Machine Learning for Autonomous Code Classification in Banking

Steve Norris

Loyola University Maryland

**Abstract**

**Objective** This report demonstrates the application of machine learning for Code classification in the loan origination process, with estimates on budgetary impact.

**Methods** Non-identifying data on historical and current loans were pooled from several data warehouses to recreate a bank’s full perspective on the individual collateral entities on record that contribute to total reported assets. The integrity of these queries was verified over several iterations across knowledgeable lines of business (LOB) where no standard practice exists. The features of this resulting dataset, of nominal and numeric types, were used to train various machine learning algorithms that seek to identify patterns and develop predictive capabilities to accurately classify new incoming data by target variable ‘Collateral Code’.

**Results** Three supervised and one unsupervised classifier were trained on the dataset; Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and K-Modes (KM). Each classifier was validated three ways – train-test validation, Leave-One-Out, and K-Fold Cross Validation. LR and RF performed similarly well, with accuracy and precision scores in ranges 40-50% and 30-55% respectively. SVM consistently exhibited the weakest performance, with respective scores around 25% and 12%, dependent on randomness. Between LR and RF, five Collateral Codes were consistently predicted with high confidence. These Codes represent 40% of all Codes assigned, indicating value as a supplement to human effort if these algorithms become a supplement to the loan origination process.

The unsupervised model (KM) offers a different approach to the problem at-hand, identifying clusters within the dataset that likely belong to the same Collateral Code without prior domain knowledge of Codes. From this perspective, a revision of the Codes themselves may be desirable, potentially easing the Code assignment process, whether performed manually or by trained algorithm. Results from this model on the current data yield X clusters identified by a unique combination of Y features. Considering the origin and state of Collateral Codes, this approach is viable.

**Conclusion** Both supervised and unsupervised machine learning frameworks demonstrate tangible potential in the classification of new pieces of collateral recorded on a loan. Supervised machine learning algorithm Random Forest exhibits the strongest performance with the data provided, followed closely by Logistic Regression, both of which may be used to classify five Codes autonomously, alleviating human effort by 40%. Results of K-Modes open the conversation to refining the list of Collateral Codes used, further easing the classification process whether by human hand or algorithm.

Table of Contents

[1 Introduction 6](#_Toc45006676)

[1.1 Motivation 6](#_Toc45006677)

[1.2 Objective 6](#_Toc45006678)

[2 Methods 7](#_Toc45006679)

[2.1 Plan of analysis 7](#_Toc45006680)

[*2.2* Data origination 8](#_Toc45006681)

[2.3 Obstacles 9](#_Toc45006682)

[2.4 Developing the Sample Dataset 9](#_Toc45006683)

[2.5 Preprocessing 12](#_Toc45006684)

[2.5.1 Numeric Features 13](#_Toc45006685)

[2.5.2 Nominal Features 16](#_Toc45006686)

[2.5.3 Encoding Nominal Features 21](#_Toc45006687)

[2.5.4 Imputation of ‘Division’ 22](#_Toc45006688)

[2.5.4.1 ‘Interest Rate’ As Discrete 22](#_Toc45006689)

[2.5.4.2 ‘Interest Rate’ As Continuous 25](#_Toc45006690)

[2.5.5 Target Variable ‘Revised Collateral Code’ 28](#_Toc45006691)

[3 Results 28](#_Toc45006692)

[3.1 Supervised classification 28](#_Toc45006693)

[3.1.1 Logistic Regression 29](#_Toc45006694)

[3.1.2 Support Vector Machine 31](#_Toc45006695)

[3.1.3 Random Forest 33](#_Toc45006696)

[3.2 Unsupervised classification 36](#_Toc45006697)

[3.2.1 K-Modes 36](#_Toc45006698)

[4 Discussion 39](#_Toc45006699)

[4.1 Model Evaluation 39](#_Toc45006700)

[4.2 Threats to validity 39](#_Toc45006701)

[5 Future work 40](#_Toc45006702)

[5.1 Next steps 40](#_Toc45006703)

[5.2 Outstanding goals 40](#_Toc45006704)

[5.3 Potential implications 40](#_Toc45006705)

[6 Reflection 41](#_Toc45006706)

[6.1 List of Collateral Codes 42](#_Toc45006707)

[7 References 43](#_Toc45006708)

**List of Tables**

|  |  |  |
| --- | --- | --- |
| 1 | Collateral Code Screenshot | 12 |
| 2 | Number of Loans Recommending Changes By Industry | 14 |
| 3 | Outliers in Numeric Data | 17 |
| 4 | Effects of One Hot Encoding | 24 |
| 5 | Classification Report, ‘Interest Rate’ as Discrete | 26 |
| 6 | Classification Report, ‘Interest Rate’ as Continuous | 28 |
| 7 | Best Accuracy by Code, Logistic Regression | 33 |
| 8 | Best Accuracy by Code, SVM | 35 |
| 9 | Best Accuracy by Code, Random Forest | 37 |
| 10 | K-Modes Cluster Definitions | 39 |
| 11 | K-Modes Frequency of Features | 42 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| 1 | Collateral Code Process Flow | 10 |
| 2 | List of Features, Sample Dataset | 13 |
| 3 | Sample Dataset, Preliminary View | 15 |
| 4 | Correlation Heat Map | 16 |
| 5 | Numeric Data Density Plots | 17 |
| 6 | Density Plots, Log Transformed | 18 |
| 7 | Density Plots, Standardized | 18 |
| 8 | Level Proportions per Categorical Feature | 19 |
| 9 | Missing Data | 22 |
| 10 | Revised Collateral Code NULL Landscape | 22 |
| 11 | Missing Data by Feature | 23 |
| 12 | Distribution of Feature ‘Division’ | 23 |
| 13 | Accuracy by Choice of K Neighbors, ‘Interest Rate’ as Discrete | 25 |
| 14 | Confusion Matrix, ‘Interest Rate’ as Discrete | 26 |
| 15 | Distribution of Feature ‘Division’ Predictions, ‘Interest Rate’ as Discrete | 27 |
| 16 | Accuracy by Choice of K Neighbors, ‘Interest Rate’ as Continuous | 28 |
| 17 | Confusion Matrix, ‘Interest Rate’ as Continuous | 29 |
| 18 | Distribution of Feature ‘Division’ Predictions, ‘Interest Rate’ as Continuous | 30 |
| 19 | Level Frequencies of Target Variable, ‘Revised Collateral Code’ | 31 |
| 20 | Classification Scores by K Folds, Logistic Regression | 33 |
| 21 | Confusion Matrix, Logistic Regression | 33 |
| 22 | Classification Scores by K Folds, SVM | 35 |
| 23 | Confusion Matrix, SVM | 35 |
| 24 | Accuracy by Tree Depth | 36 |
| 25 | Classification Scores by K Folds, Random Forest | 37 |
| 26 | Confusion Matrix, Random Forest | 37 |
| 27 | Comparison of Supervised Models | 38 |

# Introduction

## Motivation

Fulton Financial Corporation (FFC) was founded in 1882 as a small bank headquartered in Lancaster, PA and focused on servicing agricultural and business accounts in addition to personal banking. Acquisition of other banks began in 1948, growing FFC into the largest bank in Lancaster County today. This expansion provided resources that have allowed FFC to reinvest in the local community and restore neglected properties and land.

Acquisition of 30+ banks’ accounts and records has also meant the adoption of as many different sources and standards of data modeling. Consequently and in recent years, FFC is shifting its internal focus to the cleaning and standardization of these data across warehouses and flow structures.

This research began with a desire to understand one critical data element, Collateral Codes, from the perspective of Commercial Real Estate (CRE). Stakeholders in this research were identified through discussions with involved LOBs and are documented here. Preliminary study focused on illuminating the current state and health of these Codes, analyzing their efficacy as markers for dollar amount value, and documenting the loan origination process from which these Codes are born into FFC’s data warehouses.

Collateral Codes are categorical indicators that signify the type of collateral put forth to secure a loan. FFC utilizes a list of approximately 100 Codes, which are used in calculating the risk of a loan as well as FFC’s estimated assets at any time. However, these Codes may be edited up to seven times during the loan origination process, with no explanation or authentication required. Further, there is currently no formal training available to any of these LOBs on how to assign Collateral Codes. Such unreliability of this Code holds significant potential financial liability for FFC, and the exploration of machine learning for automation of Code assignment is requested.

This research has been propelled primarily by an admiration for the interactions of foundational mathematical principles with dauntless creativity and vision that are leading the world of artificial intelligence and machine learning. That arguably all human decisioning can be simulated with mathematical models is at once exhilarating and humbling, making a future that is more robust to human error tangible.

## Objective

Apply tools and methods of statistical analysis and machine learning algorithms to a sample dataset. Identify patterns in the dataset that may leverage effective autonomous assignment of Collateral Codes to live data, with the goal of supporting or replacing manual entry of this critical data element. FFC estimates significant value from this work, both in man-hours saved and the elevated reliability of calculating total assets to stakeholders. This work may also inform best practices of data collection on a broad scale across the business.

# Methods

## Plan of analysis

Data on individual loan entities exists between several data warehouses, accessible via Oracle SQL Developer (alternatively KnowledgeShare, soon deprecated). These data sources will be joined on three keys to generate a unique identifier for each collateral entity; Bank\_Number, Account\_Number, and Appl\_ID. Features extracted for analysis include:

|  |  |  |
| --- | --- | --- |
| * Purpose Code * Collateral Code * Property Type * Net Operating Income | * Annual Debt Service * Original Balance * Unused Commitment * Interest Rate | * Guarantor? * Risk Code * Relationship ID |

SQL queries will be called on FFC’s data warehouses via Jupyter Notebook (kernel: Python 3), and will be saved in .csv format once validity of the data is verified by informed lines of business. If necessary, data may be loaded into R Studio should speed become a significant issue.

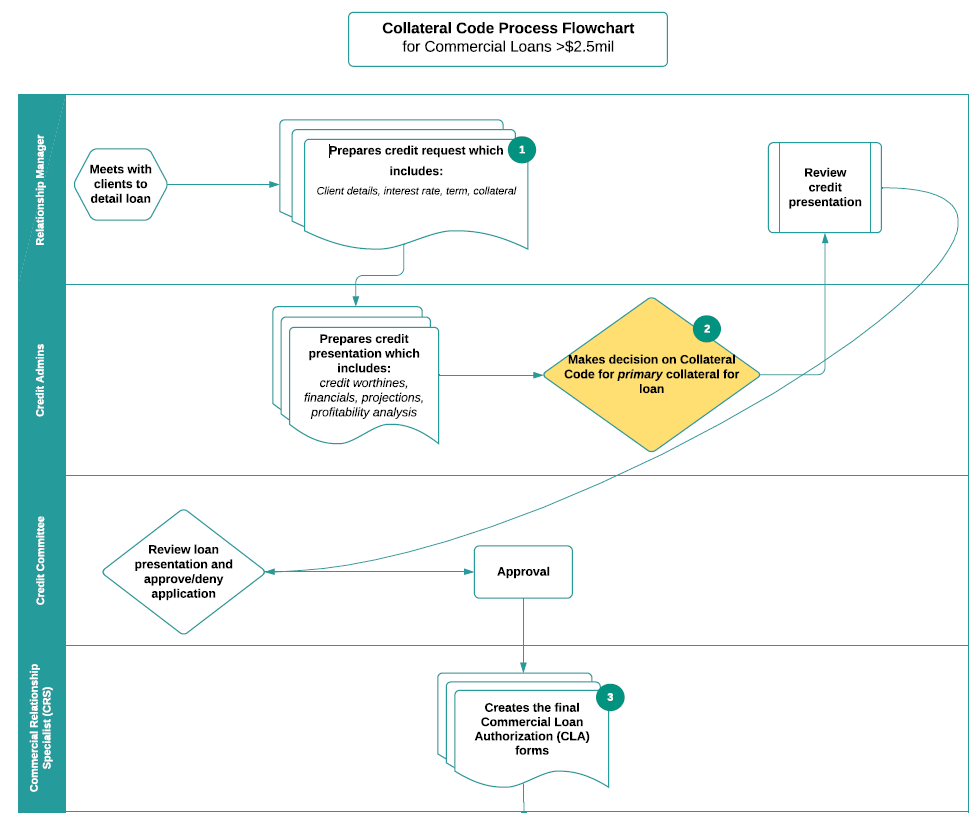
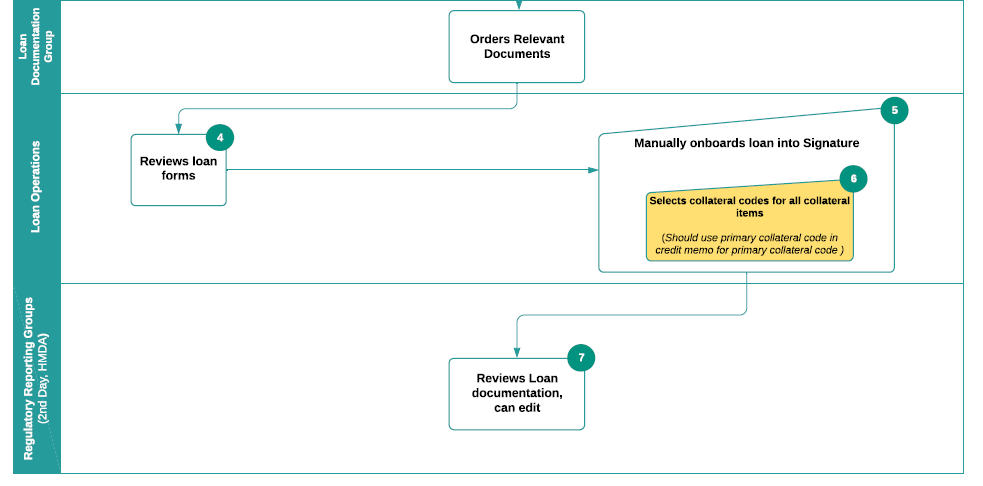
With Python as the language of focus, libraries utilized in this research will include pandas, numpy, matplotlib, seaborn, and scikit-learn (sklearn).

Following preprocessing of the data, a set of supervised and unsupervised machine learning algorithms will be deployed, including Logistic Regression, Support Vector Machine, Random Forest, and K-Modes (a version of K-Means suited to nominal data). Supervised algorithms will then be evaluated for accuracy, recall, and precision in predicting the ground truth for the target variable. In considering model evaluation against the data, K-Fold cross validation, stratified K-Fold, and Leave-One-Out will be administered for each model, the results visualized for comparison of performance metrics. The results of K-Modes will be brought before stakeholders and SMEs to open a conversation about the viability of revising Collateral Codes with respect to discovered clusters. Because there is no national standard for this critical data element, results of this nature are accepted as possible next steps, after which the above-mentioned algorithms may be applied on new data.

## Data origination

Figure - Collateral Code Process Flow

developed by authors. Numbers indicate potential edits to the Collateral Code.



## Obstacles

Given the scale of FFC’s business structure and the legacy of its largely inherited data, each line of business generates its own SQL queries when gathering data for reporting. Further compounding this inconsistency is the collection of individual data warehouses in use at FFC, and the hundreds of tables housing all customer- and account-level data. Subsequently, no two LOBs generate identical query results when investigating the same data, crippling effective communication and bringing into question the validity of Dashboards and quarterly statements. This context presented challenges for research and required collaboration with all involved LOBs to generate a sound and valid query that joined the right tables to produce a single source of truth describing all loans and collateral entities.

## Developing the Sample Dataset

After labored discussion, four tables were identified as sufficient and necessary to bring in those features that fully describe a collateral entity:

* INFORMENT.m\_COLLATERAL\_REL
* INFORMENT.m\_PRODUCT\_OFFER\_PURCHASE
* INFORMENT.m\_IP\_POP\_RELATIONSHIP
* INFORMENT.m\_INVOLVED\_PARTY

Primary keys needed to merge these tables include ‘bank\_number’, ‘appl\_id’, ‘account\_number’, and ‘collateral\_id’. The result table contains 26 features:

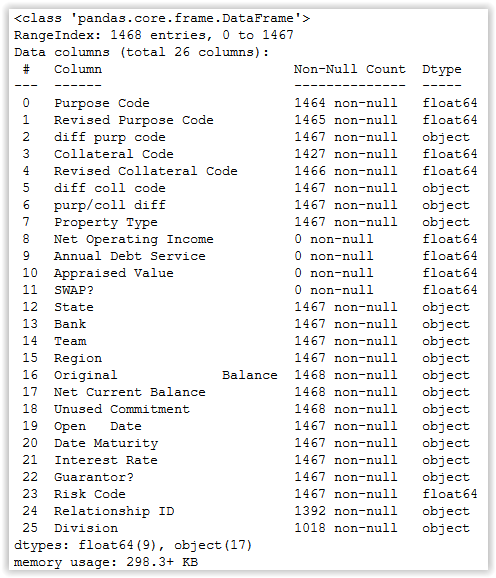


Figure 2 – List of Features, Sample Dataset

Understanding the lack of inherent reliability in the validity of any one Collateral Code assigned in the data warehouses, the first critical step was to establish ground truth. Using the validated SQL query, a sample dataset of 1467 observations was generated and distributed to RMs and Loan Ops with the request of updating the Collateral Code assigned, if necessary. Because this step required significant human effort of many individuals across departments, the size of the sample dataset represented an unfortunately small fraction of all data available for research. The results of this questionnaire were collected over the following two weeks and indicated that 41% of those Codes had been improperly assigned, based on the data of each entity available.

Table 2 – Number of Loans Recommending Changes by Industry

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Row Labels** | **Rec. Change** | **Stay Same** | **Grand Total** | **% diff** |
| Industrial | 46 | 19 | 65 | 70.77% |
| Retail | 96 | 97 | 193 | 49.74% |
| Multifamily | 126 | 136 | 262 | 48.09% |
| Homebuilder | 214 | 303 | 517 | 41.39% |
| Office | 85 | 138 | 223 | 38.12% |
| Self-Storage | 3 | 7 | 10 | 30.00% |
| Hospitality | 8 | 21 | 29 | 27.59% |
| Special Purpose | 5 | 19 | 24 | 20.83% |
| Other | 21 | 123 | 144 | 14.58% |
| **Grand Total** | **604** | **863** | **1467** | **41.17%** |

These findings provided two major insights; the current state of Code assignment is demonstrably inadequate, and a reliable ground truth had been established, so research could move forward.

## Preprocessing

Following the collection of revised Collateral Codes for the sample dataset, all individual files were merged into a single source file for analytic work.

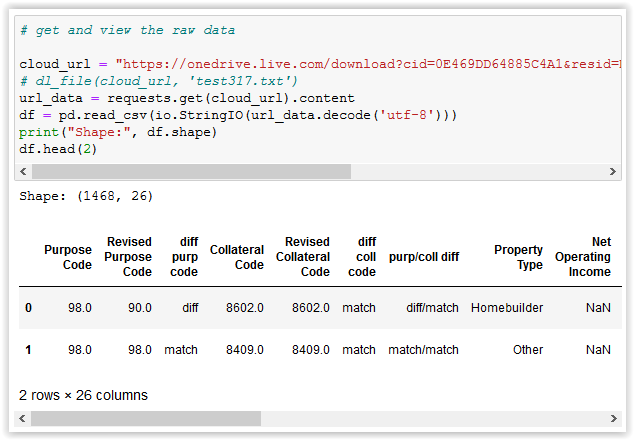


Figure 3 – Sample Dataset, Preliminary View

Initial examination of the dataset revealed several common cleaning steps would be necessary. Excess whitespace and special characters were stripped from column names and data values. Additionally, several columns identified as crucial to the full picture were fully null. These were dropped, with further discussion with LOBs noted for a future date. Because the problem statement of this research is not a time-series investigation of trends, date-time columns were also dropped. While there is a potential for time-series analysis that may identify trends in manual misclassification at each level of the loan origination process, it is beyond the scope of this research, but noted for future work.

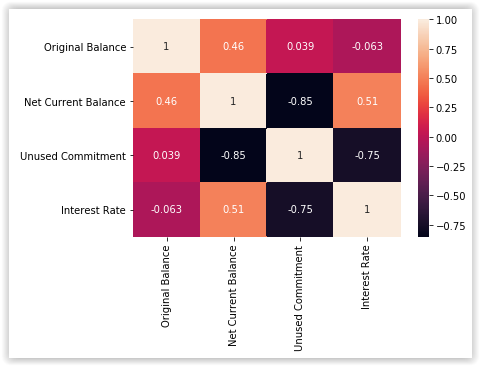
### Numeric Features

There are four numeric features in this dataset:

* Original Balance
* Net Current Balance
* Unused Commitment
* Interest Rate

An exploratory illustration of correlations within this subset reveals expected insights.

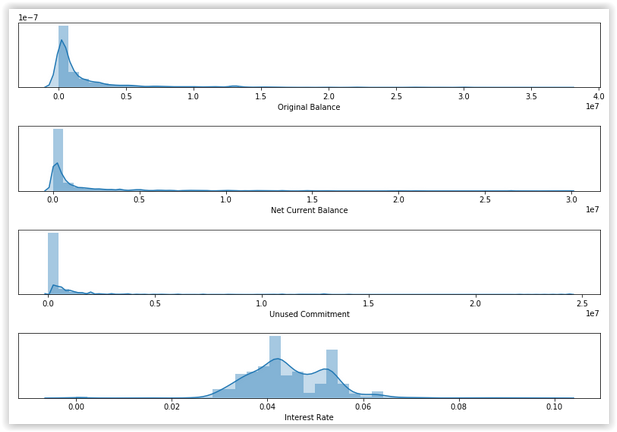
Figure 4 - Correlation Heat Map



‘Unused Commitment’ indicates funds not yet utilized and tends to correlate inversely with ‘Net Current Balance’. Because the true relationship is impacted by other underlying factors for reasons of account activity beyond the scope of present research, both features are retained for modelling.

Best practices with numeric data always involve plotting density curves, which in this setting emphasize the typical behavior of monetary data in banking.

Figure 5 – Numeric Data Density Plots



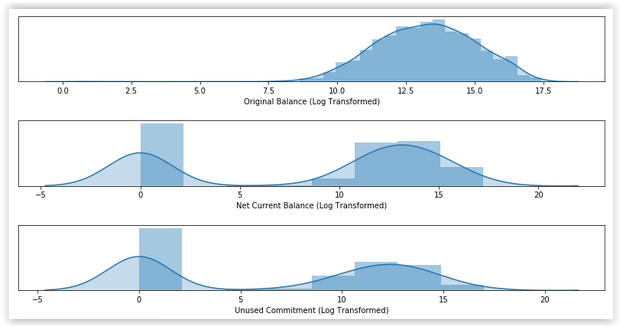
Intuitively, the distribution of ‘Interest Rate’ is over a different scale and spread than that of the other features and will be processed independently. For the other features, it is apparent that some accounts hold wealth orders of magnitude larger than the average. These may be sentinel values or human error, so will be investigated further.

Table 3 – Outliers in Numeric Data



Rows identified contain data that appears to be neither sentinel nor error, but real dollar amounts. Dropping these rows would sacrifice the dataset’s reflection of the true distribution for these features, which is by nature severely right-skewed. Instead, a log transformation will produce a more normal distribution, offering stronger predictive power to the models.

Figure 6 – Density Plots, Log Transformed

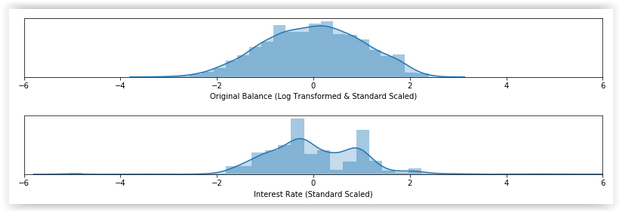


After transformation, ‘Original Balance’ is much closer to an ideal Gaussian distribution. There is one low-end outlier on this new scale, a single row dropped for ease. ‘Net Current Balance’ and ‘Unused Commitment’ are now clearly bimodal, and will be treated as binary (1 – has some value, 0 – is $0.00). New columns will be encoded as such, replacing the original columns to avoid redundancy.

It is worth noting existing criticism to log transformation and the exploration of newer alternatives to answer the problem of kurtosis (Feng et al., 2014). These methods may be useful in future work.

As a final step to prepare this data fully for consumption by machine learning algorithms, ‘Original Balance’ and ‘Interest Rate’ are standardized, a function that fits the data about µ = 0 with σ = 1.

Figure 7 – Density Plots, Standardized



### Nominal Features

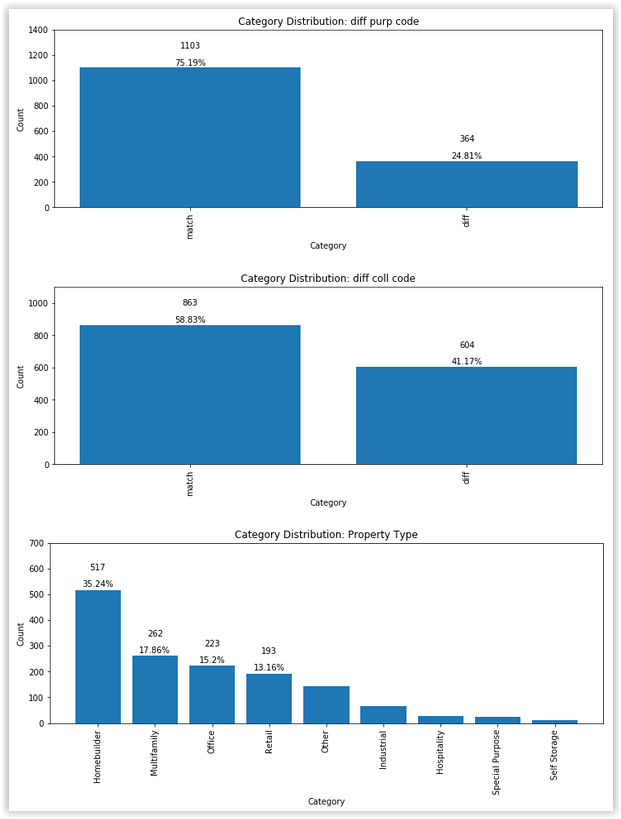
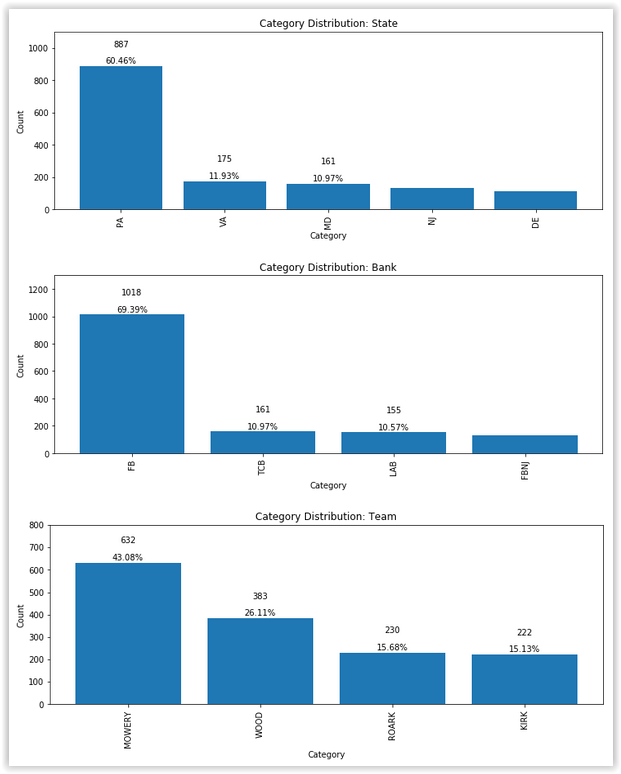
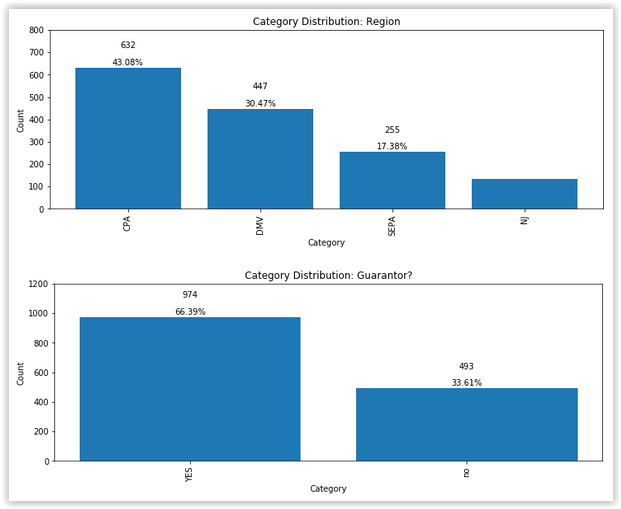


Figure 8 – Level Proportions per Categorical Feature



Domain knowledge of the data dictated that, while numeric in appearance, most columns are in fact nominal, including Purpose Code, Collateral Code, and Risk Code. These features were therefore encoded as pandas *objects* for later analysis as categorical variables eligible for encoding. Proportions of the levels within each categorical feature are visualized above.

It is worth noting that ‘diff coll code’ represents the column generated by the RMs and Loan Ops members that submitted their edits, and the 604 updated Collateral Codes are reflected from this view.

All features were next evaluated for missing data (*null* in pandas), visualized with support from the missingno python library. The decision to impute data for these missing values or exclude the impacted rows from analysis will be informed per involved feature.

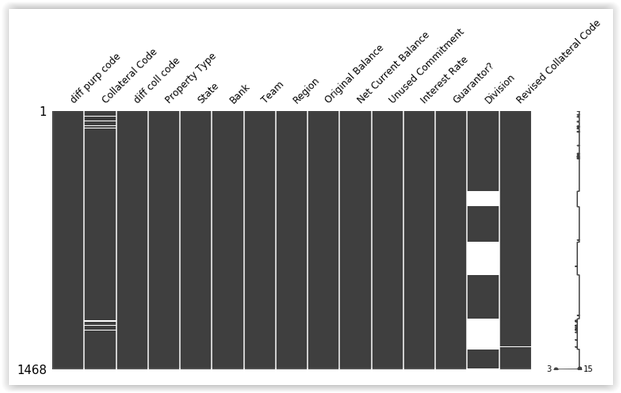


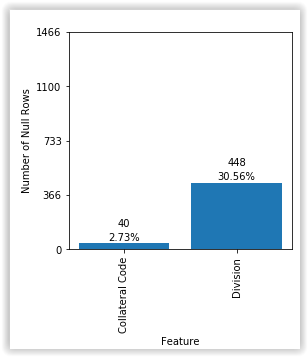
Figure 9 – Missing Data

Of the 1468 observations present in the dataset, the sparkline on the right of Figure 6 identifies rows 3 and 15 (not to be confused with index) as most egregious. These were simply dropped, following an examination of missing data within the target variable.



Figure 10 – Revised Collateral Code NULL Landscape

Figure 11 – Missing Data by Feature

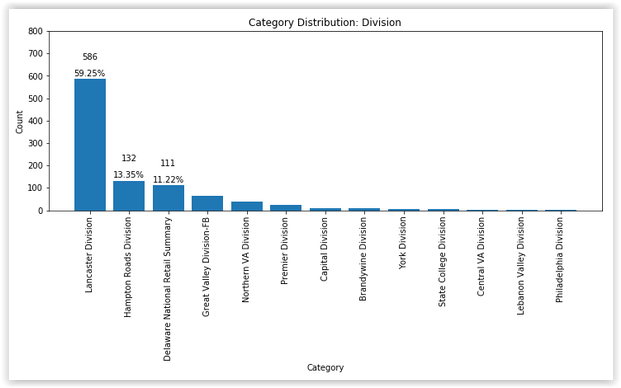


Of the remaining features with missing data, there are two common approaches; simply drop the rows affected if data loss is minimal, or impute the missing data based on the characteristics of data in that feature space.

For ‘Collateral Code’, the former option means a loss of 2.73% of the dataset and is thus chosen.

For ‘Division’ the situation is markedly different. 30% of the data would carve down an already meager sample size and cripple a model’s ability to gain appreciable insights from the data. Imputation will address this scenario. It must then be decided how to impute, which depends on the type of data in question. ‘Division’ is categorical, and one fast method fills missing rows with the mode, or most often represented level, of that feature. This option makes less sense for more uniform distributions, since the leading category may only be by a small margin, but then gains (in the current setting) an additional 30% boost in representation as the missing rows are filled in. The distribution within ‘Division’ is plotted here to level set baseline.

Figure 12 – Distribution of Feature ‘Division’



Clearly, filling all missing values of ‘Division’ with the most prevalent level ‘Lancaster Division’ would expect an accuracy score of 59%, which may be improved upon by other means.

A better alternative to brute-force mode-based imputation utilizes machine learning, treating ‘Division’ as the target variable being predicted. For this, the classifier K-Nearest Neighbors (KNN) will act on the preprocessed dataset to predict missing data for ‘Division’, just as will be done to the primary target variable, Collateral Code. A predicted distribution reflective of the known distribution will be the marker for success. However, a fully preprocessed dataset is required, so this step will be revisited in a later section.

### Encoding Nominal Features

Paramount to effective machine learning, algorithms expect categorical data to be encoded in some way. Common approaches include *label* encoding and *one hot* encoding. While both translate text to numbers, the end result may misinterpret the data. With label encoding, levels of a feature are ranked from 0 up, implying order to the categories that may not exist. Explored in great detail among data scientists (Sethi, 2020). One hot encoding avoids this pitfall with a different compromise; for all levels of a feature, as many columns are appended to the dataset with row data labeled 0 or 1 to indicate level membership. Increasing dimensionality in this way may not be feasible for large datasets, but will work for this research. For each nominal feature, the columns generated from one hot encoding are listed:

Table 4 – Effects of One Hot Encoding

|  |  |
| --- | --- |
| **Columns**  **Generated** | **Feature Name** |
| 2 | diff purp code |
| 52 | Collateral Code |
| 2 | diff coll code |
| 9 | Property Type |
| 5 | State |
| 4 | Bank |
| 4 | Team |
| 4 | Region |
| 2 | Interest Rate hi/lo |
| 2 | Guarantor? |

There will be 93 total columns after encoding is applied, still a reasonable size that will not significantly hinder runtime of models. Pandas’ method *get\_dummies* will perform the encoding, and the transformed dataset will be ready for the final stage of preprocessing.

### Imputation of ‘Division’

With numeric features transformed and standardized and nominal features encoded, the dataset is ready to train machine learning algorithms. But before addressing the original target variable ‘Revised Collateral Code’, the remaining missing data in ‘Division’ must be imputed intelligently.

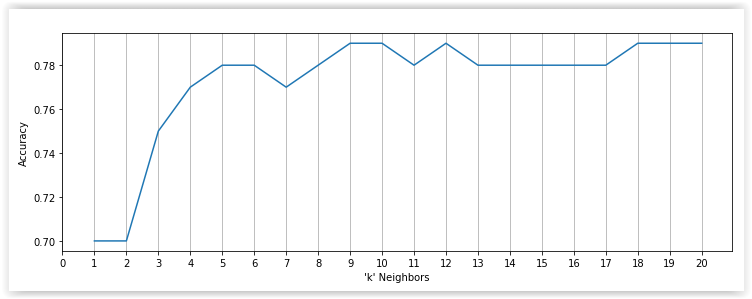
Considering ‘Division’ as the target variable, a model may train on the remaining predictive features to learn how to fill this data. It may be recalled that numeric feature ‘Interest Rate’ appeared moderately bimodal within an otherwise Gaussian distribution about µ = 0. In an effort to retain as much data as possible, two training sessions will be performed to fill ‘Division’, once having discretized ‘Interest Rate’ (respecting the bimodal behavior), and again as continuous (without addressing the modest bimodality explicitly). Comparison of results will dictate the ultimate representation.

#### ‘Interest Rate’ As Discrete

As will be the process for all models, the dataset is first split into training and testing subsets, a process accomplished in one step via sklearn’s *train\_test\_split*. By default, the training set is a 25% cut of the original data.

Details on the inner workings of KNN will be outlined below; for now, it is enough to note that a number for *K* must be chosen for the model, and to report the model’s resulting performance to judge whether imputation in this manner will be a reliable exercise.

Figure 13 – Accuracy by Choice of K Neighbors, ‘Interest Rate’ as Discrete



Iterating over possible values for *K*, the highest prediction accuracy (78.54%) is achieved at *K* = 9. From the known distribution of levels in ‘Division’, this is a clear improvement over mode-based imputation. Running KNN again with *K* = 9 produces the following classification report.

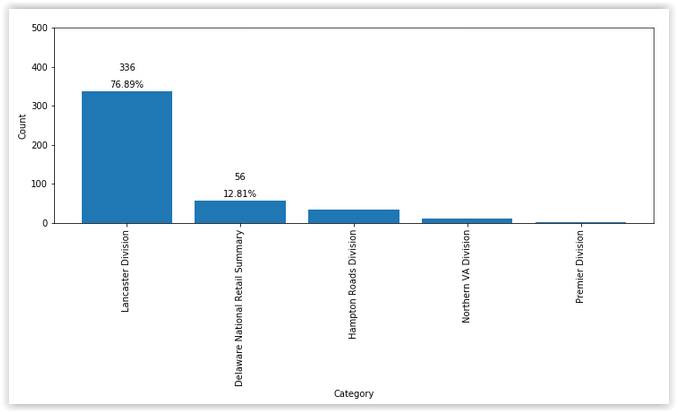
Table 5 – Classification Report, ‘Interest Rate’ as Discrete

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Level** | **Precision** | **Recall** | **F1-score** | **Support** |
| Bandywine | 0.00 | 0.00 | 0.00 | 0 |
| Central VA | 0.00 | 0.00 | 0.00 | 1 |
| Delaware Nat’l Retail | 0.96 | 0.87 | 0.92 | 31 |
| Great Valley - FB | 0.33 | 0.30 | 0.32 | 10 |
| Hampton Roads | 0.79 | 0.66 | 0.72 | 41 |
| Lancaster | 0.82 | 0.90 | 0.86 | 145 |
| Northern VA | 0.33 | 0.18 | 0.24 | 11 |
| Premier | 0.50 | 0.50 | 0.50 | 8 |
| State College | 0.00 | 0.00 | 0.00 | 0 |



Figure 14 – Confusion Matrix, ‘Interest Rate’ as Discrete

Figure 15 – Distribution of Feature ‘Division’ Predictions, ‘Interest Rate’ as Discrete



For each level, Table 4 records Precision and Recall performance metrics for the predictive model, as well as the F1-score, the harmonic mean of both. These figures are most potent for levels ‘Delaware National Retail Summary’, ‘Hampton Roads’, and ‘Lancaster’, indicating that KNN demonstrates strong predictive capability for these levels. Figure 15 highlights counts of predictions made and is color-coded by density, reflected clearly in Figure 16. The distribution of all predicted levels closely mirrors that of the ground truth, indicating that the model has discovered meaningful relationships in the data from the training set and may generalize to new data well.

Next, the same process will be administered again from the perspective of ‘Interest Rate’ as a continuous data type. This treatment has the benefit of retaining the most data, and will be preferable if results are similar.

#### ‘Interest Rate’ As Continuous

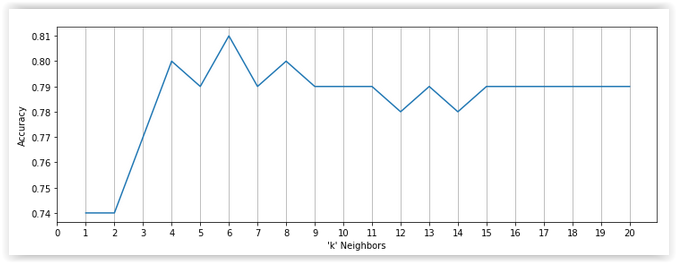


Figure 16 – Accuracy by Choice of K Neighbors, ‘Interest Rate’ as Continuous

Iterating over possible values for *K*, the highest prediction accuracy (80.57%) is achieved at *K* = 6. From the known distribution of levels in ‘Division’, this is a clear improvement over mode-based imputation. Running KNN again with *K* = 6 produces the following classification report.

Table 6 – Classification Report, ‘Interest Rate’ as Continuous

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Level** | **Precision** | **Recall** | **F1-score** | **Support** |
| Bandywine | 0.00 | 0.00 | 0.00 | 0 |
| Central VA | 0.00 | 0.00 | 0.00 | 1 |
| Delaware Nat’l Retail | 0.96 | 0.87 | 0.92 | 31 |
| Great Valley - FB | 0.55 | 0.60 | 0.57 | 10 |
| Hampton Roads | 0.75 | 0.66 | 0.70 | 41 |
| Lancaster | 0.84 | 0.91 | 0.87 | 145 |
| Northern VA | 0.50 | 0.27 | 0.35 | 11 |
| Premier | 0.57 | 0.50 | 0.53 | 8 |

Figure 17 – Confusion Matrix, ‘Interest Rate’ as Continuous

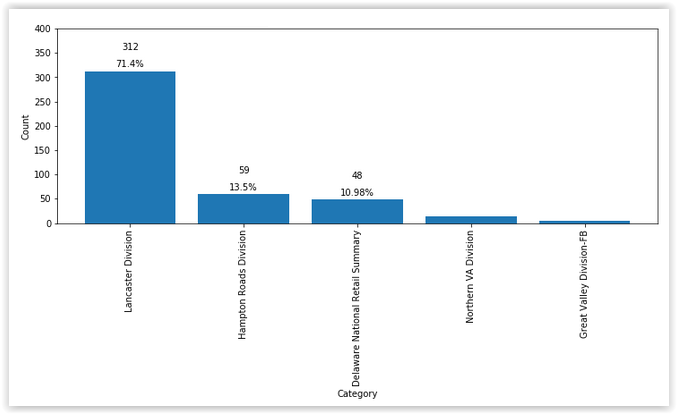


Figure 18 – Distribution of Feature ‘Division’ Predictions, ‘Interest Rate’ as Continuous

For each level, Table 5 records Precision and Recall performance metrics for the predictive model, as well as the F1-score, the harmonic mean of both. These figures are again most potent for levels ‘Delaware National Retail Summary’, ‘Hampton Roads’, and ‘Lancaster’, indicating that KNN demonstrates strong predictive capability for these levels. Figure 18 highlights counts of predictions made and is color-coded by density, reflected clearly in Figure 19. The distribution of all predicted levels closely mirrors that of the ground truth, indicating that the model has discovered meaningful relationships in the data from the training set and may generalize to new data well.

Both interpretations of ‘Interest Rate’ have led to similar predictive power of KNN on unknown ‘Division’ levels. Because treating ‘Interest Rate’ as continuous preserves the most data, this will be the data type used moving forward.

With the dataset cleaned, imputed, and encoded, the major work of this research is ready to begin. Much like was done to predict unknown levels for ‘Division’, various machine learning algorithms will be trained on a portion of the dataset to uncover underlying patterns that, to date, have driven a slower manual assignment process over multiple lines of business and with no documentation or instruction.

### Target Variable ‘Revised Collateral Code’

Figure 19 – Level Frequencies of Target Variable, ‘Revised Collateral Code’

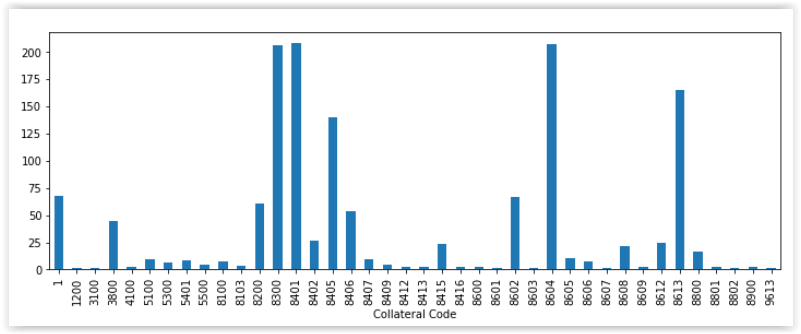


Figure 20 displays the frequencies of known labels in the dataset, and offers insight into real-world expectations of unseen data. During model evaluation and analysis, it will be critical to compare predicted proportions against the baseline, and to consider bias that may manifest in favor of majority labels. Clearly, it would be insufficient to pick a model that simply classifies all new data as Code 8604, for example, despite the misleading resultant accuracy score that would yield. A model must bear representation in mind, and balance correct classifications with misclassifications *per* Code. Stratification will assist in this goal.

# Results

## Supervised classification

With the support of Relationship Managers and members of the Loan Ops team, ground truth exists for this dataset. This allows for supervised classification, in which the algorithm operates from the context of those known labels to classify with unlabeled data. Sklearn is a Python library that offers an array of supervised classifiers.

In the next section, three supervised learning classifiers will be deployed on the dataset. Each will be evaluated to answer key questions including:

* What is the expected accuracy for prediction of new unlabeled data?
* What is the expected accuracy between labels in the target variable?
* What variation exists for the classification error?
* What can be said for model generalization?

The first method of model evaluation, training-testing-validation for a single pass on the data, removes a portion of the dataset for testing by removing the known target labels and treating those observations as new unseen data.

The next technique is Leave-One-Out (LOO) evaluation. While not generally employed in practice for its tendency of high variance, it is worth investigating here for its unique strategy. Here, the testing set is a single observation, and provides a pessimistic estimation of performance in generalized use. In practice, this leads to lower evaluation scores.

Finally, stratified K-Fold cross validation (CV) performs several passes over the dataset, each with a different partition structure for the training and testing sets. Stratification ensures that proportions of level representation are maintained in each subset, ideal here where some levels of the target variable are imbalanced. This addresses inherent bias in the data and protects the model from applying undue weight to majority categories.

Accuracy, precision, and recall are then calculated from comparing these predictions to the actual labels. Comparing these evaluation metrics across classifiers will determine which is best suited to the business problem.

### Logistic Regression

In its elementary form, regression typically applies to numeric data and may be used to predict a continuous-valued response feature. Appropriate settings include risk analysis or stock market trends. Because prediction in this setting classifies data into distinct categories, a variation known as multinominal logistic regression may be deployed on nominal data that has been encoded to appear numeric.

This technique applies a softmax function that predicts the probability the *i*th observation’s class, *yi,* is *k*, where *K* is the total number of classes possible.

Fitting a *LogisticRegression* object and training it on the split data, results follow:

Logistic Regression output

Single-pass Accuracy: 37.54%

Leave-One-Out Accuracy: 15.81%

Stratified 10-Fold Cross Validation Accuracy: 41.83%

Mean Classification scores:

Accuracy: 41.83%

Recall: 41.83%

Precision: 38.13%

Figure 20 – Classification Scores by K Folds, Logistic Regression

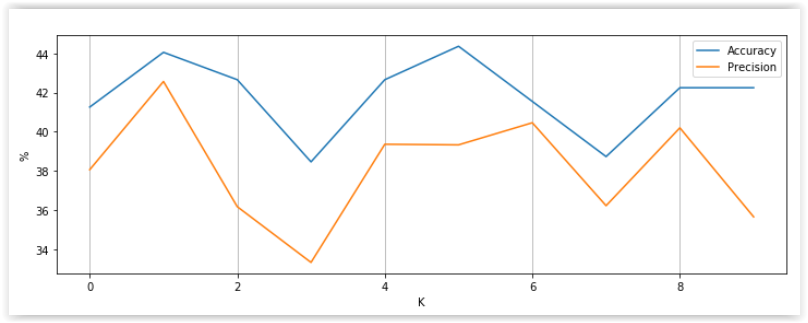
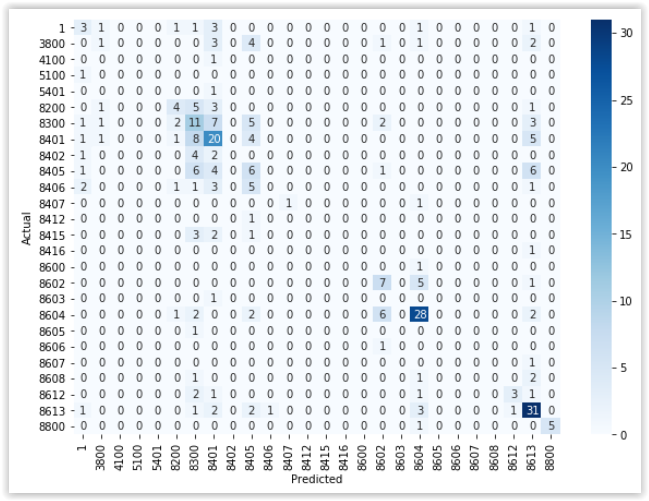


Table 7 – Best Accuracy by Code, Logistic Regression

|  |  |  |
| --- | --- | --- |
| **Expected** | **Accuracy** | **Precision** |
| 8613 | 0.7381 | 0.5345 |
| 8604 | 0.6829 | 0.6829 |
| 8401 | 0.5000 | 0.3774 |
| 8300 | 0.3438 | 0.2391 |

Figure 21 – Confusion Matrix, Logistic Regression



In a single pass, reported accuracy of the model is 37.54%, dependent on random state. LOO performed very poorly, with 15.81% accuracy. Stratified 10-Fold CV offered the best scores, with accuracy of 41.83% and precision 38.13%, averaged over 10 iterations. Figure 20 articulates each fold, and shows moderate variance, especially for precision.

Table 6 sorts predicted Collateral Codes by accuracy score from the logistic regression model, figures that are reflected visually in Figure 22. Comparing this with the known distributions of Code labels, these top results align with expectation, and demonstrate intelligence in the model beyond blind guess or brute force.

It is worth noting why recall is not displayed in Figure 21. Because the data is predicting a multi-class target variable, overall recall and accuracy define the same behavior, and therefore overlap graphically. This behavior continues for subsequent models.

### Support Vector Machine

SVM works to classify data into groups by discovering hyperplanes in *n*-dimensions that optimally isolate the data while minimizing structural risk. Another algorithm suited to encoded nominal data, the support vectors are those data points that define the bounds of each hyperplane, and SVM iteratively moves these planes until gain is negligible.

Deploying SVM on the dataset produces:

Support Vector Machine output

Single-pass Accuracy: 22.81%

Leave-One-Out Accuracy: 12.11%

Stratified 10-Fold Cross Validation Accuracy: 25.40%

Mean Classification scores:

Accuracy: 25.40%

Recall: 25.40%

Precision: 11.06%

Figure 22 – Classification Scores by K Folds, SVM

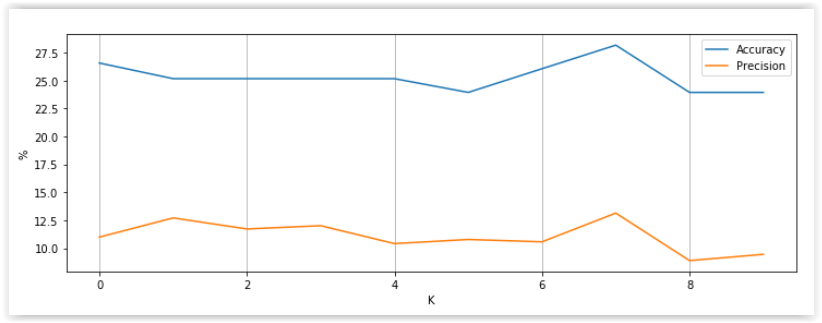
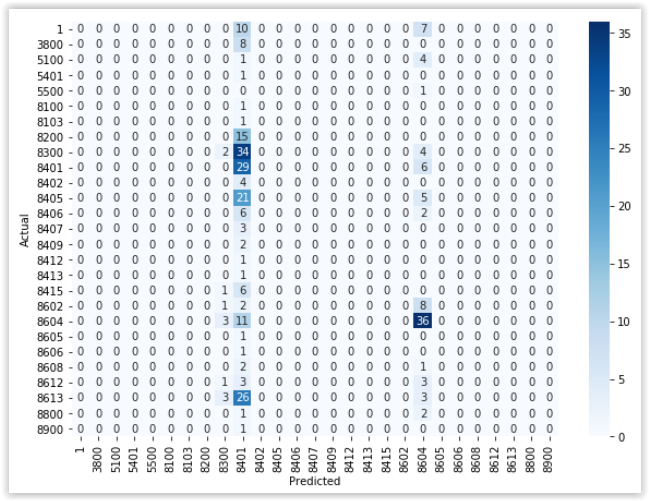


Table 8 – Best Accuracy by Code, SVM

|  |  |  |
| --- | --- | --- |
| **Expected** | **Accuracy** | **Precision** |
| 8604 | 0.7200 | 0.4390 |
| 8401 | 0.8286 | 0.1510 |

Figure 23 – Confusion Matrix, SVM



SVM demonstrated an average accuracy of 25.40% among all labels predicted, the highest precision being 43.90% for Code 8604. This Code outstanding, SVM shows little ability to predict other Codes, and would not be the appropriate model for this business need.

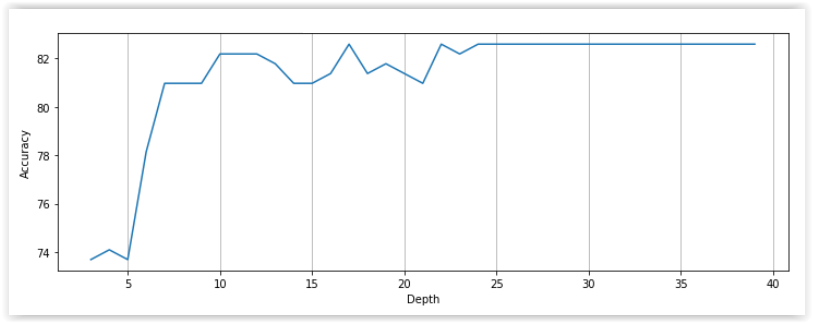
Evident from Figure 24, Code 8401 explains how accuracy can be a misleading statistic, if taken alone. While it is the case that approximately 83% of actual 8401 Codes were correctly identified, precision stresses the pitfall of brute force. Of all codes predicted to be 8401, only 15% were. It is up to researchers and clients to weigh the cost of misclassification. In the setting of this research, such performance adds little value to the business, and fails to outperform untrained human effort this research aims to replace.

### Random Forest

Random Forest (RF) is an ensemble classifier, so named as it is built of many individual Decision Tree classifiers, producing a democratic vote for the predicted label of new data. This algorithm tends to generalize well, and is expected to produce actionable results in this setting.

As with K-Fold cross validation, one must tell the algorithm how large a tree may become. Findings based on tree depth between 1 and 40 levels are plotted in Figure 25.

Figure 24 – Accuracy by Tree Depth



A tree depth of 10 strikes a balance between high accuracy and parsimonious tree depth, and is therefore selected for the model parameter. Output is as follows:

Random Forest output

Single-pass Accuracy: 44.21%

Leave-One-Out Accuracy: 17.16%

Stratified 10-Fold Cross Validation Accuracy: 46.87%

Mean Classification scores:

Accuracy: 46.87%

Recall: 46.87%

Precision: 43.19%

Table 9 – Best Accuracy by Code, Random Forest

Figure 25 – Classification Scores by K Folds, Random Forest



|  |  |  |
| --- | --- | --- |
| **Expected** | **Accuracy** | **Precision** |
| 8613 | 0.8571 | 0.4737 |
| 8604 | 0.8750 | 0.7292 |
| 8602 | 0.8000 | 0.9231 |
| 8401 | 0.7027 | 0.3023 |
| 8300 | 0.3019 | 0.3265 |



Figure 26 – Confusion Matrix, Random Forest

The third supervised approach applied to the dataset, Random Forest shows the greatest potential to operate on discovered patterns in the data and correctly classify Collateral Codes in place of human effort. Of particular note are Codes 8604 and 8602. RF demonstrated high accuracy and precision in its predictive capability with these labels. It may be desirable to use RF as a preliminary backend filter for incoming data that checks for these Codes, pushing the remainder downstream to manual classification.

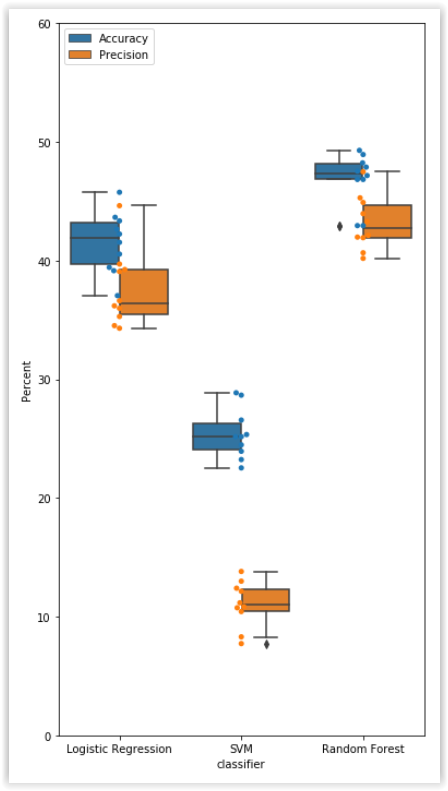


Figure 27 - Comparison of Supervised Models

Comparison of supervised classifiers is plotted in Figure 28. It is expected that more data would bolster the performance of all, but preliminary findings suggest Support Vector Machines are ill-suited for this dataset and task. Poor accuracy was met with poorer precision, offering little reliability over manual effort.

Logistic Regression and Random Forest show promising and comparable results, especially for a few Codes that cover a significant portion of all collateral entities at FFC. Random Forest performs marginally better, and over a tighter range than Logistic Regression, offering both higher accuracy and precision in most cases.

## Unsupervised classification

While supervised learning algorithms rely on datasets that have established ground truth for the predicted variable, unsupervised learning attempts to learn from data that is not explicitly labeled by generating its own likely groupings, or *clusters*. Although this dataset does include ground truth for its data, these labels are defined by FFC, and are not industry standard. It may be worth reevaluating Collateral Codes themselves if patterns in the data suggest more suitable clusters.

### K-Modes

A variation of K-Means for continuous data, K-Modes defines clusters by the categories shared. Known as Manhattan Distance, the formula

calculates the distance between two points by summing the lengths of the projections of the line segment between those points onto *n*-coordinate axes. This classifier also requires one parameter to run, the number of clusters to draw in the feature space. Data output from this classifier include a cost score (used to rank performance between trials) and the features that define those clusters.

This algorithm is available from the Python library *kmodes*. Deploying K-Modes on the dataset (with target variable removed) yields the following cluster behavior:

Lowest cost: 5306

Clusters identified: 27

Table 10 – K-Modes Cluster Definitions

|  |  |  |
| --- | --- | --- |
| Cluster: 1  diff purp code\_match  Collateral Code\_8604.0  diff coll code\_match  Property Type\_Homebuilder  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_high  Guarantor?\_no  has\_Unused\_Commitment  Lancaster Division | Cluster: 2  diff purp code\_match  diff coll code\_match  Property Type\_Other  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment | Cluster: 3  diff purp code\_match  Collateral Code\_8613.0  diff coll code\_match  Property Type\_Homebuilder  State\_VA  Bank\_FB  Team\_WOOD  Region\_DMV  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Hampton Roads Division |

|  |  |  |
| --- | --- | --- |
| Cluster: 4  diff purp code\_diff  diff coll code\_diff  State\_MD  Bank\_TCB  Team\_WOOD  Region\_DMV  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Delaware National Retail Summary | Cluster: 5  diff purp code\_match  diff coll code\_match  State\_VA  Bank\_FB  Team\_WOOD  Region\_DMV  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division | Cluster: 6  diff purp code\_match  diff coll code\_match  State\_PA  Bank\_FB  Team\_KIRK  Region\_SEPA  Interest Rate hi/lo\_low  Guarantor?\_no  has\_Unused\_Commitment  Lancaster Division |
| Cluster: 7  diff purp code\_match  Collateral Code\_8401.0  diff coll code\_match  Property Type\_Office  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division | Cluster: 8  diff purp code\_match  Collateral Code\_8401.0  diff coll code\_match  Property Type\_Office  State\_DE  Bank\_FB  Team\_ROARK  Region\_DMV  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division | Cluster: 9  diff purp code\_match  Collateral Code\_8604.0  diff coll code\_match  Property Type\_Homebuilder  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Lancaster Division |
| Cluster: 10  diff purp code\_diff  diff coll code\_match  State\_MD  Bank\_TCB  Team\_WOOD  Region\_DMV  Interest Rate hi/lo\_low  Guarantor?\_no  has\_Unused\_Commitment  Lancaster Division | Cluster: 11  diff purp code\_match  Collateral Code\_8200.0  diff coll code\_diff  Property Type\_Multifamily  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Hampton Roads Division | Cluster: 12  diff purp code\_match  diff coll code\_match  State\_NJ  Bank\_FBNJ  Team\_ROARK  Region\_NJ  Interest Rate hi/lo\_high  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division |
| Cluster: 13  diff purp code\_match  Collateral Code\_8606.0  diff coll code\_diff  Property Type\_Homebuilder  State\_VA  Bank\_FB  Team\_WOOD  Region\_DMV  Interest Rate hi/lo\_high  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Lancaster Division | Cluster: 14  diff purp code\_match  diff coll code\_diff  Property Type\_Homebuilder  State\_PA  Bank\_LAB  Team\_KIRK  Region\_SEPA  Interest Rate hi/lo\_high  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Lancaster Division | Cluster: 15  diff purp code\_diff  diff coll code\_match  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division |

|  |  |  |
| --- | --- | --- |
| Cluster: 16  diff purp code\_match  diff coll code\_diff  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_high  Guarantor?\_no  has\_Unused\_Commitment  Lancaster Division | Cluster: 17  diff purp code\_match  diff coll code\_match  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_no  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Delaware National Retail Summary | Cluster: 18  diff purp code\_match  diff coll code\_match  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_no  has\_Unused\_Commitment  Lancaster Division |
| Cluster: 19  diff purp code\_match  diff coll code\_diff  State\_DE  Bank\_FB  Team\_ROARK  Region\_DMV  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Lancaster Division | Cluster: 20  diff purp code\_match  Collateral Code\_1.0  diff coll code\_match  Property Type\_Homebuilder  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_high  Guarantor?\_YES  has\_Unused\_Commitment  Lancaster Division | Cluster: 21  diff purp code\_match  Collateral Code\_8300.0  diff coll code\_match  Property Type\_Multifamily  State\_PA  Bank\_LAB  Team\_KIRK  Region\_SEPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division |
| Cluster: 22  diff purp code\_match  diff coll code\_match  State\_PA  Bank\_LAB  Team\_KIRK  Region\_SEPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division | Cluster: 23  diff purp code\_match  Collateral Code\_8300.0  diff coll code\_match  Property Type\_Multifamily  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division | Cluster: 24  diff purp code\_match  Collateral Code\_8606.0  diff coll code\_diff  Property Type\_Homebuilder  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Hampton Roads Division |
| Cluster: 25  diff purp code\_match  Collateral Code\_8604.0  diff coll code\_match  Property Type\_Multifamily  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  has\_Unused\_Commitment  Lancaster Division | Cluster: 26  diff purp code\_match  diff coll code\_diff  State\_PA  Bank\_FB  Team\_MOWERY  Region\_CPA  Interest Rate hi/lo\_low  Guarantor?\_YES  has\_Net\_Current\_Balance  Lancaster Division | Cluster: 27  diff purp code\_match  Collateral Code\_8604.0  diff coll code\_match  Property Type\_Homebuilder  State\_DE  Bank\_FB  Team\_WOOD  Region\_DMV  Interest Rate hi/lo\_high  Guarantor?\_YES  has\_Unused\_Commitment  Lancaster Division |

|  |  |
| --- | --- |
| **Feature** | **Frequency** |
| diff purp code\_match | 24 |
| Bank\_FB | 21 |
| Guarantor?\_YES | 21 |
| Lancaster Division | 21 |
| has\_Net\_Current\_Balance | 20 |
| Interest Rate hi/lo\_low | 20 |
| diff coll code\_match | 19 |
| State\_PA | 18 |
| has\_Unused\_Commitment | 16 |
| Region\_CPA | 14 |
| Team\_MOWERY | 14 |

It may be important for SMEs to know which features contribute most to cluster identification (Table 10). Context will also be critical, as the presence of ‘Bank FB’ is a rare occurrence in the dataset (see Figure 9), yet appears highly influential in defining cluster space. Alternatively, those features not appearing on this list may be given significant weight in practice, blurring the meaning of current Collateral Codes.

Table 11 – K-Modes Frequency of Features

This is one of many unsupervised approaches used extensively today by statisticians and researchers, and offers a unique reframe of the question founding this research. Other similar algorithms may offer out-of-the-box thinking to the problem of Collateral Codes.

# Discussion

## Model Evaluation

As with all machine learning, more data is generally better. Gleaning ground truth for this dataset was an entirely manual process, and a significant ask for several lines of business to dedicate their time. Utilizing ground truth allowed for the exploration of several supervised approaches, the leading of which was Random Forest classification, which performed very well for a few prominent codes. While full automation may not be practical today, supplementing manual entry with Random Forest classification could alleviate up to 20% of current time and labor by assigning those codes where confident and passing the rest on to Relationship Managers to assign as usual. With new data, these algorithms are expected to gain precision with other codes as well, and may gradually take over the assignment of all codes.

## Threats to validity

It was realized early on that Collateral Codes suffer from a lack of integrity for several key reasons. This may be the case for any other feature in the dataset, since they all flow down the same origination process. This research relies on the assumption that these data are reasonably accurate and reliable. In addition, machine learning algorithms are notoriously brittle to time, and future work with these algorithms relies on these data processes remaining largely consistent.

# Future work

## Next steps

Effective data management and modeling is an ongoing endeavor at FFC. The researchers hope this work contributes to that effort, and illuminates novel solutions to persistent data problems.

Future work could go one of two directions. If working with the current list of Collateral Codes, building on the current dataset will bolster the ground truth for all codes, and make these algorithms more robust to new data. If, on the other hand, it is decided to reevaluate the list of Collateral Codes, there may already be distinct hyperplanes that can cluster new data more accurately than human intuition has allowed. Beyond the present work with Collateral Codes, these same approaches may apply to other data entered manually, saving costs in time and labor, and elevating the reliability of data at Fulton.

## Outstanding goals

Training documentation would strengthen the reliability of Collateral Codes, the formation of which was unfortunately beyond the scope of this research. With time and funding, such documentation may be developed for all manuallyentered fields, formalizing the underlying patterns and logic that machine learning algorithms strive to discover.

## Potential implications

The financial impact of unreliable Codes was roughly estimated, and should not be taken as a conclusion of significance to this research. It is mentioned here as solely a demonstration of the potential impact to FFC. If the number of loans on record can be assumed accurate by the queries performed (around 130k), and that the actual number of loans that will be in new systems will be much less than that due to the lack of Residential mortgages, leases, lines of credit, small business, etc., it may be estimated that the number of loans currently to be around 35,000. If credit analysts take 5 minutes to review the Collateral Code that exists, compare it to actual code, and input it into the system, and cost per reviewer is $35/hr., the cost estimate of this work around could be roughly estimated to be 3,000 hours at a cost of $105,000. This researcher would indicate that the time to review the code and input it might be much longer than 5 minutes, and the cost per hour for the reviewer may vary, so this estimate might change. This also does not include the opportunity cost to the business from this task being performed.

# Reflection

It has been the ambition of the researcher to demonstrate the potential for advanced statistical models and the application of machine learning to supplement and ultimately dominate roles otherwise prone to human error. Through this work with Collateral Codes at Fulton Financial Corporation, results show real and tangible promise of this eventuality. Any room for improvement is exciting and fuels continued exploration in these models and processes.

## List of Collateral Codes

|  |  |
| --- | --- |
| **0000 SERIES UNSECURED:** | |
| 1 | UNSECURED |
| **1000 SERIES STOCK:** | |
| 1100 | STOCKS (our bank) |
| 1200 | COMMON LISTED STOCK |
| 1201 | UNLISTED STOCK |
| **2000 SERIES BONDS:** | |
| 2100 | BONDS |
| **3000 SERIES UCC'S** | |
| 3100 | ACCTS RECEIVABLE |
| 3200 | INVENTORY |
| 3300 | EQUIPMENT - ALL TYPES (including farm) |
| 3400 | FARM PRODUCTS |
| 3500 | FIXTURES |
| 3600 | AIRCRAFT |
| 3700 | WATERCRAFT |
| 3800 | ALL BUSINESS ASSETS/MULTIPLE SECURITIES |
| 3900 | UCC RECORDED AT REC OF DEEDS OFFICE |
| **4000 SERIES TITLES:** | |
| 4057 | SNOW MOBILE |
| 4058 | TRAVEL TRAILER |
| 4059 | MOTORCYCLE |
| 4060 | CAMPER |
| 4100 | NEW VEHICLE |
| 4101 | TRACTOR TRAILERS |
| 4200 | USED VEHICLE |
| 4500 | TITLED EQUIPMENT |
| **5000 SERIES ASSIGNMENTS:** | |
| 5100 | ASSIGN OF DEPOSIT |
| 5300 | ASSIGN OF LEASE & RENT |
| 5301 | ASSIGN OF EQUIPMENT LEASE |
| 5400 | ASSIGN OF IRREVOCABLE TRUST |
| 5401 | ASSIGN OF INVESTMENT/BROKERAGE ACCT/ MUTUAL FUNDS |
| 5500 | MISC ASSIGNMENTS |
| 5600 | ASSIGN OF SALES CONTRACT |
| 5700 | ASSIGN OF NOTES REC (ASSUMPTIONS), SECURED BY R/E |
| 5800 | ASSIGN OF PARTNERSHIP INTEREST |
| **6000 SERIES OTHER:** | |
| 6100 | ASSIGN OF LIFE INSURANCE POLICY |
| 6300 | NEG PLEDGE AGREEMENT (REAL ESTATE) |
| 6400 | CASH VALUE LIFE INSURANCE |
| **7000 SERIES IDA:** | |
| 7300 | REVENUE NOTES |
| 7400 | GENERAL OBLIGATION NOTES |
| **8000 SERIES MORTGAGES:** | |
| 8000 | MTG MOBILE HOME |
| 8001 | MTG PREFERRED SHIPS MORTGAGE (FISHING VESSELS) |
| 8100 | MTG 1-4 FAMILY OWNER OCCUPIED |
| 8101 | MTG CONDOMINIUM |
| 8103 | MTG TOWNHOMES |
| 8200 | MTG 1-4 FAMILY N/O/O INVESTMENT |
| 8300 | MTG MULTIFAMILY INVESTMENT/RENTAL |
| 8401 | MTG OFFICE BUILDING |
| 8402 | MTG HOTEL/ MOTEL |
| 8403 | MTG RESTAURANT |
| 8404 | MTG OFFICE/WAREHOUSE BUILDING |
| 8405 | MTG RETAIL SPACE |
| 8406 | MTG INDUSTRIAL |
| 8407 | MTG SELF STORAGE |
| 8408 | MTG MEDICAL BUILDING |
| 8409 | MTG CHURCH |
| 8410 | MTG EDUCATIONAL |
| 8411 | MTG RECREATIONAL COMPLEX |
| 8412 | MTG DAYCARE CENTER |
| 8413 | MTG MARINA |
| 8414 | MTG GOLF COURSE |
| 8415 | MTG OTHER SPECIAL PURPOSE |
| 8416 | MTG AUTO (CAR LOT/CAR WASH/AUTO REPAIR) |
| 8417 | MTG CONVENIENCE STORE |
| 8418 | MTG GAS STATION |
| 8419 | MTG OFFICE/CONDOMINIUMS |
| 8420 | MTG OFFICE/RETAIL |
| 8421 | MTG RETAIL CONDOMINIUMS |
| 8422 | MTG RETAIL SHOPPING CENTERS |
| 8600 | MTG PRE CONSTRUCTION & LAND DEVELOPMENT |
| 8601 | MTG CONSTRUCTION HOTEL/ MOTEL |
| 8602 | MTG RESIDENTIAL LOTS (FINISHED) |
| 8603 | MTG COMMERCIAL LOTS (FINISHED) |
| 8604 | MTG RESIDENTIAL LOTS- UNDER DEVELOPMENT |
| 8605 | MTG COMMERCIAL LOTS- UNDER DEVELOPMENT |
| 8606 | MTG 1-4 FAMILY UNDER CONSTRUCTION/ RENOVATION |
| 8607 | MTG COMMERCIAL PROP- CONSTRUCTION/RENOVATION |
| 8608 | MTG CONSTRUCTION APARTMENT BUILDINGS |
| 8609 | MTG CONSTRUCTION OFFICE BUILDINGS |
| 8610 | MTG CONSTRUCTION OFFICE CONDOMINIUMS |
| 8611 | MTG CONSTRUCTION RELIGIOUS FACILITIES |
| 8612 | MTG CONSTRUCTION RETAIL BUILDINGS |
| 8613 | MTG CONSTRUCTION 1-4 FAMILY RESIDENTIAL HOME (new home construction) |
| 8614 | MTG CONSTRUCTION TOWNHOUSES & DUPLEXES |
| 8615 | MTG CONSTRUCTION RESIDENTIAL CONDOMINUIMS |
| 8800 | MTG UNDEVELOPED/VACANT LAND |
| 8801 | MTG VACANT LAND RESID/MULTI-FAMILY |
| 8802 | MTG VACANT LAND INDUSTRIAL |
| 8803 | MTG VACANT LAND AGRICULTURAL |
| 8900 | MTG FARMLAND (INCLUDING BUILDINGS) |
| **9000 SERIES INSURANCES:** | |
| 9080 | TITLE INSURANCE |

# References

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log-transformation and its implications for data analysis. In *Shanghai Arch Psychiatry* 26(2) (pp 105-109).

Sethi, A. (2020). One-hot encoding vs. label encoding using scikit-learn. *Analytics Vidhya*. Available from: <https://www.analyticsvidhya.com/blog/2020/03/one-hot-encoding-vs-label-encoding-using-scikit-learn/>