTreeClassifier takes as input two arrays: an array X, sparse or dense, of size [n_samples, n_features] holding the training samples, and an array Y of integer values, size [n_samples], holding the class labels for the training samples. The result of training and evaluation of the DT model on train and test data is shown below in a table. The accuracy outcome and performance will be discussed in detail in the ``Conclusions'' section.

[77]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
```

```
                           random_state=None, splitter='best')
```

[79]:
```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

Train Gaussian NB Classifier According to (Pedregosa et al, 2011), Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes' theorem with the ``naive'' assumption of conditional independence between every pair of features given the value of the class variable. GaussianNB implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian and the parameters (mean, sd) are estimated using maximum likelihood. The result of training and evaluation of the Gaussian NB model on train and test data is shown below in a table. The accuracy outcome and performance will be discussed in detail in the ``Conclusions'' section.

[81]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

[83]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

Train KNN Classifier According to (Pedregosa et al, 2011), Neighbors-based classification is a type of instance-based learning or non-generalizing learning it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. The neighbors classification in KNeighborsClassifier is a most commonly used technique. The optimal choice of the value is highly data-dependent: in general a larger k-value suppresses the effects of noise, but makes the classification boundaries less distinct. The result of training and evaluation of the KNN model on train and test data is shown below in a table. The accuracy outcome and performance will be discussed in detail in the ``Conclusions'' section.

[85]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

[87]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

Train Random Forest Classifier According to (Pedregosa et al, 2011), there are two ensemble based averaging algorithms based on randomized decision trees:

the RandomForest algorithm and the Extra-Trees method. Both algorithms are perturb-and-combine techniques specifically designed for trees. This means a diverse set of classifiers is created by introducing randomness in the classifier construction. The prediction of the ensemble is given as the averaged prediction of the individual classifiers. As other classifiers, forest classifiers have to be fitted with two arrays: a sparse or dense array X of size [n_samples, n_features] holding the training samples, and an array Y of size [n_samples] holding the target values (class labels) for the training sample. The result of training and evaluation of the Random Forest model on train and test data is shown below in a table. The accuracy outcome and performance will be discussed in detail in the ``Conclusions'' section.

[89]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

[91]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

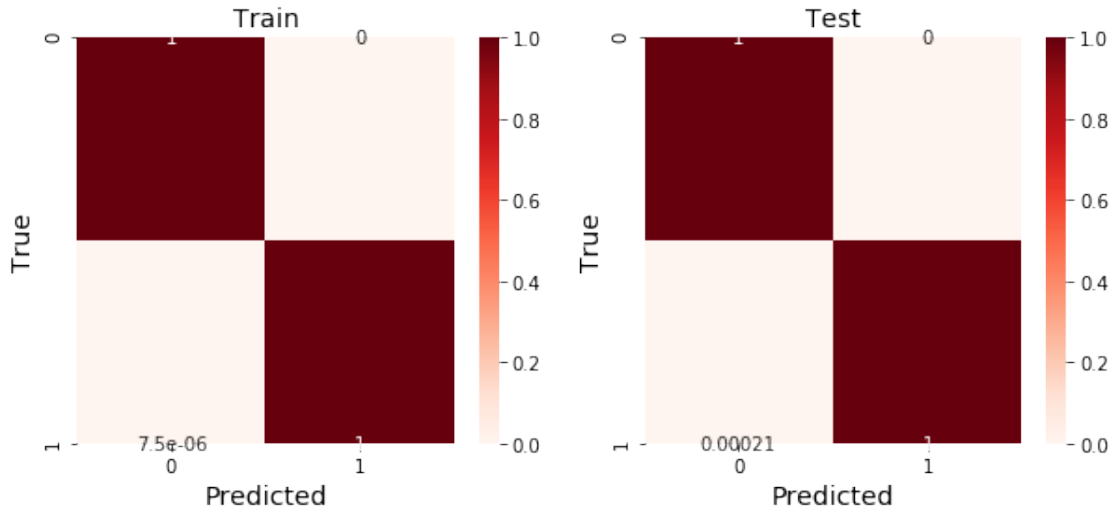
Confusion Matrix for Random Forest Tree Classifier (one example) The confusion matrix is a useful method to describe the performance of a classification model on a set of test data for which true values are known. Below is a sample observation of confusion matrix for the random forest classifier model. The following information can be extracted from a confusion matrix.

True Positive (TP) : Model correctly predicted Positive cases as Positive.

False Positive (FP): Model incorrectly predicted Negative cases as Positive.

False Negative (FN): Model incorrectly predicted Positive cases as Negative.

True Negative (TN) : Model correctly predicted Negative cases as Negative.



Train LR Classifier According to (Pedregosa et al, 2011), Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function. The sklearn implementation of logistic regression can fit binary, One-vs-Rest, or multinomial logistic regression with optional Ridge, Lasso or Elastic-Net regularization. The result of training and evaluation of the Random Forest model on train and test data is shown below in a table. The accuracy outcome and performance will be discussed in detail in the ``Conclusions'' section.

[96]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

[98]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

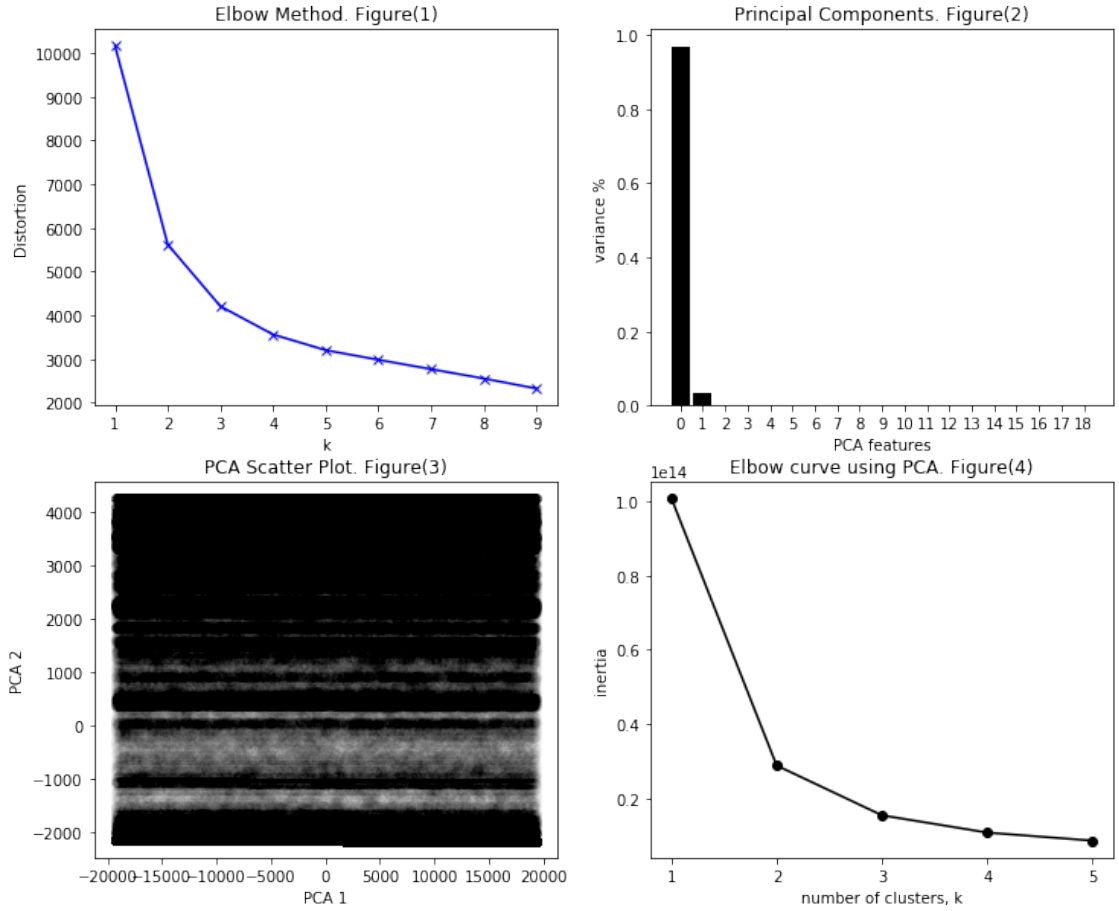
0.10.2 Unsupervised Learning - KMeans Clustering Exploration

According to (Pedregosa et al, 2011), the KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to large number

of samples and has been used across a large range of application areas in many different fields. The k-means algorithm divides a set of N samples X into K disjoint clusters , each described by the mean of the samples in the cluster. The means are commonly called the cluster ``centroids''; note that they are not, in general, points from , although they live in the same space. The Kmeans clustering method seems to exhibit very poor accuracy for our research question hovering around 50% with and without using PCA on different variants of the dataset and may not be the ideal model for implementation.

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variable. (Wikipedia, 2019).

The first elbow curve below (Figure-1) shows that the optimal number of clusters that can be obtained from the pre-processed dataset with 19 feature variables is 2 or 3. Through this research, the dimension of the dataset was reduced using PCA to understand the cumulative variability of the dataset explained by the principal components and the number of components that explained more than 90% of variability were used in the model training and evaluation using principal components. Based on the plot below it looks like the first two principal components explain more than 90% of variability (Figure-2). The elbow curve of the PCA transformed dataset shows that the optimal number of clusters that can be obtained from the dataset is 2 (Figure-4).



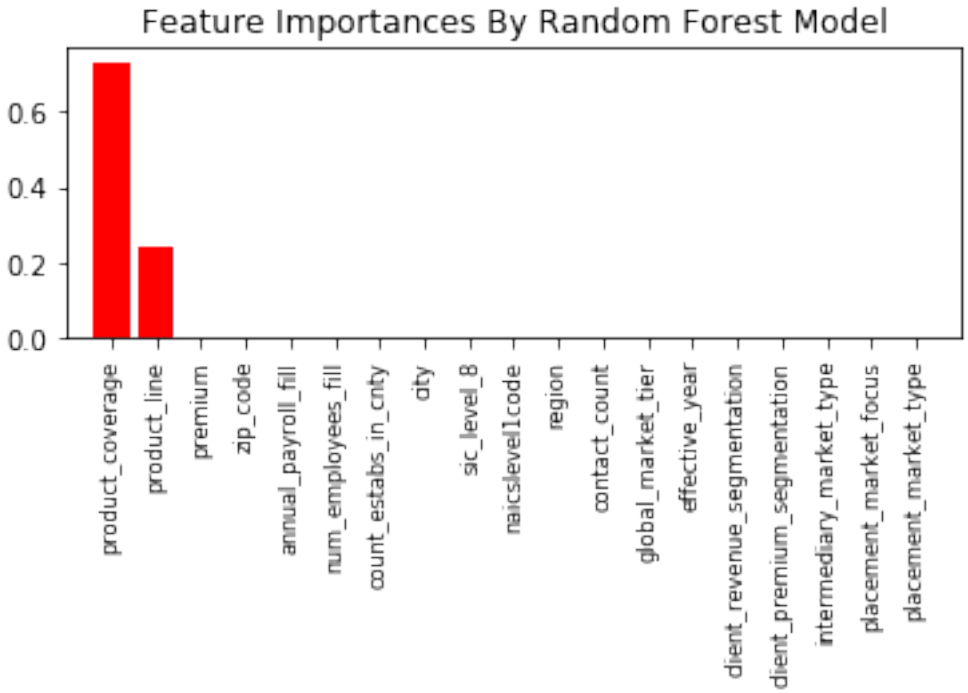
[70]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

[73]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

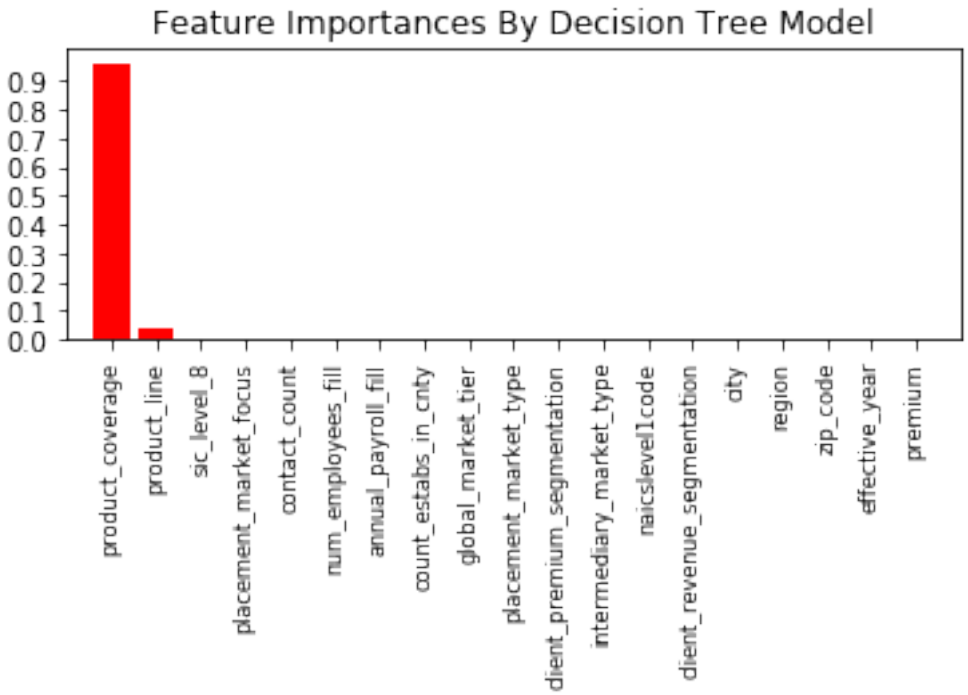
0.11 Feature Importance

The below plots show the importance of features on the tree based modeling methods used on our pre-processed dataset. The red bars are the feature importances of the tree based methods and the plot shows the number of features that are informative compared to other features and how they contributed to the model accuracy. Although the Product Coverage and Product Line attributes are shown as the most important features in both the methods, there are other features/attributes of low importance and are different between the tree based methods which is an interesting observation.

0.11.1 Random Forest Classifier



0.11.2 Decision Tree Classifier



0.12 Conclusions

Model Pipeline

To answer the research question, multiple models with different algorithms and parameter setting were built without any major hyper parameter tuning. To train and evaluate the models, a pre-processed dataset with an optimal set of features was required. To accomplish this, different pre-processing methods and feature selection techniques were employed to identify a final set of 20 features. The objective of the research as the data, pre-processing & feature extraction techniques were explored was to conduct multiple experiments using different algorithms to build an optimized model that predicts with high accuracy. We explored different dimensionality reduction techniques on both supervised and unsupervised methods and observed that the original set of features produced better accuracy than the reduced dimensions. It is necessary for any model that will be operationalized to explore different hyper parameter setting for all the different models including the ones that came up with a low accuracy score to achieve high business value. Also exploring the hyper parameter space for all the models would help understand deeply on what features are impacting the accuracy and offer valuable learnings for future implementations.

Pre-processing and Feature Engineering

The research also explored the possibility of deriving new features/encoded features and visually explored the importance of all the features in certain models that were built. It is a essential data preparation activity to identify or derive new features and include them in the experimentation specifically in cases where the raw data is not directly usable. These derived features can significantly complement the model and positively impact the model performance but a right balance should be maintained because overdoing the feature engineering process may result in overfitting and skew the model. The research conducted confirms the understanding that model processing improved and performed better through pre-processing techniques. Excluding features from the model training that were least important was of high significance, feature importance information can be further evaluated to even further reduce the feature space and will be of great help during operationalization. Also, imputation strategies were employed, unneeded features were dropped, numeric data was scaled to ensure uniform variance, data type conversions and semantic adjustments were made, highly correlated features were excluded, outliers were eliminated and encoding strategy was used to transform categorical attributes. All of these data pre-processing and feature engineering steps had a significant positive impact on the model training and evaluation and helped in developing an optimized model with only essential features.

Outcome of Experiments

As described in the model pipeline section for the problem setting, multiple models were explored including one clustering method (KMeans), Decision Tree,

Random Forest, LDA, Gaussian NB, KNN, Logistic Regression with different parameter setting and cross validation was used to identify the best estimator that produced the best accuracy score. During model exploration, there was an attempt to fit support vector machine model, but it resulted in poor performance and will not be an ideal solution for the problem that we are trying to solve. The KNN classifier although produced a reasonable accuracy will not be an ideal candidate for operationalization because of the concern on performance on larger datasets in case of need to tune the model further using historical data in the future. KMeans clustering method produced the lowest accuracy of all the models that were explored and will not be the ideal candidate for Production implementation for this specific research questions. Tree/Ensemble methods seem to be recommended models for real time implementation but there is also a need to understand the difference in performance if more observations are included in the training set. The experimental results for each of the algorithms that were explored can be found below.

Model Algorithms

Following are the different algorithms that were explored for the research question. The next logical to this research would be to build home grown optimizers as a future extension to this research.

1. KNN Classifier
2. Gaussian NB
3. Random Forest Classifier
4. Decision Tree Classifier
5. LDA
6. SVC (Support Vector Machine)
7. Logistic Regression Classifier
8. KMeans Clustering

Results Table

[105]:

Supervised

	Model description	Train Accuracy Score	Test Accuracy Score	Train Pred Time	Test Pred Time
0	SKLearn Decision Tree Classifier Model	1.000000	1.000000	0.072768	0.027038
1	SKLearn Decision Tree Classifier Model using FA	0.987761	0.657676	0.834687	0.076382
2	SKLearn Gaussian NB Classifier Model	0.827706	0.826547	0.278197	0.069566
3	SKLearn Gaussian NB Classifier Model using FA	0.827706	0.826547	0.123843	0.030935
4	SKLearn KNN Classifier Model	0.842978	0.800670	53.222334	14.713638
5	SKLearn KNN Classifier Model using FA	0.836987	0.802507	24.542491	7.393853
6	SKLearn Random Forest Classifier Model	0.999999	0.999964	0.709367	0.191311
7	SKLearn Random Forest Classifier Model using FA	0.902171	0.803306	3.962626	0.579339
9	SKLearn LR Classifier Model	0.827695	0.826541	0.034400	0.010244
10	SKLearn LR Classifier Model using FA	0.827706	0.826547	0.016523	0.005841

[107]: **Unsupervised**

	Model description	Train Accuracy Score	Test Accuracy Score	Silhouette Score	Train Pred Time	Test Pred Time
0	KMeans Clustering method using PCA	0.499875	0.501022	0.544758	0.076584	0.017936
1	KMeans Clustering method	0.500330	0.506555	0.576710	0.128574	0.042658

0.12.1 References

1. American Institute For Chartered Property Casualty Underwriters (2019). Fundamentals of Underwriting. The Institutes. Retrieved from https://www.theinstitutes.org/doc/resources
2. Chester, A., Ebert,S., Kauderer,S. and McNeill,C. (2019, Feb). From art to science: The future of underwriting in commercial P&C insurance. Mckinsey & Company. Retrieved from https://www.mckinsey.com/industries/financial-services/our-insights/from-art-to-science-th
3. Berger,L., Cummins,D., Danzon,P., Doherty,N., Garven,J., Harrington,S., Klein,R., McDonald,J., Roddis,R., Stewart,B., Stewart,R., (1991). Cycles and Crises in Property/Casualty Insurance: Causes and Implications for Public Policy. Retrieved from https://www.naic.org/documents/. p.1-420. Kansas City, MO:NAIC Publication No. 71
4. Pedregosa et al. (2011), Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830
5. Boodhun, Noorhannah & Jayabalan, Manoj. (2018). Risk Prediction in Life Insurance Industry using Supervised Learning Algorithms. Complex & Intelligent Systems. 10.1007/s40747-018-0072-1.
6. Wikipedia (2019). Factor Analysis. Retrieved from https://en.wikipedia.org/wiki/Factor_an
7. Wikipedia (2019). Principal Component Analysis. Retrieved from https://en.wikipedia.org/wiki/Principal_component_analysis