Case Study 5: SVM / SGD

Tim Cabaza

July 8, 2024

**Abstract**

Cybe security, big company with firewall interactions.

build a program about whether or nor to accept or deny access – automate it, takes up a lot of manpower and resources. they have historical data about how they have chosen to accept and deny the requests. auto accept or auto deny the request, accurate and function at speech, 60k rows with data to play with – accuracy and performance to make decision about whether or not this can be implemented.

Action is the column we are trying to predict. a port is an address that allows different points of connection, no value other than an address, like last 4 digits of your telephone number.

The case study is an exercise in use RandomizedSearchCV to compare and beat the results of a tuned-random forest model versus a tuned XGboost Model. The goal of the classifier is to predict a binary target (1/0) of whether a business will go bankrupt to inform business decisions on divesting from said companies. The data provided is a 5 year period from 5 different business financial data and the stakeholder believes that a classifier model can accurately provide insight into future bankruptcies. Note this is not Time Series data. The following is a list of steps that are covered in the case study: Exploratory Data Analysis, Data Cleansing, Label Encoding, Normalizing, RandomForestClassifer, XGBOOST, Randomized Classifier Cross validation.

**1 Introduction**

The financial data provided is over a 5 year period from 5 different businesses.

**Table 1: Data**

|  |  |
| --- | --- |
| **Rows** | 43,405 |
| **Columns** | 65 |

**2 Methods**

**Exploratory Data Analysis:** The data had no missing values but given the nature of some of the multi-class categorical features with high numbers of distinct values encoding and consolidating of classes of variables was required. The dimensionality and time to train/test model would increase significantly and most likely crash due to lack of memory as SVM stores the dot product for each point. Consolidating the high cardinality of variables where appropriate was important to stay within memory constraints and still produce a high performing model to show the stakeholder proof of concept.

**Table 2:** Data

|  |
| --- |
| **Source Port:** Categorical in nature but has 22k distinct values and vast majority were "other" category and everything else was 1% or less. To prevent the data from getting to wide the column was categorized into two categories Rare and other and then one-hot-encoded. A similar approach is followed for the subsequent categorical variables where applicable.  A screenshot of a white background  Description automatically generated  **Figure 1&2:** Before ~22k distinct values.    **Figure 2&3:** After transformation, the data had only two categories. |
| **Destination Port:**    **Figure 3&4:** Before ~3k distinct values.    **Figure 5&6:** After transformation, the data had only five categories. |
| **NAT Source Port:**    **Figure 7&8:** Before ~29k distinct values.    **Figure 9&10:** After transformation, the data had only two categories. |
| **NAT Destination Port:**    **Figure 11&12:** Before ~29k distinct values.    **Figure 13&14:** After transformation, the data had only two categories. |
| **\*Target\* Action:**    **Figure 15&16:** The target variable was left at 4 distinct categories, but reset-both as so rare that it can essentially be ignored when looking at the results of any model. |

Columns Bytes, Bytes Sent, Bytes Received, Packets, Elapsed Time, Packet Sent, Packet Received were continuous in nature. Bytes and Packets were highly correlated. The target variable was dropped and the features were normalized using Scikit Learns Standard Scaler.

**Table 3:** Data to use for modeling

|  |  |
| --- | --- |
| **Data** | **Shape** |
| Target (y) | 65532, 1 |
| Features (X) | 65532, 21 |

The data is pushing the limits of SVM modeling capabilities and there is no cross-validation ability built into the SVM or SGD models. To find the optimal hyperparameters and configuration settings for each model the smaller dataset was used (~13k samples) using a stratified version of the original dataset. For final modeling and scoring assessment the full-dataset was used for both model types.

For SVM hyperparameter tuning and configuration, Scikit Learn’s LinearSVC was used to loop over a space for the regularization parameter strength (C). The rest of the settings were configured in sensible manner that addressed the muli-class target variable the appropriate loss and penalty type.

**Table 4: LinearSVC Test / Train Split**

|  |  |
| --- | --- |
| **Parameter** | **Values/Selection** |
| C | 1000 |
| Penalty | L2 |
| Loss | Hinge |
| Dual | Auto |
| Multi-class | Ovr |
|  |  |

The regular SVC package was also used to see if a better model can be assessed for tuning. However this exercise between LinearSVC and regular SVC illustrated the time difference between the two.

**Table 7: SVC**

|  |  |
| --- | --- |
| **Parameter** | **Values/Selection** |
| Kernel | rbf |
| C | 1000 |
| Decision Function Shape | Ovr |
| Gamma | Scale |

Once the optimal parameters were found via a 5 fold cross-val accuracy score, the tuned regularization parameter was used and few more configurations were explored.

For SGD, Scikit Learn’s SGDClassifier was utilized. For hyperparameter tuning a train/test split was used to tuning and configuration.

**Table 8: SGDClassifier**

|  |  |
| --- | --- |
| **Parameters** | **Values/Selection** |
| Kernel | rbf |
| Alpha | 1E-8 |
| Loss | Hinge |
| Penalty | L2 |
| Learning Rate | Optimal |
| Max Iterations | 1000 |
| N\_jobs | -1 |
| Early Stopping | True |
| Shuffle | True |

The final model configurations (LinearSVC, SVC, SGDClassifier) were then run on the entire dataset and the cross-val accuracy score collected.

**3 Results**

The results were

**Table 9**

**LinearSVC Final Model on Full Dataset**

|  |  |
| --- | --- |
| Model Parameters | C=1000, penalty='l2', loss='hinge', dual='auto', multi\_class='ovr', random\_state=42, max\_iter=1000 |
| Classification Report |  |
| Confusion Matrix |  |

**Table 10**

**SVC Final Model on Full Dataset**

|  |  |
| --- | --- |
| Model Parameters | C=1000, decision\_function\_shape='ovr', gamma='scale', kernel='rbf', random\_state=42 |
| Classification Report |  |
| Confusion Matrix |  |

**Table 11**

**SGDClassifier Final Model Full-Dataset**

|  |  |
| --- | --- |
| Model Parameters: | loss='hinge', penalty='l2', alpha=1E-8, learning\_rate='optimal', shuffle=True, random\_state=42, max\_iter=1000, n\_jobs=-1, early\_stopping=True, validation\_fraction=0.1 |
| Classification Report: |  |
| Confusion Matrix |  |

**4 Conclusions**

The case study assessed a financial datasets of five different countries over a five year period. A Random Forest model and XGboost models were created with the tuned XGboost model outperforming the rest with an AUC score 0.99.

Overall both the

**Appendix: Code**