SPRUCE

Evergreens

Research Project:
Results and Recommendations

Team Evergreens

Wendy Zhang, Tianyu Wong, Junwei Xu,

Kaiwen Luo, Qian Zhao, Xinying Xu, Kewen Gu

Executive summary

As a neutral third party that uses FinTech to coordinate transactions between homeowners and their lender or real estate institution faster and cheaper, Spruce aims to leverage technology to improve title insurance assessment and issuance to reduce the time and cost needed to close a real estate deal. During the semester, Team Evergreens explored a dataset with 54 variables and over 500,000 observations to help Spruce determine the features that best predict title defects.

This report covers the project understanding, research questions, hypothesis, and the two main business problems that Spruce faces: rising competition and threats from new technology. This report also reviews the evolution of our understanding of Spruce's business and the treatment of the data. As we gain a deeper understanding along the way, there are adjustments to the methodologies and evaluation criteria applied.

The technical part of this report includes data pre-processing, data exploration, data visualization, and data processing. Oversampling and undersampling techniques are leveraged to overcome the problem with imbalanced data. After rebalancing the data, ten feature selection methods are used to select the top features that lead to a higher likelihood of title defect. After thoughtful consideration, we decided to use five models, including Linear Regression, Logistic Regression, Random Forest, XGBoost, and Support Vector Machine, to test the predictiveness of the selected features from

each feature selection method. The combination of the Elastic Net feature selection method and Linear Regression model generates the best results.

The features selected by Elastic Net lead us to the business part of this report. The nine variables are E_DENSITY, E_POV_TOTAL, AreaBuilding, E_POP10, E_MEDVALOCC, TaxAssessedValueTotal, TaxMarketValueTotal, E_MEDHHINC, and E_NONFAMHH. We then dive deeper into each of these selected variables to speculate the reason for the correlation between each feature and title defects and suggest what Spruce can do with these variables.

Project Understanding

Based on the business challenges of rising competition and threats of new technology we identified for Spruce, we constructed the following research questions:

a. Can historical property transactional data sufficiently predict title defects?

b. If so, which features of the data are significant in the predictive process?

Our null hypothesis for the first research question is that the given historical data cannot accurately predict title defects.

A. Project Timeline

The timeline of our project is as follows:

Week 1: Define Spruce's business problem, project purpose, and scope

Week 2: Explore data and raise questions about any outliers and unclear variables

Week 3-4: Clean data, use feature selection method to select the most predictive variables for defective titles and test selected variables using logistic regression, random forest, and gradient boosting classification models

Week 5: Realize problems with our methods and adjust subsequent methods

Week 5~10: Test predictiveness of selected variables using other models while keep refining our methods based on trial and error

Final Weeks: Synthesize findings and prepare deliverables for final report and presentation.

B. Criteria

- True Positives: correctly classified title defect
- False Positives: incorrectly classified title defect
- True Negative: correctly classified non-title defect
- False Negative: incorrectly classified non-title defect
- Accuracy: # of correctly classified obs / total # of obs
- Precision: True Positives / (True Positives + False Positives)
- Recall: True Positives / (True Positives + False Negatives)
- F-1: 2 * (Precision * Recall) / (Precision + Recall). F-1 Score can be interpreted as a weighted average of the precision and recall, or a balance between the two.

C. Deliverables

The deliverables for our project include an accurate predictive model, a presentation to our clients and also a detailed report.

Evolutions and Adjustments

Initially, we followed the standard procedures of data analytics: we performed data exploration, preprocessing, feature selection, and model construction. The most prominent characteristic of the given dataset is that it's extremely imbalanced, with a 0.35:99.65 ratio of defective versus non-defective title observations. As we had little knowledge of dealing with imbalanced data, we neglected this imbalance. However, the results we obtained were not promising: all models failed to predict positive observations as the portion of positive entries is too small in the original dataset. After that, we focused on using imbalanced learning techniques to increase true positive predictions. After much research and self-learning on the Internet, we found resampling to be a great way to tackle this problem. Although applying resampling increases true positives, it also reduces true negatives, which becomes another problem for us. Therefore, it confirms with us that our goal for this project is to find a balance between predicted true positives and true negatives.

We initially used accuracy as a metric to measure the performance of our models. However, later we found out that accuracy is not a good measure for assessing how well we did on predicting positive or negative observations. Thus, we switched to use recall, precision, and f-1 scores as the new measures on our model performance.

Data Processing

A. Data exploration

We are given a dataset acquired from the land offices in three counties in New York and includes 500,250 observations and 54 variables. Based on the data dictionary, there are three feature groups: property characteristics, homeowner demographic profile, and Equifax credit and risk characteristics. Our target variable or the outcome variable is dr_Title_Defect_Ind, where value 0 indicates no title defects, and value 1 indicates title defects. Most of the properties have no title defects while the proportion of defective title accounts for less than 0.35%. Hence, we categorized the data frame as an imbalanced. In addition, we discovered several variables with a large percentage of missing values and some columns with outliers, we dealt with the issue in the following steps based on our professional perspective.

B. Preprocessing

a. Fill Missing Values

We used the function missing_value_table to evaluate the number and percentage of missing values in each column. Some variables like TaxMarketImprovementsPerc have nearly 50% of null values. In order to avoid being put into the risks of losing valuable information, we did not plan to simply delete those data entries. Instead, we decided to fill out those empty cells by referring to the existing data points. For numerical variables, median values were calculated and used to replace the NULL values. For categorical variables, missing cells were filled with the most frequent values within that column.

b. Handle Outliers

An outlier is a data point that is distant from the rest of the observations. Normally, outliers are presented due to mistakes made during the data collection or data inputting phase. To identify those extreme values in our data frame, we used the median and quartile range. Compared to other numeric statistics like mean or standard deviations, median and quartile range are less sensitive to outliers. Box plots were also utilized to locate the outliers in each variable. For the purpose of not affecting the prediction models, we decided to trim out those data entries which have outliers

c. Splitting the Data

Before proceeding with resampling techniques to balance the dataset, we split the data into the training dataset and testing dataset. For the purpose of testing the accuracy of our selected features and models, we need to test the models on the original testing set instead of the testing set generated from the resampling techniques. The main objective is to avoid overfitting or underfitting issues and improve the accuracy of our predictive models.

C. Resampling Techniques

a. Oversampling with SMOTE

Since the major issue for our dataset is imbalanced data, we explored methods to make our dataset more balanced and representative. Recently, oversampling the minority class observations has become a common approach to improve the quality of predictive modeling. By oversampling, models are sometimes better able to learn patterns that differentiate classes. We used the SMOTE (Synthetic Minority Oversampling Technique) to randomly increase the size of the minority class, which significantly increased our recall score.

b. Undersampling with NearMiss

Another way to cope with imbalanced data is to undersample the majority class. And NearMiss is an under-sampling technique. It aims to balance class distribution by randomly eliminating majority class examples. When instances of two different classes are very close to each other, we remove the instances of the majority class to increase the spaces between the two classes. This helps in the classification process. To prevent the problem of information loss in most under-sampling techniques, near-neighbor methods are widely used.

Model Selection

a. Linear regression

Linear regression model is an analysis that uses the least square function to model the relationship between independent and dependent variables. If regression analysis includes two or more independent variables and the relationship between dependent and independent variables is linear, it is called multiple linear regression analysis. With simple logic, the linear regression model is easy to implement, and the results also have good interpretability.

b. Logistic regression

Logistic regression model is a classification method that uses a logistic function to model a binary dependent variable. It is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

c. Random Forest

Random forest is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

d. XGBoost

Gradient boosting is a powerful machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion as other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

e. Support Vector Machine

The basic model of Support Vector Machine is to find the best separation hyperplane on the feature space to maximize the interval between positive and negative samples on the training set. SVM is a supervised learning algorithm used to solve binary classification problems. SVM can also be used to solve nonlinear problems with the introduction of kernel methods.

Feature Selection and Results

Feature selection is important before modeling because we can select significant variables to improve model accuracy. The three key benefits of performing feature selection on our data are: reducing overfitting, increasing model accuracy, and decreasing model training time.

a. Univariate Selection

Univariate feature selection works by selecting the best features based on univariate statistical tests. It can be seen as a preprocessing step to an estimator. Univariate feature selection uses the scikit-learn library, which provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features. And there are 8 variables were selected based on Univariate selection.

	Recall Score	Precision Score	F1 Score
Logistic	0.72591	0.003167	0.006308
Random Forest	**0.925053**	**0.003479**	**0.006932**
XGBoost	0.983654	0.004565	0.006278
SVM	0.970454	0.003425	0.006826

b. Feature Importance Selection

We can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives us a score for each feature of our data, the higher the score more important or relevant is the feature towards your output variable. Feature importance is an inbuilt class that comes with Tree-Based Classifiers, we will be using Extra Tree Classifier for extracting the top 8 features for the dataset.

	Recall Score	Precision Score	F1 Score
Logistic	0.834051	0.003505	0.006981
Random Forest	**0.982758**	**0.003660**	**0.007293**
XGBoost	0.976932	0.003065	0.007216
SVM	0.982758	0.003658	0.007289

c. Forward Stepwise Selection

Forward stepwise selection is an expansionary method of searching for the most predictive features that are best suited for the machine learning algorithm by starting out with zero features, then add the best feature to the set, and then the next best feature from the remaining set, and so on. Using this method, we started with selecting the top 8 features by setting the number of features to be picked (k_features) to 8 and also set k_features to 10, 12 and 15 to test for the optimal combination of features.

	Recall Score	Precision Score	F1 Score
Linear	0.857788	0.003395	0.006763
Logistic	**0.952596**	**0.003583**	**0.007139**
Random Forest	0.988713	0.003516	0.007008
XGBoost	0.986455	0.003512	0.006998
SVM	0.735892	0.004682	0.009306

d. Backward Stepwise Selection

Backward Stepwise Selection is to collect all variables first and test the model by deleting one variable to see if there are any significant changes. Then, repeating the process until there is no significant loss.

	Recall Score	Precision Score	F1 Score
Linear	**0.783296**	**0.003203**	**0.006380**
Logistic	0.794582	0.003068	0.006113
Random Forest	0.932280	0.003329	0.006635
XGBoost	0.887133	0.003195	0.006366
SVM	0.512415	0.002199	0.004379

e. Hybrid Stepwise Selection

Hybrid Stepwise Selection is a combination of forward and backward selection. The initial step is the same as forward selection, which starts with no features and incrementally adds features with the lowest significant p-value. Different from the forward selection, it will also remove any features that were previously added that now have an insignificant p-value. This makes the final model to have all of the features included being significant.

	Recall Score	Precision Score	F1 Score
Linear	**0.896163**	**0.003745**	**0.007458**
Logistic	0.963883	0.003477	0.006928
Random Forest	0.972912	0.003463	0.006901
XGBoost	0.970655	0.003458	0.006892
SVM	0.878104	0.003463	0.006898

f. Ridge Regression

Compared with linear regression, ridge regression has an additional alpha parameter which is used to tune the model and helps to prevent multicollinearity. Alpha is set manually. As the value of alpha increases, the magnitude of the coefficients of variables would decrease. In order to pick the best tuning parameter, we used ridge cross-validation. After running the model, we found that Ridge picked 52 variables and eliminated the other 2 variables based on the coefficients of variables. The 52 variables were used during the model evaluation process. We also picked the top 8 variables that have the highest absolute value of coefficients and compared the recall, precision, and f-1 score with that of the model with 52 variables. It turns out that SVM with 8 variables has the best performance.

	Recall Score	Precision Score	F1 Score
Linear	0.483069	0.007210	0.014209
Logistic	0.489841	0.007168	0.014130
Random Forest	0.702031	0.004281	0.008511
XGBoost	0.681715	0.005384	0.010685
SVM	**0.444695**	**0.007997**	**0.015712**

g. Lasso Regression

Lasso is a compression estimation. By constructing a penalty function, it can get a more refined model and make it compress some regression coefficients, that is, forcing the sum of absolute values of the coefficients less than a fixed value, as well as set some coefficients to zero. Therefore, it contains the advantage of subset contraction and is a biased estimation for processing data with complex collinearity. Besides, Lasso regression is developed on the basis of ridge regression. If the dataset contains too many features and needs to be compressed, Lasso regression would be a wise choice. Usually, the ordinary linear regression model is enough.

	Recall Score	Precision Score	F1 Score
Linear	**0.686230**	**0.006299**	**0.012485**
Logistic	0.726862	0.005676	0.011264
Random Forest	0.988713	0.003522	0.007019
XGBoost	0.988713	0.003521	0.007018
SVM	0.975169	0.003630	0.007233

h. Elastic Net Regression

Elastic Net Regression combines both penalties of Ridge and Lasso, and gives us the benefits of both models. It has been found to have predictive power better than Lasso, while still performing feature selection. Therefore, we can get the best of both worlds, performing feature selection of Lasso with the feature-group selection of Ridge.

	Recall Score	Precision Score	F1 Score
Linear	0.519187	0.010297	0.020194
Logistic	0.747178	0.003020	0.006016
Random Forest	**0.927765**	**0.003308**	**0.006592**
XGBoost	0.887133	0.003216	0.006408
SVM	0.914221	0.003265	0.006506

i. Principal Component Analysis

Principal component analysis finds a new set of dimensions such that all the dimensions are orthogonal (and hence linearly independent) and ranked according to the variance of data among them. It means more important principle axis occurs first. More importance means more variance and more spread out data.

	Recall Score	Precision Score	F1 Score
Linear	**0.643340**	**0.006162**	**0.012207**
Logistic	0.884875	0.003623	0.007217
Random Forest	0.988713	0.003521	0.007017
XGBoost	0.981941	0.003594	0.007163
SVM	0.848758	0.003861	0.007688

j. Linear Discriminant Analysis

The general linear discriminant analysis approach is very similar to a principal component analysis, but in addition to finding the component axes that maximize the variance of our data, which is used by PCA, we are additionally interested in the axes that maximize the separation between multiple classes.

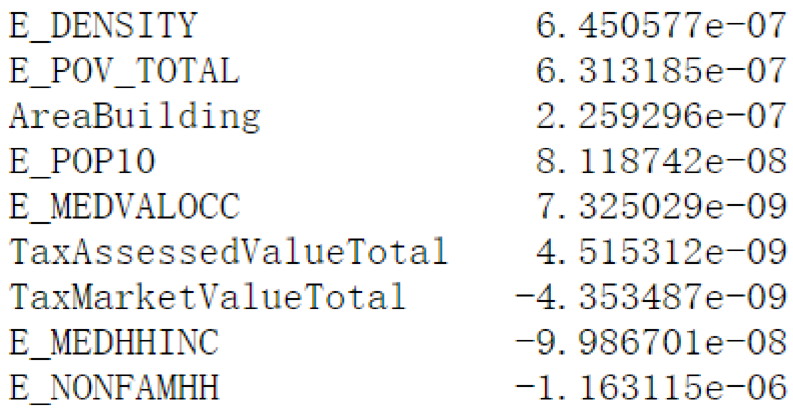
	Recall Score	Precision Score	F1 Score
Linear	0.810384	0.003257	0.006488
Logistic	0.828442	0.003271	0.006517
Random Forest	**0.805869**	**0.003363**	**0.006697**
XGBoost	0.898420	0.003226	0.006430
SVM	0.905192	0.003239	0.006454

Recommendations

	Model	Recall Score	Precision Score	F1 Score
Forward Stepwise	SVM	0.740406	0.004520	0.008986
Backward Stepwise	Random Forest	0.932280	0.003329	0.006635
Hybrid Stepwise	Linear	0.643340	0.006162	0.007458
Ridge	SVM	0.444695	0.007997	0.015712

Lasso	Linear	0.686230	0.006299	0.012485
Elastic Net	**Linear**	**0.519187**	**0.010297**	**0.020194**
Principal Component	Linear	0.643340	0.006162	0.012207
Linear Discriminate	Random Forest	0.805869	0.003363	0.006697

Using features selected by the Elastic Net in the Linear Regression model generates the highest F1 score. Thus, we recommend Spruce to focus on the variables selected by the Elastic Net technique. Thoughtful research and analysis demonstrate the potential reasons for these variables causing title defects. The selected nine variables are as follows:



```
E_DENSITY                 6.450577e-07
E_POV_TOTAL               6.313185e-07
AreaBuilding              2.259296e-07
E_POP10                   8.118742e-08
E_MEDVALOCC               7.325029e-09
TaxAssessedValueTotal     4.515312e-09
TaxMarketValueTotal      -4.353487e-09
E_MEDHHINC               -9.986701e-08
E_NONFAMHH               -1.163115e-06
```

1. E_DENSIT Y - Population density of zip code

Population density has a positive correlation with title defect; the higher the population density, the more likely a real estate would have a title defect. The more concentrated are people living in one area, the more likely there are highrises with larger number of units that accommodate more people. More housing units translate to a larger absolute number of defective real estates given even probability for defective titles. However, there could be other explanations, such as the more people living in a unit of accommodation and the more residents involved in a real estate, the higher the likelihood of title disputes. Spruce should explore the potential reasons behind the correlation between population density and title defect, and divide this feature into more specific and nuanced features to test their correlation with title defect.

2. E_POV_TOTAL - Population in poverty, total

Population in poverty has a positive correlation with title defect; the greater population in poverty, the more likely a real estate would have a title defect. A potential reason could be that more people cannot afford clerks or agents to close a real estate transaction, and are more likely to make clerical mistakes in recording property information. Spruce should explore the reasons behind more people in poverty contributing to higher likelihood of discovering a title defect.

3. E_POP10 - Population of 4/1/2010

Population in general positively correlates to title defects could be explained by the fact that the more people there are, the more demands there are for real estates and more instances of real estate transactions. More real estate transactions could lead to a higher likelihood of discovering a title defect associated with real estate. This feature alone does not tell much about what specific features of the population lead to a higher likelihood of title defect; therefore, Spruce should explore other features on top of the population alone.

4. E_NONFAMHH - Non-Family Households

E_NONFAMHH, which stands for non-family households, has a negative correlation with title defects. Families are less likely to be involved in a title defect real estate or forge property information given more burden placed on the entire family in a title dispute or legal risk. Single individuals have less family-related responsibilities and experience less burden brought upon them by title disputes or legal risks, and are more likely to take on risks and undertake transactions with less critical attention to title risks. Spruce can also explore job title, job tenure and other factors that influence a person's likelihood to be less critical to risk given the relative burden of taking risks.

5. AreaBuilding - Living square feet of all structures on the property

Area Building has a positive correlation with title defects. The larger the living square the property has, the more likely a real estate would have a title defect. One possible reason to explain this variable is that only large living areas of property brings more profits, and it involves more people and more business.

Thus, it would have a higher title contest regarding the ownership of the property.

6. TaxAssessedValueTotal - Total assessed value

Total Tax Assessed Value has a positive correlation with the title defect. The higher the total assessed value has, the more likely a real estate would have a title defect. Similarly, the higher assessed value would attract more people to take advantage of the property, and it may cause the difficulty to sell the property with a higher likelihood of title defect.

7. TaxMarketValueTotal - Total market value

Total market value has a negative correlation with title defect; the lower the total market value, the more likely a real estate would have a title defect. One possible reason is that people always want to overprice the market value so that they can sell their property. And there are other factors to influence the market value so it may cause the title defect.

8. E_MEDHHINC - Median household income

Median household income has a negative correlation with title defect; the lower the median household income, the more likely a real estate would have a title defect. One reason is that people with lower incomes don't have enough money to pay the tax of their house. Without paying tax, it may cause title defects.

9. **E_MEDVALOCC - Housing, Median Value Owner Households**

Median value households have a negative correlation with title defect; the lower the median value of owner households, the more likely a real estate would have a title defect. The reason could be people with lower income do not have sufficient knowledge about real estate or they are not willing to have a lawyer to help them pay those fees, taxes, liens and other issues. Thus, it will cause title defects.

Value Added to Spruce

- Spruce may assign heavier weight on the nine variables we recommend in its current prediction model to see improvements in its existing models.
- These features, on a high level, help Spruce narrow down the scope of which categories of factors to focus on when exploring more features that explain higher likelihood of title defects. For instance, our insight that higher population density leads to greater likelihood of title defects, should inspire Spruce to further explore features within the population density category by breaking it down into more specific levels of features. For instance, Spruce can explore the features “number of residents within an accommodation” and “number of accommodations within a specified area of land” to better understand why and how population density correlates with title defects.

Citation

- McCombe, Madeline. "Intro to Feature Selection Methods for Data Science." *Medium*, Towards Data Science, June 7, 2019, towardsdatascience.com/intro-to-feature-selection-methods-for-data-science-4cae2178a00a.
- Kyoosik Kim. "Ridge Regression for Better Usage". Jan 2, 2019. https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db
- Natasha Sharma. "Ways to Detect and Remove the Outliers". May 22, 2018. https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba
- Will Badr. "6 Different Ways to Compensate for Missing Values In a Dataset (Data Imputation with examples)". Jan 5, 2019. https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779
- Max Kuhn and Kjell Johnson. "Feature Engineering and Selection". June 21, 2019. http://www.feat.engineering/
- Sebastian Raschka. "Implementing a Principal Component Analysis (PCA) – in Python, Step by Step". Apr 13, 2014. http://sebastianraschka.com/Articles/2014_pca_step_by_step.html
- Sebastian Raschka. "Linear Discriminant Analysis – Bit by Bit". Aug 3, 2014. https://sebastianraschka.com/Articles/2014_python_lda.html