Name: Vindhya H V

Email address: vindhyavijaya98@gmail.com

Contact number: 9611291530

Anydesk address:

Years of Work Experience: 2 years 6 months

Date: 13/01/2022

Self Case Study -1: IEEE-CIS Fraud Detection

Can you detect fraud from customer transactions?

"After you have completed the document, please submit it in the classroom in the pdf format."

Please check this video before you get started:
https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg

Overview

*** Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** ***

1. In every seed of good there is a piece of bad, Now a day as the world is getting digitalized and moving towards Online Payment Systems, Fraudulent people are stepping the line and trying to make money by cheating innocent customers.

2. This case study is a small step towards helping millions of customers and saving them from fraudulent transactions. IEEE-CIS is partnering with world's leading payment service company, Vesta Corporation in providing the best solution for fraud prevention industry.

3. Here we will benchmark machine learning models on a challenging large-scale dataset provided by Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features.
4. If successful, this will improve the efficacy of fraudulent transaction alerts for millions of people around the world, helping hundreds of thousands of businesses reduce their fraud loss and increase their revenue. In this case study we will be predicting the probability that an online transaction is fraudulent, as denoted by the binary target isFraud.
5. In this case study, both predicting a fraud transaction as legit and legit transaction as fraudulent will impact a lot. So we can say that both true positive and false positives are important. We should be able to predict the probability of fraud transaction.
6. The datasets consists of –
    a. test_identity.csv and train_identity.csv – This set of data has categorical and binary features such as device type , device information , and few more features whose actual meanings are masked for security reason. isFraud column is the target column which has binary value 0 or 1 . 1 implied fraudulent transactions.
    b. test_transaction.csv and train_transation.csv – This set of data has categorical features such as ProductCD, addresses, email domains of purchaser and receiver, transaction datetime, transaction amount and few more features whose actual meanings are masked for security reason
    c. Both identity and transaction data have transactionID column in common , which can be used to join both the tables
7. We can easily guess that the number of fraudulent transactions will be very less compared to legit ones, and hence the data is heavily imbalanced. In this scenario AUC as a performance metric will give good result.

Research-Papers/Solutions/Architectures/Kernels

*** Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain

with your own diagrams. it is mandatory to write a brief description about that paper. Without understanding of the resource please don't mention it***

1. https://medium.com/@mr.priyankmishra/a-realistic-approach-to-ieee-cis-fraud-detection-25faea54137

   In this solution presented by Priyank Mishra, below are the key points about this solution

   - Author has identified a pattern of missing data in v_features, where subgroups of V_features having the same number of missing values were present
   - Transactions done for ProductCD "C" had the highest chance for being fraudulent as compared to other ProductCD categories.
   - C_features were highly inter-correlated
   - Both tree based and non-tree based algorithms are carried out on this dataset, and found out that GBDT works better than all.
   - Here author has taken 3 types of dataset to experiment, one having missing values imputed using -999 and normalized, second one imputed without normalization and third one was left untouched.
   - After obtaining AUC score from the base model, hyper parameter tuning is done to improve the model score. After hyper parameter tuning, AUC score improved by approximately 0.23.
   - To further improve the score, Data cleaning is done. Here removing redundant features, collinear features, Time Inconsistent features is done and the number of features reduced from 433 to 159.
   - In last part of the solution, feature engineering is done, Here total of 153 extra features are added such as transaction day, log of transaction amount, Device Version, interaction feature etc..
   - After final prediction model score increased from 0.923 to 0.938.
   - Mainly focus of this solution was to avoid data leakage.
2. https://www.kaggle.com/c/ieee-fraud-detection/discussion/108575

This topic is authored by First place holder of the Kaggle Competition Mr. Chris Deotte. Here Chris talks about various feature engineering techniques which he has implemented in his code.

- Imputing NAN values with a non- NAN value in features
- Label Encoding – Conversion of Non Integer values to integer values
- Memory reduction – To reduce the memory float64 are converted to float32 and int64 to int32.
- Splitting columns to increase the number of features and combining certain columns based on their correlation with the target
- Frequency encoding is used to check the frequency of column values
- Aggregation , Normalization / standardization are other important feature engineering techniques.

3. https://towardsdatascience.com/ieee-cis-fraud-detection-top-5-solution-5488fc66e95f

   In this solution by Arun Mohan , Below are the key takeaways

   - Since the data is heavily imbalanced , AUC is used as a performance metrics because accuracy will not be suitable in this case
   - Using Transaction data author is checking whether the train and test data are in Time split format. ie train data should be from earlier period and test data should be from a later period of time.
   - Here outliers from transaction amount are found out and removed to make the model more stable
   - Domain protonmail.com has 90% fraud transactions, which is an interesting finding.
   - In this solution 3 base models are used, Here are the results after hyper parameter tuning them

     Logistic regression: 0.84018(Train AUC), 0.84245(Test AUC)

     Random Forest: 0.9030(Train AUC), 0.8600(Test AUC)

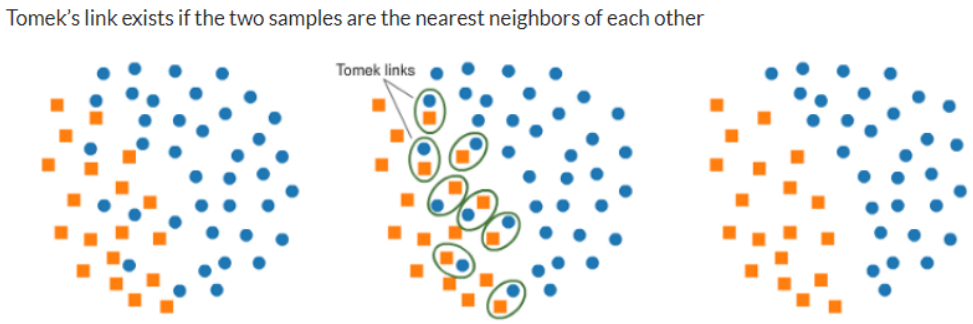
     Xgboost: 0.994(Train AUC), 0.9234(Test AUC)

Here Xgboost has been chosen as the base model.

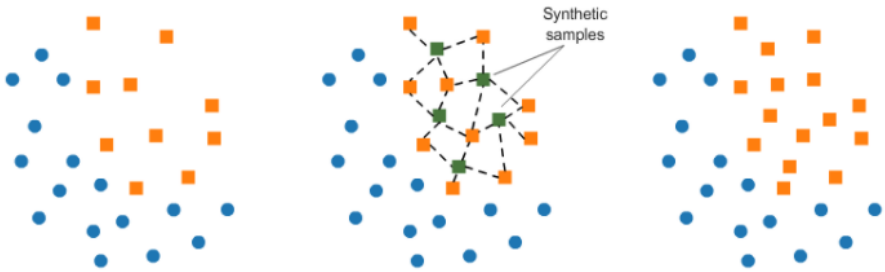
- Below is the scores author has obtained after doing all the trial and errors

Feature engineering	Local CV	Public LB	Private LB
Baseline model (xgboost)	0.923433	0.9238	0.90047
Baseline + Reduced v cols	0.923162	0.922486	0.89976
After normalizing some D cols	0.9327	0.930795	0.90718
Handmade features + encoding features	0.93643	0.93427	0.909344
UID1 aggregations	0.947	0.94231	0.91588
UID2 aggregations	0.9481	0.940167	0.917
Hyperparametertuning using randomized search + groupcross validation	0.95126	0.954825	0.928873

4. https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/
   Below are some of the tricks used to improve efficiency of models with imbalanced dataset.
   - Resampling technique – Resampling is a technique where we try to match the proportion of classes in an imbalanced dataset by either down sampling or up sampling. In down sampling, majority class samples are removed where as in up sampling , samples are added to minority class
   - Under sampling: Tomek links – Tomek links are pair of very close instances but of opposite classes. Here instances of majority class of each pair are removed to increase to increase space between each models



- Synthetic Minority Over-sampling Technique(SMOTE) –In this technique a random point from minority class is picked and its k-nearest neighbours are identified. Synthetic points are added between the chosen point and its nearest neighbours



5. https://datachannel.co/blogs/introduction-to-automated-feature-engineering-using-deep-feature-synthesis/#:~:text=Deep%20Feature%20Synthesis%20is%20an,new%20groups%20with%20better%20features.

- Here author explains about Deep Feature Synthesis (DFS) algorithm. Deep Feature Synthesis is an algorithm that creates features between sets of relational data to automate the machine learning process
- Feature Tools is a framework that is used for performing automated feature engineering. It transforms the relational and transactional databases into feature metrics to make your data ready for machine learning. Deep feature synthesis by bringing together multiple features, boosts the working of feature tools

---

First Cut Approach

*** Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** ***

*** When you are doing the basic EDA and building the First Cut Approach you should not refer any blogs or papers ***

1. Initially I would like to clean the data and do an EDA to check the pattern of data and its features. I would like to know how much each individual set of features are contributing in identifying the fraudulent transactions.
2. I will use both tree based and non-tree models and pick the one which performs better. Since I dint find any references which has used stacking to solve this problem , I will try if it works better or not
3. After picking the best among all the tried models, I will do hyper parameter tuning to see if there is any improvements in the model improvement
4. Feature engineering is very important and interesting part of any ML model, I will try to create as many features as possible and do forward feature selection.
5. I find Deep Feature Synthesis technique interesting, I will try to implement it my model to check if it adds some weightage
6. Finally I will do some trial and error by removing some not so important features and check the performance of the model
7. As per my understanding, hyper parameter tuning on the final model might increase the model performance.
8. At last I will display the results in a pretty table or using visualization technique

---

Notes when you build your final notebook:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar
2. You should not read train data files
3. The function1 takes only one argument "X" (a single data points i.e 1*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
    a. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
    b. so in your final notebook, you need to pass only those two values
    c. def final(X):
        preprocess data i.e data cleaning, filling missing values etc
        compute features based on this X
        use pre trained model
        return predicted outputs
    final([time, location])
    d. in the instructions, we have mentioned two functions one with original values and one without it
    e. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
    f. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y_predict)
4. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
5. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
6. Check this live session: https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models