M. M. Wijesundara- IT19980782

ID : IT19980782

Email : [it19980782@my.sliit.lk](mailto:it19980782@my.sliit.lk) / maneka611997@gmail.com

Link to dataset : <https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset?datasetId=2093649>

Link to public GitHub repository :

Link to Youtube video :

|  |
| --- |
| **Report:IT19980782**  The problem I am addressing is “Classifying a transaction as fraud or not using Support Vector Machine”.  We got the dataset from [www.kaggle.com.The](http://www.kaggle.com.The) dataset provides 10 columns.According to the website the description of the dataset is as follows:   1. step: represents a unit of time where 1 step equals 1 hour 2. type: type of online transaction 3. amount: the amount of the transaction 4. nameOrig: customer starting the transaction 5. oldbalanceOrg: balance before the transaction 6. newbalanceOrig: balance after the transaction 7. nameDest: recipient of the transaction 8. oldbalanceDest: initial balance of recipient before the transaction 9. newbalanceDest: the new balance of recipient after the transaction 10. isFraud: fraud transaction  * First, I installed pydataset * Then I imported all the necessary modules * Then I loaded the dataset     Here the “Type” is a categorical variable.   * I did data preprocessing: * Dropped columns with null values * Converted dummy variables to integer values(Here, the dummy variable is “type”).The result will be like something shown in the screenshot below:   IT19980782    Here the values “cash in”,”cash out”,”debit”,”payment”,”transfer” are converted into columns with numerical values.SVM requires these parameters to be integer values.  I converted the dummy variables(categorical independent variables used in regression analysis) to integer values using the below code.     * Then I concatenated the converted columns with the original dataset.(So, now there are 16 columns in total)      * Then I dropped the categorical columns which are now of no use.      * I did feature scaling as another data pre processing stape.Here I bought all values to a number between 0 and 1.     Here you can see oldbalanceOrg,newbalanceOrig,oldbalanceDest and newbalanceDest now have values between 0 and 1 unlike the original dataset.   * Also you can see that I have removed useless features like nameOrig and nameDest from the dataset,since I think that they are not relevant features when trying to classify whether a transaction is fraudulent or nor.Obviously,we cant say whether a transaction can be a fraud one simply by looking at the name.   IT19980782   * Then I separated the dataset into independent(X) and dependent(y) variables. * I splitted the data into training and testing sets.I gave 30% for testing data from the original dataset.     Here, the random\_state=1.When the random\_state is set to 1 the algorithm will always produce the same results when running with the same data and settings.   * Then I created an instance of the svm classifier.     Here, C is a regularization parameter.The larger the C ,higher the training accuracy .Then ,there will not be a smooth decision boundary.   * I fitted the model and got an accuracy of 99.89% which is an incredible outcome. * Then I made a prediction using the test data. * Then I created a crosstab.It allows us to see the strength of the model related to various techniques of classification.     There are 1906317 true negatives and 526 true positives.Since the diagonal values are high,the model is accurately predicting true values.There are 1 false positive and 1942 false negative.  Reference:  <https://medium.com/@kennymiyasato/classification-report-precision-recall-f1-score-accuracy-16a245a437a5>   * Then I printed a classification report using svm.     The model is evaluated on a dataset of 1908786 instances. The "support" column indicates the number of instances in each class.   * (Precision: what percentage of your predictions were correct?)Here precision says the percentage of correct negative predictions related to total negative predictions.All the * instances that the classifier predicted as negative are actually negative.All the instances that the classifier predicted as positive are actually positive.100% percentage of negative predictions were correct and 100% of positive predictions were correct.   IT19980782   * Recall shows the percentage of correct negative predictions related to total actual negatives and percentage of correct positive predictions out of all positive instances. Since it is 1,the classifier correctly identified all the negative values.The recall is 1.0 for negative class .This means that,out of all negative instances of the dataset,the model was able to correctly identify 100% of them.The recall is 0.21 for the positive class.This means that,out of all positive instances of the dataset,the model was able to correctly identify 21% of them. * Here there is no 1 and has only 0 because,there are no positive fraudulent transactions in our testing data. * F1 score depicts the harmonic mean of precision and recall.The closer to 1,the better the model.It is 1 for the negative class.So, perfect performance in correctly identifying negative instances and making no false predictions.It is 0.35 for the posiive class.So there is a poor performance in correctly identifying positive instances. * Support is 1906318 for negative class and 2468 for positive class. * Macro average : all classes equally contribute * Weighted average : it is affected by the size of the classes.   Reference:  <https://medium.com/@kohlishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2ce397>   * Then I coded to make some predictions with new or user generated values.First,I created 5 records as follows:     One thing to notice here is that ,we need to have more than 1 record,and, we need to have at least 2 different values for each column.Otherwise we might end up with ‘NaN’ results for column values after min-max normalization because if min and max are the same value,the difference of them will be 0.  Then I dropped the null values.I also dropped the unwanted features like ‘nameOrig’ and ‘nameDest’.Then I did min-max normalization to bring all values to a value between 0 and 1.Then finally I used my svm model to generate the predictions as follows:    Prediction for record 1: 1 (is fraud)  Prediction for record 2: 0 (not fraud)  Prediction for record 3: 0 (not fraud)  Prediction for record 4: 0 (not fraud)  Prediction for record 5: 0 (not fraud)  IT19980782  *Problems encountered :*   1. How to load csv file? 2. What are the meanings of each field? 3. How to convert categorical values ,known as dummy variables, to integer? 4. Unable to allocate 243 MiB for an array with shape[5,6362620] and data type float 64.   Solution: close all other programmes and restart the kernel by removing all outputs.  *Future work:*  Present the app in interviews.  Use it to teach a set of novelty students  Use the experience obtained in the final year Research Project  *Algorithm used:SVM*  Support Vector Machine is a supervised learning algorithm.It can be used in both regression and classification tasks.In this scenario I used it as a classification model to classify transactions as “Fraud” and “not fraud”.  The intuition of support vector machines is to find a hyperplane with the largest distance between the negative and positive margins as shown below.    SVM is called a large margin classifier as it tries to maximize the margin.  There can be datasets with noise.That means datasets with outliers.Hard margin require all data points to be classified correctly and soft margins will give some space for errors.In overfitting,the hyperplane perfectly classifies data pointsinto + and – samples.But test data might perform with errors.In underfitting both training and testing accuracy is low.  In soft margin classification some + samples maybe in – side,while some – samples will be in + side.Slack variables can be added to allow misclassification of difficult or noisy samples.By changing the hyperparameter C , we can adjust for overfitting and underfitting.With this parameter the hypothesis will also change.  In svm the most important points are the support vectors.  If we cannot draw a straight line that can classify the data points,we can project these points to a higher dimensional feature space.We can use kernels to make non seperable paroblems serperable.There are various types of kernel functions.Linear,Polynomial,Gaussian,Sigmoid,String and Chi Square are some.  SVM parameters are c and lambda.Hih c favors overfitting while high lambda favours underfitting.  SVM ‘s can work with large datasets.There are various applications of SVM s like text and hypertext categorization,image classification,protein and cancer classification,hand written character recognition.  IT19980782  *How can we increase the accuracy of our model?*   * Increase the number of samples in the dataset * Treat missing and outlier values * Convert features into new features * Choosing the best festures * Use many algorithms * Use hyperparameters for algorithmic tuning(like I have used C in above) * Combine results of multiple weak models commonly known as ensemble method. * Cross validation.This is done in the middle of traing and finally testing them model * Collect more positive instances * Adjust hyperparameters |