Model Interpretability using LIME

We have 2 datasets (“train” and “test” sets) each containing data on transaction fraud along with demographic details. The goal is to use a classifier ML model and check the trained model on the test dataset to capture whether a transaction is fraudulent or not. The objective here is also to try and interpret the results in a more transparent way rather than just taking the results of the “black box” ML model at face value. In order to achieve that, I trained a Random Forest classifier (black box) and compared the results against a Decision Tree classifier (more transparent).

This analysis demonstrates an effective fraud detection system using machine learning with strong interpretability features. The Random Forest model achieved 97.30% test accuracy while maintaining excellent interpretability through global surrogates and local explanations. The baseline decision tree model achieved 99.7% accuracy, which could have been due to overfitting. The main features explaining whether a transaction is fraud or not was related to the transaction amount while certain other factors played a minor role.

Data

The dataset consists of approximately 1.3 million transactions in the training set and a little above 0.55 million transactions in the test dataset, which comes out to be around 70/30 split for training and test data. There are no missing data in any of the columns -

```
In [9]: print(missing_values)
trans_date_trans_time    0
cc_num                   0
merchant                 0
category                 0
amt                      0
first                    0
last                     0
gender                   0
street                   0
city                     0
state                    0
city_pop                 0
job                      0
dob                      0
is_fraud                 0
age                      0
```

From transactions fraud perspective, I considered only certain features that seemed relevant instead of all the features present in the dataset. For example, a city/state combination is

enough to infer whether there could be any regional importance to imply whether some transactions could turn out to be fraud, instead of considering granular details such as latitude/longitude and street level details. There is also a column showing the date of birth of the person and this can be used to imply the age of the person, which could potentially be relevant as well to this exercise.

So after specifying the relevant features as inputs to training the models, we use the categorical features and LabelEncoder for encoding them.

```
relevant_features = ['category', 'amt', 'gender', 'city', 'state', 'job', 'age']


categorical_cols = ['category', 'gender', 'city', 'state', 'job']
```

Using a max_depth of 5, I first fit the training dataset to a DecisionTreeClassifier model. And using the predictions of this model on the test dataset, I get the training and testing accuracy. The decision tree shown below has the very 1st split on an amount <= 695.445, implying transaction amount is the strongest predictor of fraud. The gini statistics, number of transactions in the nodes ('samples'), count of fraud or not ('value') and majority prediction at that node are shown in the tree chart. On the right hand side of the chart, we can see large transaction amounts lead to higher fraud probability while small transaction amounts on the left are almost always not fraud.

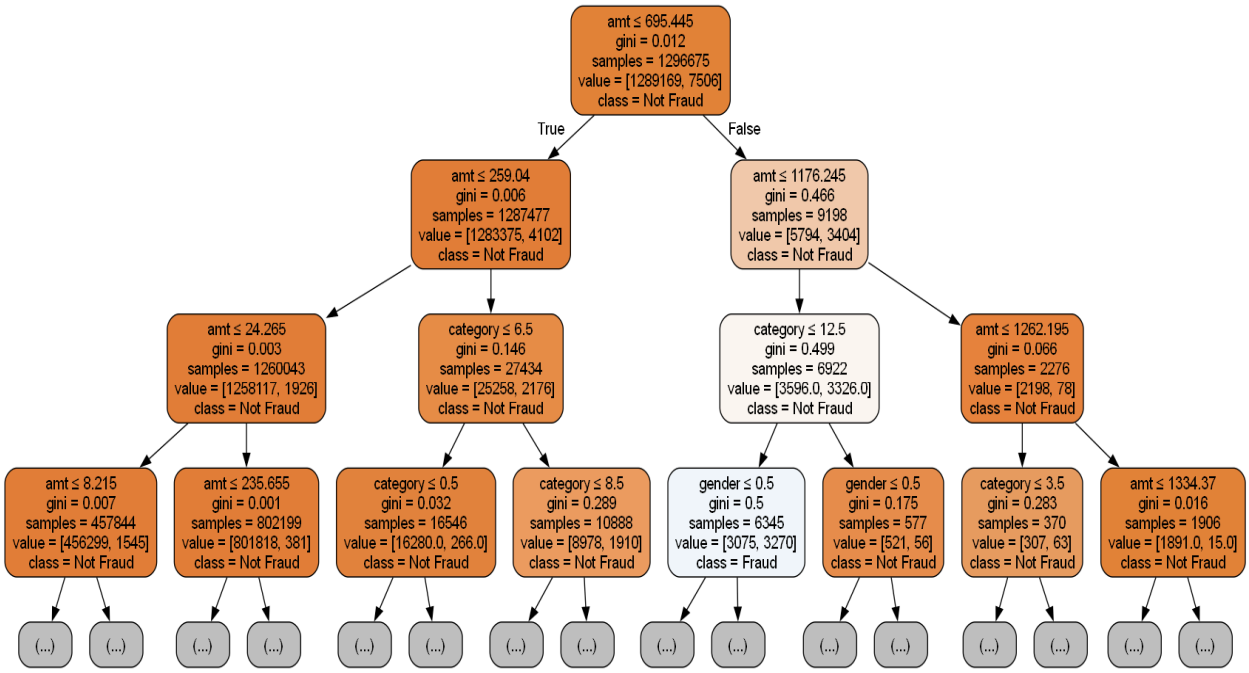

Fig1: Fraud tree from Decision tree classifier

Similarly, I fit a Random Forest classifier model as well using n_estimators =200, max_depth=5 and min_samples_leaf=100. A Random Forest model basically fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting - thus providing a better generalization compared to Decision Tree. As seen from the below accuracy metrics on the training and test dataset for each model, we can see that the Decision Tree classifier has a higher accuracy, which is likely suffering from overfitting.

Metric	Decision Tree Classifier	Random Forest Classifier
Training accuracy	99.64%	97.21%
Testing accuracy	99.72%	97.30%

For the Random Forest classifier, we retrieve the feature importance to check how much each of these variables factor into explaining the classification into fraud vs no fraud -

```
Top Features by Importance:
   variable  importance
1       amt    0.867301
0  category    0.101183
6       age    0.017675
2    gender    0.005281
5       job    0.003113
3      city    0.003108
4     state    0.002339
```

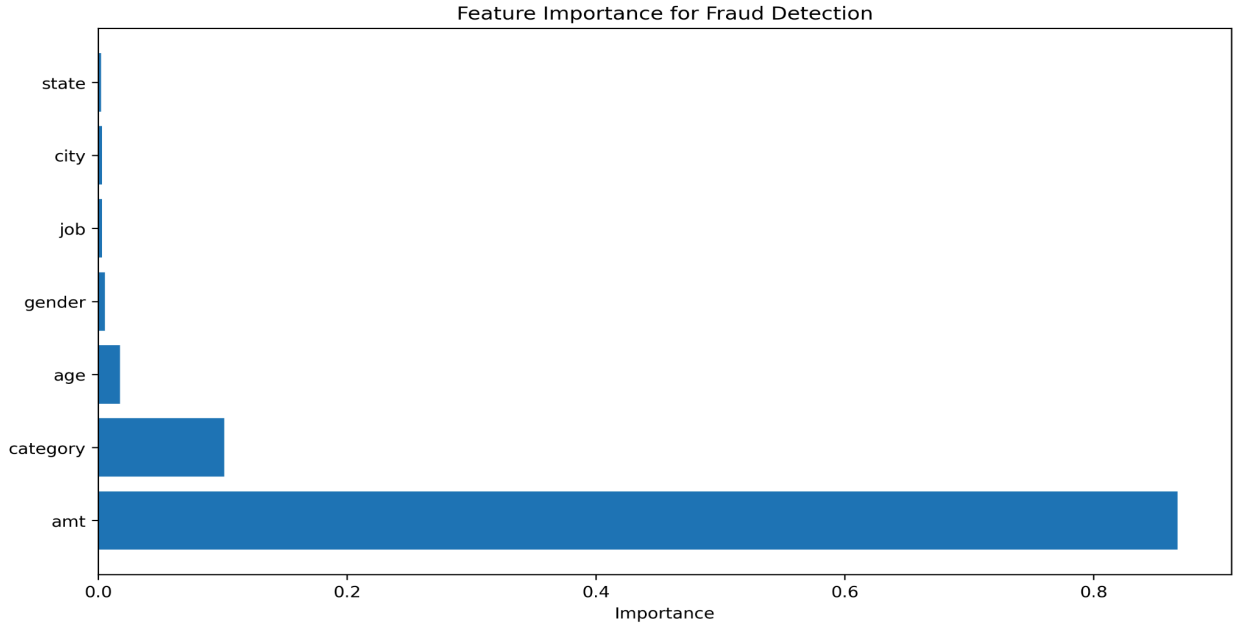

Fig2: Feature importance for fraud detection

The bar chart above also captures the above features by their importance.

So we can see that the amount is the single most important factor with ‘category’ being in 2nd place by a distance while ‘age’, ‘gender’, ‘job’ and ‘city/state’ playing very minor roles. Transaction amount and category are the dominant fraud indicators, accounting for 96.85% of predictive power.

Global Surrogate Model Performance

I created a surrogate interpretable approximation of the black box Random Forest model in order to mimic the RandomForest model’s performance. The surrogate Decision Tree successfully mimics the Random Forest and can explain over 90% of the black box Random Forest model’s behavior while being completely interpretable. This can be seen from the following metrics for the surrogate model -

Train MSE: 0.0014
Test MSE: 0.0014
R^2 Score (Train): 90.88%
R^2 Score (Test): 90.41%

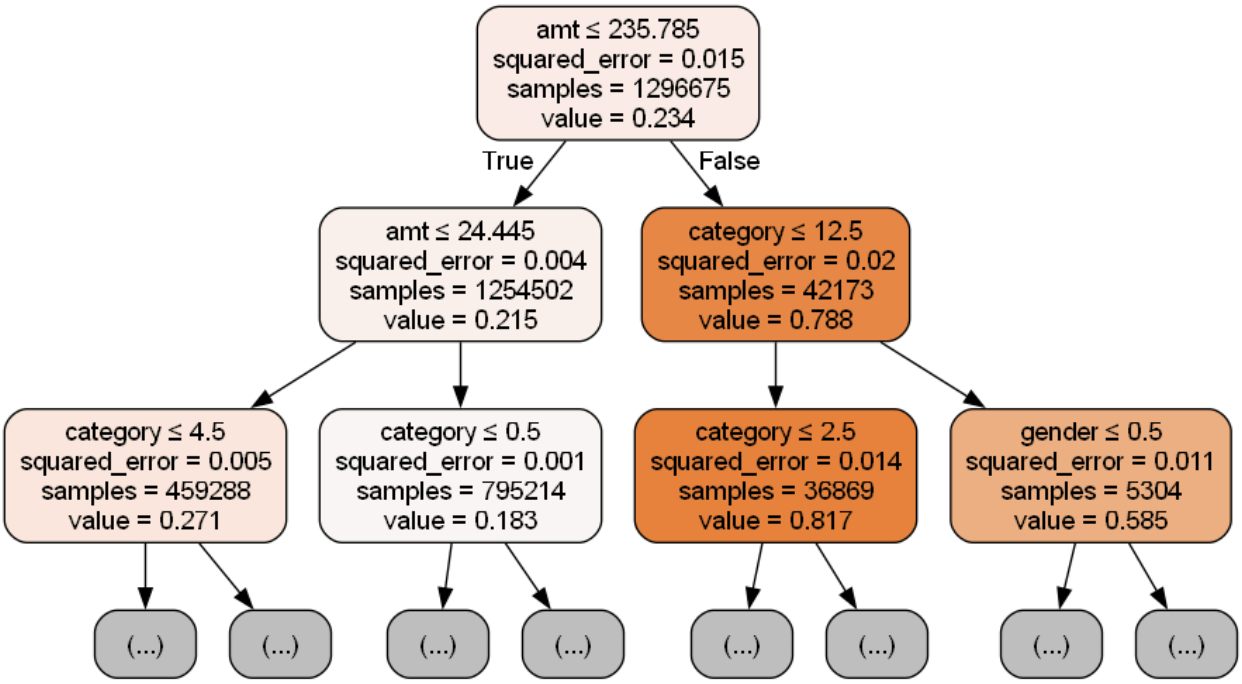

Fig3: Surrogate Tree

The surrogate tree above does not predict fraud directly from the data, but rather shows what the black box Random Forest model would predict. The value here in each box of the tree is fraud probability learned from the Random Forest model. The above tree basically shows the following interpretation:

- Root split: amt ≤ 235.785. Again, transaction amount is the most important driver in the original model too.

- If amt ≤ 24.445 and category ≤ 4.5 → predicted fraud probability ~0.27 (low).

- If amt > 235.785 and category ≤ 12.5 → fraud probability jumps to ~0.79.

- Gender also plays a role deeper in the tree, but less important compared to amount & category.

Some sample prediction probabilities from the Random Forest and the surrogate model and their differences are shown below, which also confirms that the surrogate model does well in explaining the black box Random Forest model as the difference in each of the 3 cases here is less than 5% -

```
Sample predictions comparison:
Index | Black Box Prob | Surrogate Pred | Difference
    0 | 0.1882         | 0.2170         | 0.0288
    1 | 0.2316         | 0.1857         | 0.0459
    2 | 0.1555         | 0.1857         | 0.0302
```

Local Explanations (LIME Analysis)

Use LIME explainer, I looked at two cases each of fraud and no fraud and we can look at the interpretation below:

Fraud Case Patterns:

Case 1 (Confirmed Fraud):

```
Fraud Case 0 - Top features:
  9.65 < amt <= 47.52: -0.1109
  6.00 < category <= 10.00: 0.0192
  gender <= 0.00: 0.0172
```

As seen from the 1st result, the primary driver in this case is a medium transaction amount (i.e. between $9.65-$47.52) actually decreased the fraud probability, while specific categories and gender factor increased the fraud risk slightly.

Case 2 (Confirmed Fraud):

```
Fraud Case 1 - Top features:
  amt > 83.14: 0.3899
  age > 63.00: 0.0126
  category > 10.00: 0.0098
```

In the 2nd case, we see the primary driver is a high transaction amount (>$83.14) which significantly increased risk of fraud (+0.39) while other factors such as older age (>63) and specific category increased risk by a minor amount as well.

Non-Fraud Case Patterns:

Case 1 (Non-Fraud):

```
Non-Fraud Case 0 - Top features:
  amt <= 9.65: -0.0739
  category <= 3.00: -0.0666
  49.00 < age <= 63.00: 0.0098
```

For the specific non-fraud examples, we see in the 1st case that the primary driver is a low transaction amount (<$9.65) and specific categories which decreased the risk, while age group in a certain range (49-63) increased the risk by a small amount.

Case 2 (Non-Fraud):

```
Non-Fraud Case 1 - Top features:
  9.65 < amt <= 47.52: -0.1225
  category <= 3.00: -0.0732
  state > 37.00: -0.0099
```

Similarly, in the 2nd case, the primary factor is a medium amount ($9.65-$47.52) which decreased the risk, while secondary factors such as specific categories and states provided some additional protection against fraud.

Conclusion

Overall, a high detection accuracy rate and interpretability from the surrogate model shows that the model provides excellent fraud detection capability with strong interpretability. The clear focus on transaction amount and category provides a framework for straightforward rule-based implementation.