Introduction to Machine Learning (SS 2022)
Programming Project

Author 1
Last name: Seok
First name: Chaewon
Matrikel Nr.: 12116473

Author 2
Last name: Vögele
First name: Marina
Matrikel Nr.: 01623436

Author 3
Last name: Lintner
First name: Christina
Matrikel Nr.: 12018499

I. Introduction

The goal of our project was to use machine learning for binary classification of financial transactions. These transactions are classified as either fraudulent (class 1) or not fraudulent (class 0). The provided data set consists of 227845 data points, 394 of which are categorized as fraud. Each data point consists of 30 numerical features, which include time, amount, and 28 unspecified properties. There is no missing data, but the data set is imbalanced since there are a lot more regular transactions than fraudulent ones.

II. Implementation / ML Process

For the financial transaction problem three different models were tested: the Logistic Regression model, the Gaussian Naive Bayes model and the K-Nearest Neighbors model. Our selected model for the financial transactions project is the Logistic Regression model. In the following section we are going to describe all three models.

A. Model 1 - Logistic Regression:

Logistic regression is besides linear regression a part of the regression family and belongs to the machine learning paradigm of supervised learning. In regression the relationship between the independent features and the depending outcome is analysed. The goal is to train the model so that it understands the relation between features and corresponding outcome and it can predict the correct outcome just by using the features.

Logistic regression uses the sigmoid function to determine the relationship between features and outcome. It is used for classification and can be used in cases where the resulting outcome only has two possible values. In the case of financial transactions the two available outcomes are fraudulent or not fraudulent transactions, where fraudulent has the value 1 and not fraudulent is symbolised with 0. Which means the model is a suitable classifier for binary classification problems. Logistic regression also can determine the probability of the occurrence of an outcome. Since the model is working with probabilities the outcome is between 0 and 1, where 1 implies a 100% chance of fraud and 0 a not fraudulent transaction. In the implementation of the project the sklearn logistic regresion model and the included functions for trainung, predicting and evaluating the model where used.

Data Preparation

At the beginning the transaction data set has a total of 28 features form ‘Feature0’ to ’Feature27’, without the columns ‘Amount’ and ‘Time’. The decision boundary in logistic regression is a hyperplane in a space with the dimension of the number of features. Using all 28 features means the resulting hyperplane is 27 dimensional in a 28 dimensional space. This will not only make the representation of the data difficult but it can also result in overfitting and have a negative impact on new unseen data. To also avoid the curse of dimensionality the features are going to be reduced for the final model. For the reduction of the features three different approaches were tested.

For each approach the logistic regression model was trained with the resulting features on a subset of the finance transaction data set. A different subset of the dataset was used for testing the accuracy. The test data set consist of 68354 transactions of which 112 are fraudulent transactions and 68242 are not fraudulent.

The first approach was simply selecting features, which separate the data well into the two classes. Based on plots of different combination of the features the features 9, 12, 13, and 16 were selected. The logistic regression model was evaluated with the selected features and 76 of the 112 fraudulent transactions where assigned to the correct class. 5 transactions where wrongly specified as fraud and 36 where wrongly assigned as valide transactions.

The second approach was using principal component analysis (PCA). PCA computes a new set of features with a lesser dimensionality by trying to keep as much information as possible. Five dimensions of features where tested 15, 10, 5, 3 and 2. The problem with PCA is that it does not consider the class, so the results of the reductions with 5, 3 and 2 dimensions where not very accurate. The exact result of each dimension is listed in the table (Figure 1). Using this approach as final implementation would mean we still have to work with at least 10 features to get a good prediction for fraudulent transactions. The first approach with only 4 features would be more suitable.

The last approach was using linear discriminant analysis (LDA). In comparison to PCA, LDA takes the class into consideration and extracts the features to the number of classes - 1. In the case of the binary problem of the financial transaction task, this reduction results in one feature. Training and testing the model with the one feature resulting from LDA is better than the result of the PCA approach. Out of the 112 fraudulent transactions 73 have been predicted correctly. Comparing all results of the three attempts the PCA is the worst. It needs more features to acquire the same accuracy as the two others. The first approach with selecting four suitable features is slightly more accurate than the linear discriminate analysis. The first approach will be used for reducing the dimension and the features.

	True Fraudulent	False Fraudulent	True Not Fraudulent	False Not Fraudulent	Accuracy
Feature Selection	76.0	5.0	68237.0	36.0	0.9994001814085496
PCA 15 dim	80.0	9.0	68233.0	32.0	0.9994001814085496
PCA 10 dim	78.0	8.0	68234.0	34.0	0.9993855516868069
PCA 5 dim	47.0	8.0	68234.0	65.0	0.9989320303127834
PCA 3 dim	21.0	10.0	68232.0	91.0	0.9985223981039088
PCA 2 dim	12.0	4.0	68238.0	100.0	0.9984785089387 6
LDA	73.0	6.0	68236.0	39.0	0.9993416625215789

Fig. 1. Accuray of the logist regression model using three different approcches of feature and dimesnion reduction.

B. Model 2 - Naive Bayes:

Naive Bayes is a common method for binary classification problems, and one of its main advantages is that is relatively quick. It uses Bayes' theorem of posterior probabilities to categorize data points into one of two classes. Specifically, we used the Gaussian Naive Bayes method, which assumes the individual features to follow a normal distribution. Plotting the distribution of the features showed that there is indeed a bell curve for most of them.

According to the data the model is trained on, for each class a base probability is determined, as well as the distribution of each feature's values for that class. The base probabilities can also be set manually, but in our case it made sense to determine them from the training data since the imbalanced data is most likely an accurate representation of the actual distribution. A new data point is then categorized by calculating the posterior probability for each feature, assuming that it belongs to class 0, and then the same process is repeated for class 1. The class with the higher overall probability is then chosen as the model's prediction.

The Naïve Bayes method assumes all of the features to be independent from one another, which is why it is very quick. However, this can also have a negative impact if two features correlate, because in that case, the same probability is basically factored into the decision twice. This could be avoided by identifying and sorting out correlating features during preparation of the data set.

For our experimentation, we used the implementation provided by Scikit-Learn. Training the model on 70% of the provided data set and then using the remaining 30% as test data resulted in an accuracy of 99,38%. More precisely, 426 out of 68354 test data points were mislabelled, 393 of which were false positives and 33 were false negatives. If the model's main use was to identify possibly fraudulent transactions for further investigation, it is relatively accurate since only 33 fraudulent transactions were not categorized correctly. However, if the model was used to provide a final decision about the nature of a transaction, 393 false positives are not ideal at all. As it turned out, the accuracy metric that was used in the training phase is not the most representative for imbalanced data sets like the one we worked with, so the final score (where ROC AUC score was used) turned out to be significantly lower.

To possibly improve the accuracy of the predictions, we also tried training the Naive Bayes model using 10-fold cross validation. This raised the overall accuracy to 99,89%. Only 78 out of the 68354 test data points were labelled incorrectly, but interestingly, the majority of misclassifications were now false negatives, of which there were 76, and only 2 classifications were false positives. If the goal of the machine learning model was to find as many fraudulent transactions as possible, the cross validation therefore decreases the accuracy in that regard significantly. This is also reflected in the ROC AUC score, which was only 64,15% as opposed to 82,43% for the initial model. To make that result more tangible, the cross validation model only found 30 out of 106 fraudulent transactions in the test data, while the initial model found 73.

C. Model 3 - K-Nearest Neighbor:

- Data Pre-processing: Extracted 'Class' attribute from data and set that into y, and the rest was made X. Then I split the data using 'train_test_split' method into 90% training data and 10% test data. This was initially 20% test data, but was adjusted to prevent overfitting.

- Method: I used K-Nearest Neighbor(KNN) method as my model to classify the dataset. KNN is a non-parametric, supervised method within Density Estimation which is capable of both regression and classification. It uses between-sample geometric distance to classify a sample. When density estimation p(x)=K/nV, KNN uses fixed K and determines V from the data.

- Since KNN algorithm internally calculates the distance between the points, the computation time taken by the algorithm will be more as compared to other algorithms in certain cases. It is advised to use the KNN algorithm for multiclass classification if the number of samples of the data is less than 50.000, which makes it unsuitable for the given dataset.

III. Results

A. Model 1 - Logistic Regression:

In this section we are looking at the performace of the selected logistic regression model on the training and validation sets.

The logistic regression model was trained on the training data set provided on OpenOlat. 159.491 entries reduced to

four features were used for training the model. The remaining 68.354 have been used for testing the model. The resulting accuracy was 0.9994001814085496 have a look at the tabel 1. 76 transactions were correctly classified as fraudulent and 68.237 as not fraudulent, 5 were wrongly assigned as fraudulent and 36 as not fraudulent. Evaluating the model on the validation sets on JupyterHub returns a train dataset score of 0.9437898678773832 and a test dataset score of 0.981221934468779.

Despite of only using four selected features of the initial 30 features, the logistic regression model returns a satisfying prediction accuracy of over 98%.

In the figures 2 and 3 the class predictions of the logistic regression model are compared to the real class the transaction belongs to. Wrongly classified financial transaction are orange, fraudulent transactions are grey and not fraudulent ones are red. Figure 3 displays the selected features 9 and 13 and figure 2 the features 12 and 16.

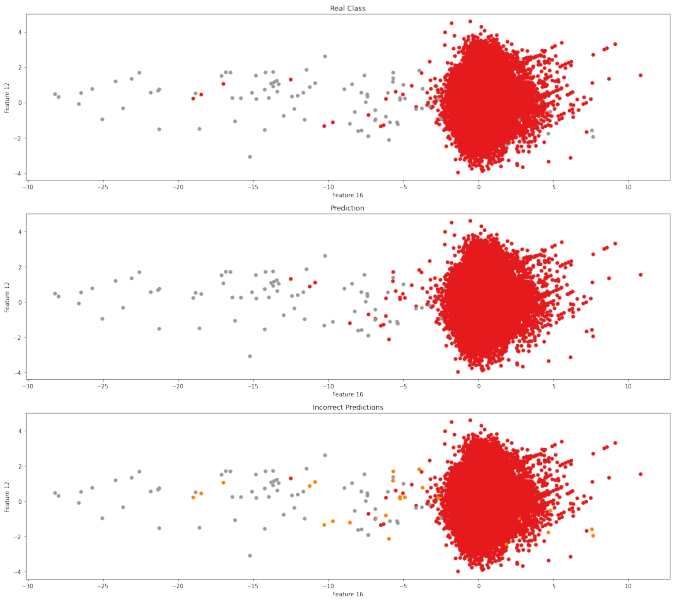

Fig. 2. Results of the logistic regression model displayed with the features 12 and 16.

B. Model 2 - Naive Bayes:

On the final test script the Gaussian Naive Bayes model only reached an accuracy of 82,42% on the train dataset and 79,26% on the test dataset. This difference to the results from the training and testing phase described in section II can be attributed to the different accuracy metrics used. The accuracy metric which was used in the training notebook is biased because of the imbalanced data set. With that many more regular transactions it is easier to get a good accuracy score. This explains why the more balanced ROC AUC score that was used in the test script returned a significantly lower accuracy value. The numbers of misclassification for the initial model and the model which used cross validation are displayed in the figure 4.

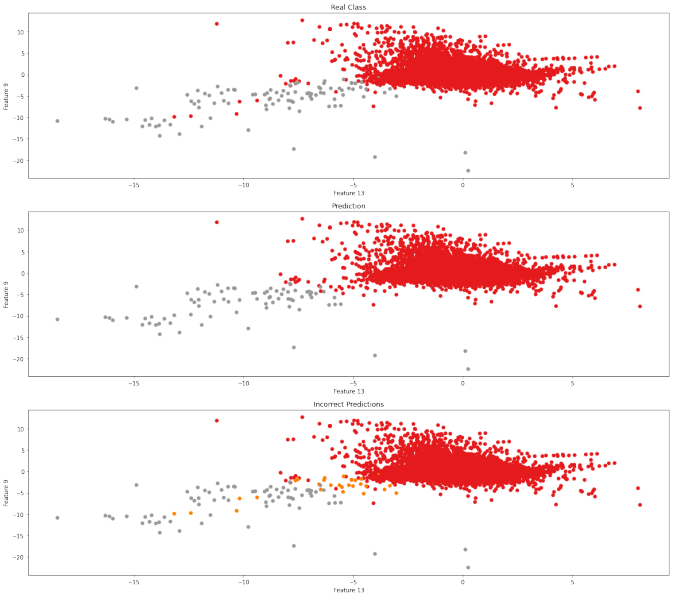

Fig. 3. Results of the logistic regression model displayed with the features 9 and 13.

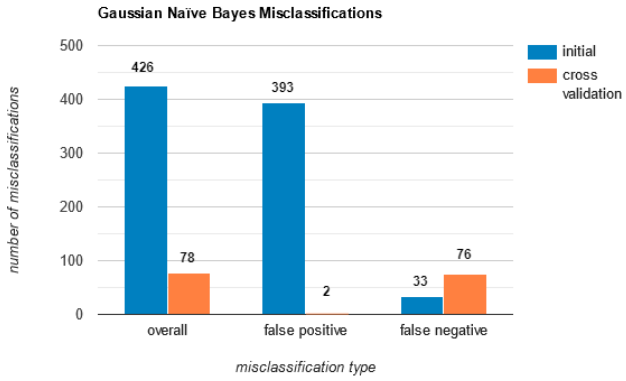

Fig. 4. Comparison of inital naive bayes model and cross validation.

C. Model 3 - K-Nearest Neighbor:

The result is displayed in figures 5 and 6.

IV. Discussion

A. Model 1 - Logistic Regression:

Compared to naive bayes the model has a good accuracy as well in the training phase as in the final validation notebook. The score is always above 94%. By reducing the features and the dimension significantly the model is safe of overfitting as the good result of the final validation proofs. Also the model can be visualised with only using four features and not 30. The visualisation could also be improved by only using 3 features but the accuracy would suffer from that. And as for the task of finding fraudulent financial transaction a high accuracy should be the goal reducing the features further would be not advisable.

As described in detail in II three different approaches for the

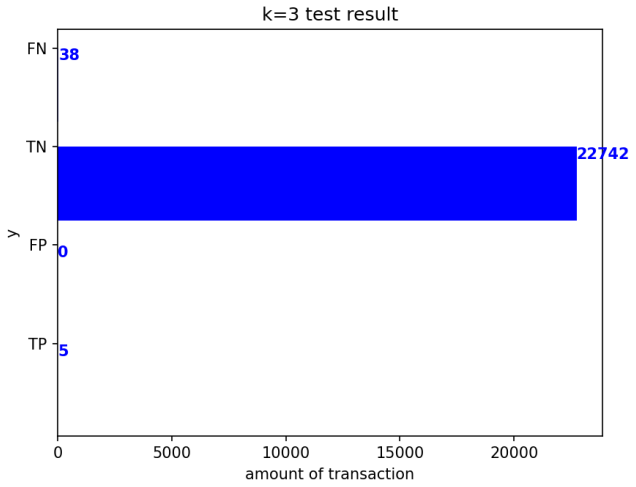

Fig. 5. Result of classification with respect to different k values, k=3

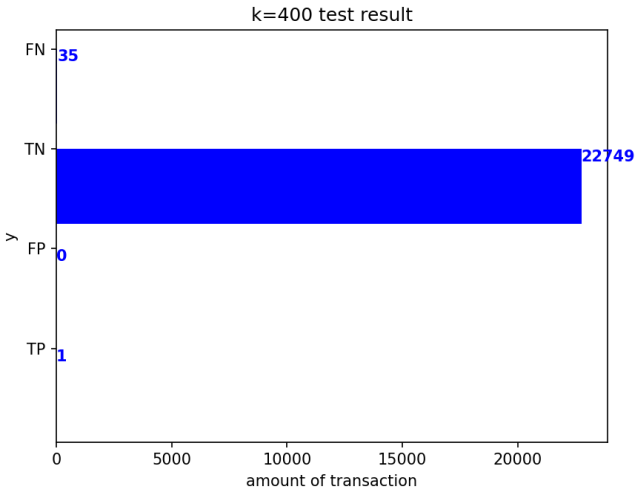

Fig. 6. Result of classification with respect to different k values, k=400

reduction of the features and the dimension of the logistic regression model have been tried. PCA and LDA where tried but not chosen for the final implementation. In the case of PCA too much features where needed to get a satisfying result. LDA had only one remaining feature which made it hard to visualise the data. Furthermore both had a higher rate of misclassifying than the used approach of selecting fore features.

B. Model 2 - Naive Bayes:

Naive Bayes was an interesting choice for us because it assumes independence of the individual features, which is not very likely for the financial transactions of the given project. Accordingly, the results were not as good as for the logistic regression model. One advantage of the Naive Bayes model is that the initial version which did not include cross validation trained really fast, which is what we expected. Overall, cross validation did not seem to improve any aspect of the model. Another advantage of the Naive Bayes model is that it is insensitive to irrelevant features. While it is an efficient and scalable model, the results did not warrant choosing Naive Bayes as our final submission.

C. Model 3 - K-Nearest Neighbor:

In KNN algorithm, choosing the number k is a crucial step. It was stated that the general approach would be setting k as square root of number of samples, so initially I tried to set k as 400, which made the execution time too long and not much difference in test scores from significantly smaller k values. So I reset the k to 3, which showed the best accuracy among k values of 1 to 10. I wanted to visualize the result better with more precise numbers in the y axis, so that comparison between the train/test data would be clear. With more time I would like to take some features from the data and visualize it in a dotted plot, which would show the decision boundary.

D. Discarded Ideas

For the financial transactions project we also considered using decision trees, but these tend to overfit. Since regression can have the same problem, we did not want to choose two models with possible overfitting issues. Our final logistic regression model does not overfit, since we used feature reduction and also the train and test scores are similarly high. We also considered Support Vector Machines but did not choose that model for further experimentation because it takes a long time to train and we were unsure if it would be suited for our data set.

Some things we did not fully consider when initially choosing our models, for example that imbalanced data can have a significant impact on the prediction accuracy e.g. when working with KNN. There are advantages and disadvantages to every binary classification model, and some tradeoff is almost unavoidable. Overall, our goal was to choose three models with different strengths and weaknesses for an interesting comparison.

V. Conclusion

A. Model 1 - Logistic Regression:

For the evaluation test set performance in the transactions notebook on JupiterHub the model was trained locally on the transactions dataset available on OpenOlat and saved to a file. The trained and saved logistic regression model with the use of four features was then loaded in the validation file. The performance was very fast and had a high accuracy. It has a train dataset score of 0.9437898678773832 and a test dataset score of 0.981221934468779.

B. Take-Away:

Choosing the appropriate model is not an easy task. The decision which model to use not only depends on the size of dataset and the size of the features, it also depends on the requirements regarding the result and computation time. For example K-Nearest-Neigbors takes a long time to train. In comparison the Naive Bayes model is faster because the features are evaluated independently.