Machine Learning Models Comparisons

April 2023

1 Linear SVM

1.1 Hyperparameters

1.1.1 Possible Hyperparameters

Parameter	Description	Used	Comment
`penalty`	Specifies the norm used in the penalty term	Yes	
`loss`	Specifies the loss function to use	Yes	
`dual`	Controls the formulation of the optimization problem	No	We have n_samples > n_features
`tol`	Specifies the tolerance for stopping criterion	Yes	
`C`	Controls the penalty for misclassification	Yes	
`fit_intercept`	Controls whether to include the bias term in the model	Yes	
`intercept_scaling`	Scales the bias term	No	Features are scaled using StandardScaler
`max_iter`	Specifies the maximum number of iterations for the solver	No	Excluded for performance reasons
`class_weight`	Balances the classes in the training data	No	We handle class balancing by using undersampling
`multi_class`	Specifies how to handle multi-class classification problems.	No	Our problem is binary classification

Table 1: Usage of LinearSVC hyperparameters in the model, their descriptions, and justification for their inclusion or exclusion.

1.1.2 Comparison of Best Hyperparameters with Different Sample Sizes

Since our dataset is so large, we tuned the hyperparameters of the LinearSVC classifier using a random search on four different sample sizes: 10.000, 100.000, 500.000, and 1.000.000 transactions. For each sample size, we evaluated the performance of the classifier using the AUC-ROC score. From these metrics, we can decide the hyperparameters to use on the full dataset of 6.362.620 transactions.

- 10.000
- 100.000
- 500.000
- 1.000.000

Sample Size	**penalty**	**loss**	**toll**	**C**	**fit_intercept**	**ROC-AUC**
10.000	l2	hinge	0.01	3792.69	false	0.789
100.000	l2	squared_hinge	0.001	545.56	true	0.918
500.000	l2	squared_hinge	0.0001	29.76	true	0.930
1.000.000	l2	hinge	0.01	29.76	true	0.939

Table 2: Comparison of the best found hyperparameters and ROC-AUC score for the LinearSVC model with different sample sizes.

1.2 K-best Feature Selection

Using the hyperparameters found before, we can use K-best feature selection on the entire dataset.

K-best	Accuracy	Precision	Recall	F1 Score	ROC AUC	Selected Features
3	0.5596	0.0029	1.0000	0.0058	0.7795	nameDest
4	0.5596	0.0029	1.0000	0.0058	0.7795	type, nameDest
5	0.5662	0.0030	1.0000	0.0059	0.7828	type, nameDest, oldbalanceDest
6	0.5662	0.0030	1.0000	0.0059	0.7828	type, nameOrig, nameDest, oldbalanceDest
7	0.9393	0.0194	0.9276	0.0380	0.9334	type, amount, nameOrig, nameDest, oldbalanceDest
8	0.9431	0.0205	0.9221	0.0402	0.9326	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest
9	0.9393	0.0194	0.9282	0.0380	0.9337	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest, newbalanceDest
10	0.9394	0.0194	0.9257	0.0379	0.9326	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest, newbalanceDest

Table 3: Comparison of the different metrics on an increasing number of K in K-best feature selection.

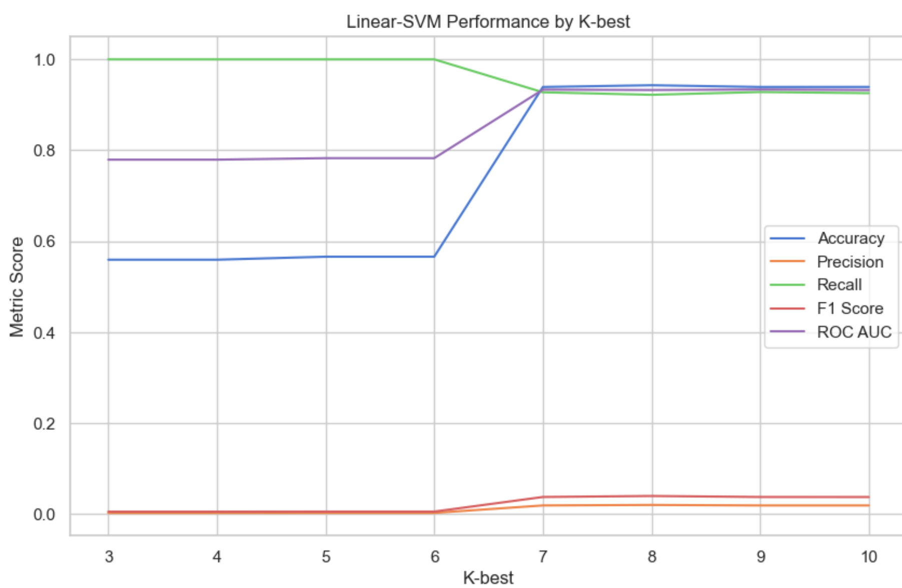

Figure 1: Figure 1: Metrics scores compared to K values

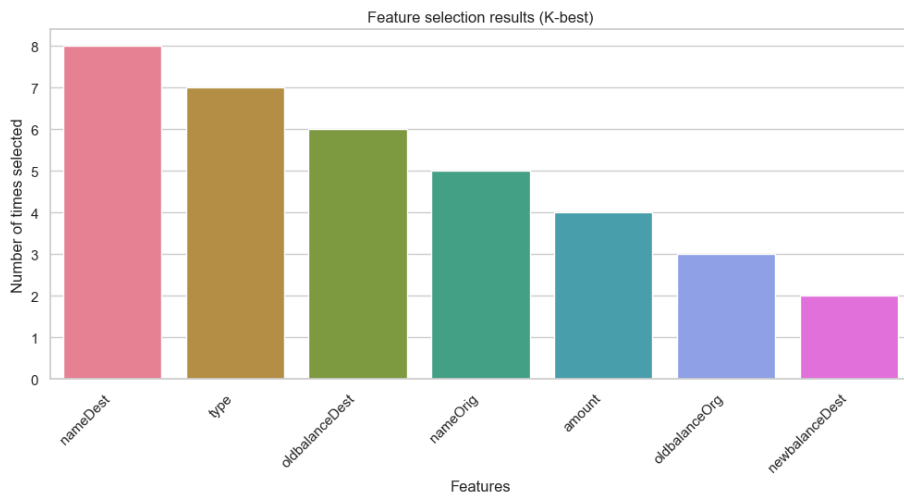

Figure 2: Figure 2: Most picked features when testing multiple K values

1.3 Metrics on Samples

For these samples, we use hyperparameters found in the 1.000.000 sample and k=8

Sample	Accuracy	Precision	Recall	F1 Score	ROC AUC
1	0.8950	0.0118	0.9767	0.0233	0.9358
2	0.9430	0.0208	0.9719	0.0407	0.9574
3	0.9373	0.0199	1.0000	0.0391	0.9686
4	0.9475	0.0219	0.8935	0.0428	0.9205
5	0.9363	0.0181	0.9000	0.0354	0.9182
6	0.9444	0.0199	0.9036	0.0389	0.9240
7	0.9636	0.0297	0.9061	0.0575	0.9349
8	0.9343	0.0177	0.9147	0.0347	0.9245
9	0.9325	0.0161	0.8594	0.0315	0.8960
10	0.9493	0.0228	0.9183	0.0445	0.9338

Table 4: Metrics of different data samples 1.000.000 transactions each

1.4 Trained on Complete Dataset

For these samples, we use hyperparameters found in the 1.000.000 sample and k=8

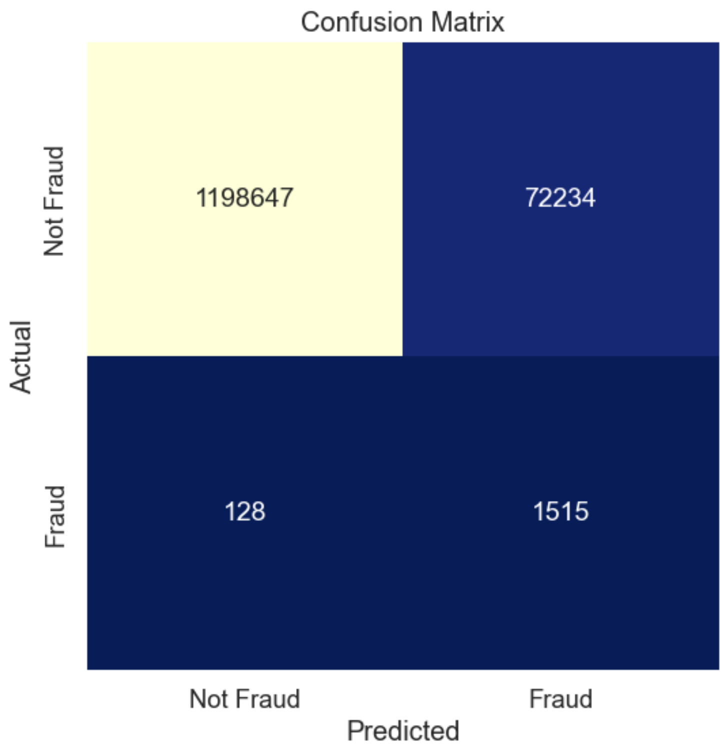

Figure 3: Figure 3: Confusion matrix of the test results on the full dataset

- Accuracy: 0.9431
- Precision: 0.0205
- Recall: 0.9221
- F1 Score: 0.0402
- ROC AUC: 0.9326

2 Logistic Regression

2.1 Hyperparameters

2.1.1 Possible Hyperparameters

Parameter	**Description**	**Used**	**Comment**
`penalty`	Specifies the norm used in the penalty term	Yes	
`solver`	Algorithm to use in the optimization problem	Yes	
`dual`	Controls the formulation of the optimization problem	No	We have n_samples > n_features
`tol`	Specifies the tolerance for stopping criterion	No	Excluded for performance reasons
`C`	Controls the penalty for misclassification	Yes	
`fit_intercept`	Controls whether to include the bias term in the model	Yes	
`intercept_scaling`	Scales the bias term	No	Features are scaled using StandardScaler
`max_iter`	Specifies the maximum number of iterations for the solver	No	Excluded for performance reasons
`class_weight`	Balances the classes in the training data	No	We handle class balancing by using undersampling
`multi_class`	Specifies how to handle multi-class classification problems.	No	Our problem is binary classification

Table 5: Usage of LogisticRegression hyperparameters in the model, their descriptions, and justification for their inclusion or exclusion.

2.1.2 Comparison of Best Hyperparameters with Different Sample Sizes

Since our dataset is so large, we tuned the hyperparameters of the LogisticRegression classifier using a random search on four different sample sizes: 10.000, 100.000, 500.000, and 1.000.000 transactions. For each sample size, we evaluated the performance of the classifier using the AUC-ROC score. From these metrics, we can decide the hyperparameters to use on the full dataset of 6.362.620 transactions.

- 10.000
- 100.000
- 500.000
- 1.000.000

Sample Size	**solver**	**penalty**	**fit_intercept**	**C**	**ROC-AUC**
10.000	newton-cg	none	true	0.013	0.775
100.000	lbfgs	none	true	1438.45	0.971
500.000	lbfgs	none	false	10000.0	0.957
1.000.000	lbfgs	none	true	0.0002	0.964

Table 6: Comparison of the best found hyperparameters and ROC-AUC score for the LogisticRegression model with different sample sizes.

2.2 K-best Feature Selection

Using the hyperparameters found before, we can use K-best feature selection on the entire dataset.

K-best	Accuracy	Precision	Recall	F1 Score	ROC AUC	Selected Features
3	0.5596	0.0029	1.0000	0.0058	0.7795	nameDest
4	0.7221	0.0041	0.8795	0.0081	0.8007	type, nameDest
5	0.6863	0.0037	0.8947	0.0073	0.7904	type, nameDest, oldbalanceDest
6	0.6914	0.0037	0.8941	0.0074	0.7926	type, nameOrig, nameDest, oldbalanceDest
7	0.9282	0.0158	0.8904	0.0310	0.9093	type, amount, nameOrig, nameDest, oldbalanceDest
8	0.9289	0.0160	0.8911	0.0313	0.9100	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest
9	0.9289	0.0160	0.8911	0.0313	0.9100	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest, newbalanceDest
10	0.9289	0.0160	0.8911	0.0313	0.9100	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest, newbalanceDest

Table 7: Comparison of the different metrics on an increasing number of K in K-best feature selection.

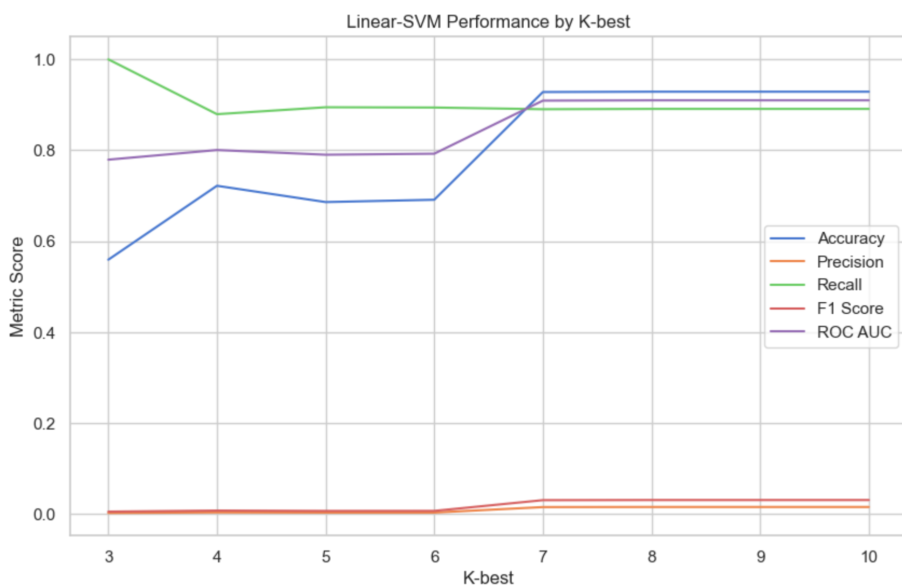

Figure 4: Figure 1: Metrics scores compared to K values

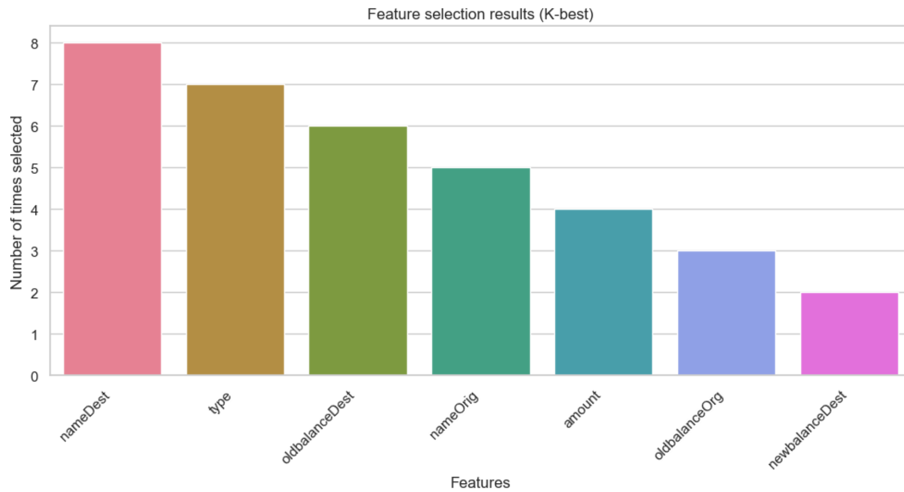

Figure 5: Figure 2: Most picked features when testing multiple K values

2.3 Metrics on Samples

For these samples, we use hyperparameters found in the 1.000.000 sample and k=8

Sample	Accuracy	Precision	Recall	F1 Score	ROC AUC
1	0.8761	0.0096	0.9339	0.0190	0.9050
2	0.9105	0.0121	0.8795	0.0239	0.8951
3	0.9135	0.0125	0.8588	0.0247	0.8862
4	0.9126	0.0129	0.8669	0.0254	0.8898
5	0.8797	0.0095	0.8885	0.0188	0.8841
6	0.8975	0.0110	0.9116	0.0217	0.9046
7	0.9157	0.0126	0.8776	0.0249	0.8966
8	0.8944	0.0107	0.8876	0.0212	0.8910
9	0.8984	0.0105	0.8398	0.0207	0.8691
10	0.9028	0.0119	0.9066	0.0234	0.9047

Table 8: Metrics of different data samples 1.000.000 transactions each

2.4 Trained on Complete Dataset

For these samples, we use hyperparameters found in the 1.000.000 sample and k=8

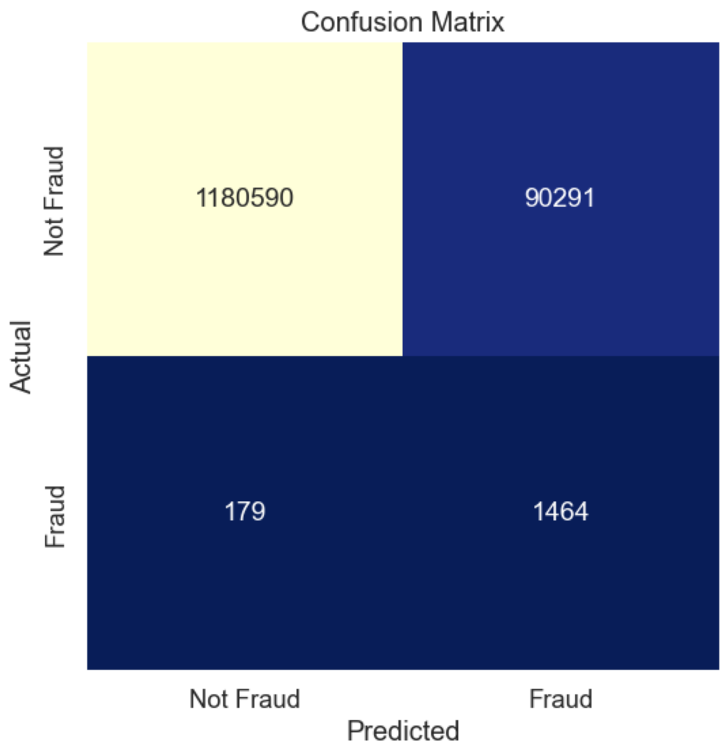

Figure 6: Figure 3: Confusion matrix of the test results on the full dataset

- Accuracy: 0.90276
- Precision: 0.011853283817469603
- Recall: 0.9066147859922179
- F1 Score: 0.023400622677513305
- ROC AUC: 0.9046849131094545

3 Random Forest Classifier

3.1 Hyperparameters

3.1.1 Possible Hyperparameters

Parameter	Description	Used	Comment
`n_estimators`	Number of decision trees in the random forest term	Yes	
`max_depth`	Maximum depth of each decision tree	Yes	
`min_samples_split`	Minimum numbers of samples required to split an internal node in each tree	Yes	
`min_samples_leaf`	Minimum number of samples required to be a leaf node in each tree	Yes	
`max_features`	Number of features to consider when looking for the best split at each node in each tree	Yes	
`criterion`	Function used to measure the quality of each split	No	Excluded for performance reasons
`bootstrap`	Whether to sample the data with replacement	No	Excluded for performance reasons
`class_weight`	Balances the classes in the training data	No	We handle class balancing by using undersampling
`min_weight_fraction_leaf`	Minimum weighted fraction of the input samples required to be a leaf node in each tree	No	Excluded for performance reasons

Table 9: Usage of RandomForestClassifier hyperparameters in the model, their descriptions, and justification for their inclusion or exclusion.

3.1.2 Comparison of Best Hyperparameters with Different Sample Sizes

Since our dataset is so large, we tuned the hyperparameters of the RandomForestClassifier classifier using a random search on four different sample sizes: 10.000, 100.000, 500.000, and 1.000.000 transactions. For each sample size, we evaluated the performance of the classifier using the AUC-ROC score. From these metrics, we can decide the hyperparameters to use on the full dataset of 6.362.620 transactions.

- 10.000
- 100.000
- 500.000
- 1.000.000

Sample Size	**n_estimators**	**min_samples_split**	**min_samples_leaf**	**max_features**	**max_depth**	**ROC-AUC**
10.000	300	2	1	auto	10	0.833
100.000	300	5	1	auto	20	0.975
500.000	100	10	1	auto	10	0.980
1.000.000	300	2	1	auto	10	0.986

Table 10: Comparison of the best found hyperparameters and ROC-AUC score for the RandomForestClassifier model with different sample sizes.

3.2 K-best Feature Selection

Using the hyperparameters found before, we can use K-best feature selection on the entire dataset.

K-best	Accuracy	Precision	Recall	F1 Score	ROC AUC	Selected Features
3	0.5596	0.0029	1.0000	0.0058	0.7795	nameDest
4	0.8523	0.0070	0.8065	0.0139	0.8294	type, nameDest
5	0.8494	0.0069	0.8131	0.0138	0.8313	type, nameDest, oldbalanceDest
6	0.8350	0.0066	0.8436	0.0130	0.8393	type, nameOrig, nameDest, oldbalanceDest
7	0.9875	0.0932	0.9976	0.1704	0.9925	type, amount, nameOrig, nameDest, oldbalanceDest
8	0.9824	0.0682	0.9988	0.1276	0.9906	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest
9	0.9836	0.0728	0.9988	0.1358	0.9912	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest, newbalanceDest
10	0.9826	0.0689	0.9988	0.1290	0.9907	type, amount, nameOrig, oldbalanceOrg, nameDest, oldbalanceDest, newbalanceDest

Table 11: Comparison of the different metrics on an increasing number of K in K-best feature selection.

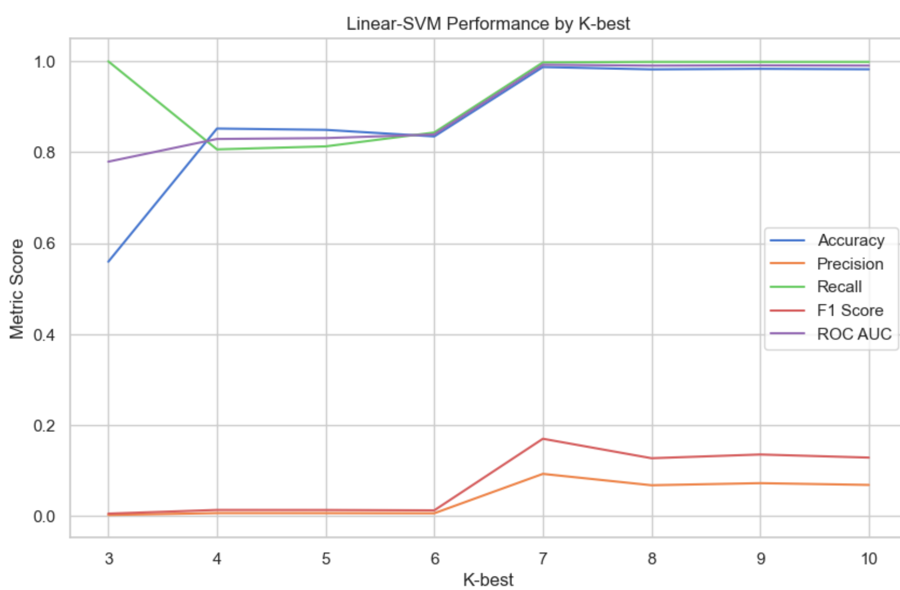

Figure 7: Figure 1: Metrics scores compared to K values

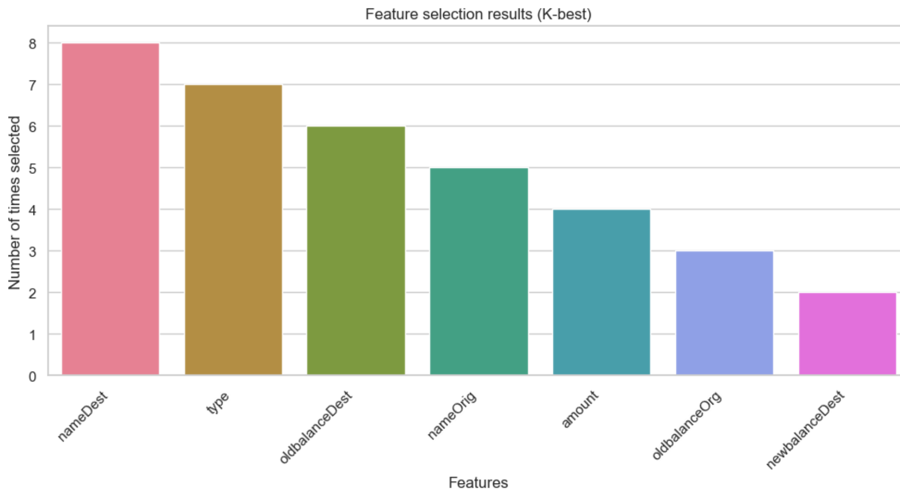

Figure 8: Figure 2: Most picked features when testing multiple K values

3.3 Metrics on Samples

For these samples, we use hyperparameters found in the 1.000.000 sample and k=7

Sample	Accuracy	Precision	Recall	F1 Score	ROC AUC
1	0.8862	0.0096	0.8599	0.0190	0.8731
2	0.9792	0.0558	0.9880	0.1056	0.9836
3	0.9576	0.0287	0.9843	0.0558	0.9709
4	0.9775	0.0544	0.9848	0.1031	0.9811
5	0.9771	0.0534	0.9923	0.1013	0.9847
6	0.9028	0.0111	0.8715	0.0218	0.8872
7	0.9522	0.0247	0.9878	0.0482	0.9700
8	0.9799	0.0603	1.0000	0.1137	0.9899
9	0.9791	0.0573	0.9922	0.1083	0.9856
10	0.9785	0.0564	1.0000	0.1069	0.9892

Table 12: Metrics of different data samples with 1,000,000 transactions each and hyperparameters found in the 1,000,000 sample with k=8.

3.4 Trained on Complete Dataset

For these samples, we use hyperparameters found in the 1.000.000 sample and k=7

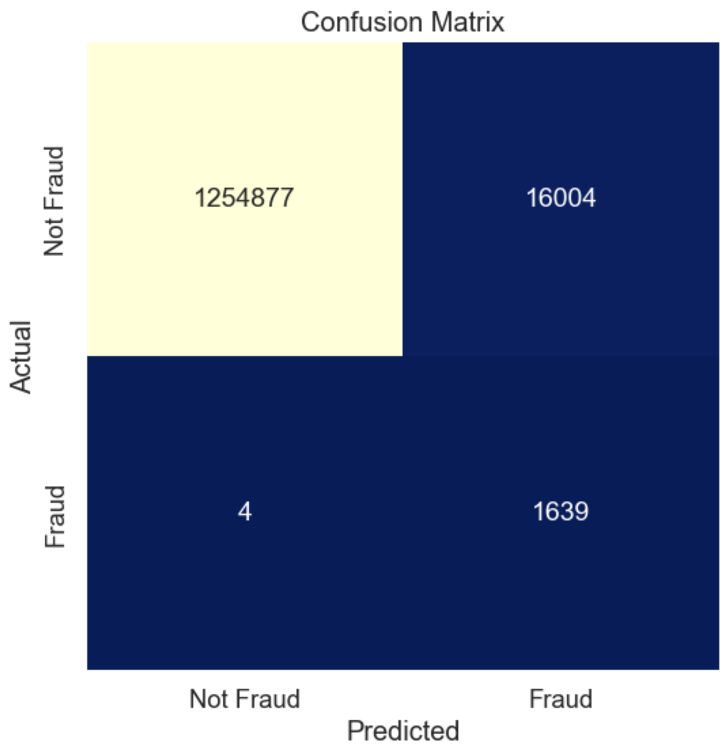

Figure 9: Figure 3: Confusion matrix of the test results on the full dataset

- Accuracy: 0.9874202765527409
- Precision: 0.09289803321430595
- Recall: 0.9975654290931223
- F1 Score: 0.16996785232811362
- ROC AUC: 0.9924862949762001