Machine Learning
- Day 3 –

CC Fraud Detection

Georges Sakr - ESIB

Day 3: Credit Cards Fraud Detection

Georges Sakr

ESIB

Outline

- Deep convolutional neural networks
- Image classification
- CIFAR Dataset

CC Fraud

- Credit card fraud can be defined as “Unauthorized account activity by a person for which the account was not intended. Operationally, this is an event for which action can be taken to stop the abuse in progress and incorporate risk management practices to protect against similar actions in the future”.

2 types

- ***Offline fraud*** is committed by using a stolen physical card at call center or any other place .

- ***On-line fraud*** is committed via internet, phone, shopping, web, or in absence of card holder.

Brief od the dataset

- Content The datasets contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

Data description

- It contains only numerical input variables which are the result of a PCA transformation.
- Unfortunately, due to confidentiality issues the data does not identify features V1, V2, … V28 only ‘Time’ and ‘Amount’.
- Feature ‘Time’ contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature ‘Amount’ is the transaction Amount.
- Feature ‘Class’ is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Class imbalance

- Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification.

Code read the dataset

-
  ```
  pip install pandas
  ```

-
  ```
  data = pd.read_csv('creditcard.csv')
  print(data.head())
  ```

-
  ```
  data.isna().any() # find any missing data
  ```

Feature Engineering

Correlation : In statistics, dependence or association is any statistical relationship, whether causal or not, between two random variables or bi variate data.

In the broadest sense **correlation** is any statistical association, though in common usage it most often refers to how close two variables are to having a **linear relationship** with each other.

Feature Map

- Let's define the feature map from one layer to another layer. Of course, we can have multiple feature maps that learn independently from each hidden layer.

- For instance, we can start with 28 x 28 input neurons for processing MINST images and then recall *k* feature maps of size 23 x 23 neurons each (again with a stride of 5 x 5) in the next hidden layer.

Code

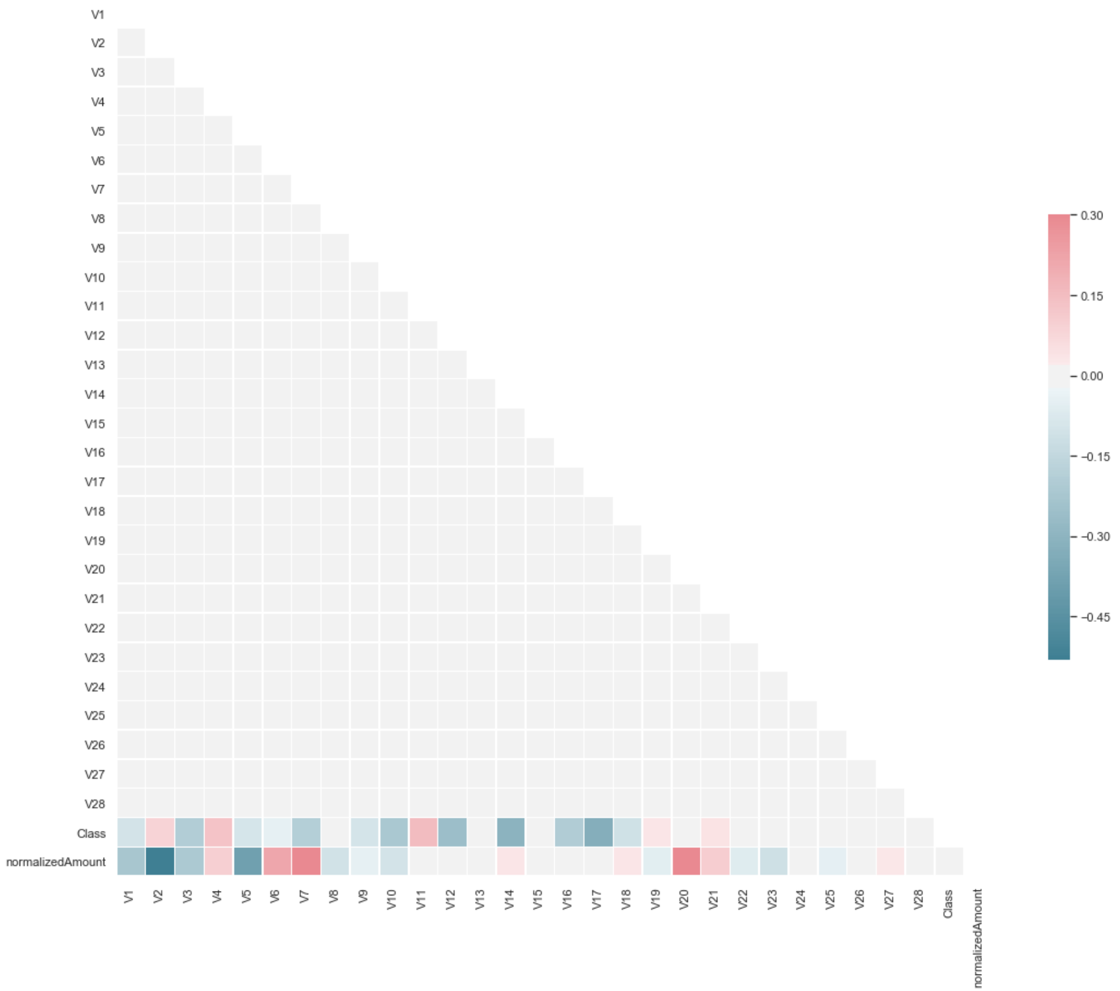
-
  ```
  data.corrwith(data.Class).plot.bar(
  figsize = (20, 10), title = "Correlation with class",
  fontsize = 15,
  rot = 45, grid = True)
  ```

-
  ```
  plt.imshow()
  ```

Correlation between variables

- # Generate a mask for the upper triangle
- mask = np.zeros_like(corr, dtype=np.bool)
- mask[np.triu_indices_from(mask)] = True
- # Set up the matplotlib figure
- f, ax = plt.subplots(figsize=(18, 15))
- # Generate a custom diverging colormap
- cmap = sn.diverging_palette(220, 10, as_cmap=True)
- # Draw the heatmap with the mask and correct aspect ratio
- sn.heatmap(corr, mask=mask, cmap=cmap, vmax=.3, center=0, square=True, linewidths=.5, cbar_kws={"shrink": .5})

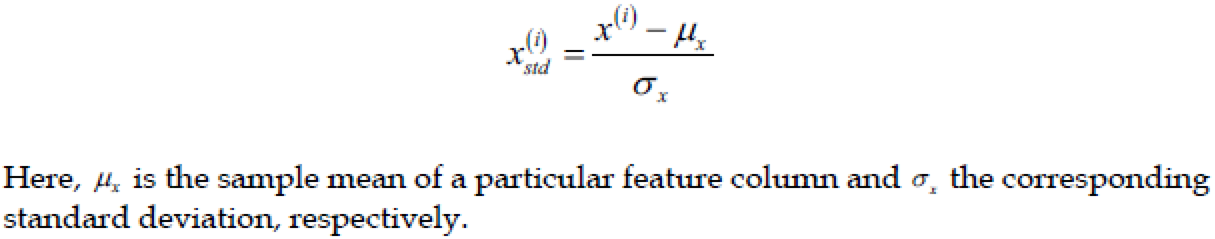
Correlation between Variable



Feature Scaling

- Bringing features onto the same scale
- Feature scaling is a crucial step in our preprocessing pipeline that can easily be forgotten.
- Decision trees and random forests are one of the very few machine learning algorithms where we don’t need to worry about feature scaling.
- However, the majority of machine learning and optimization algorithms behave much better Using standardization, we center the feature columns at mean 0 with standard deviation 1 so that the feature columns take the form of a normal distribution, which makes it easier to learn the weights.

Feature Scaling



$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

Here, μ_x is the sample mean of a particular feature column and σ_x the corresponding standard deviation, respectively.

Code

```
from sklearn.preprocessing import StandardScaler
data['normalizedAmount'] = StandardScaler().fit_transform(data['Amount'].values.reshape(-1,1))
data = data.drop(['Amount'],axis=1)
data = data.drop(['Time'],axis=1)
data.head()
```

Model Training

```
X = data.iloc[:, data.columns != 'Class']
y = data.iloc[:, data.columns == 'Class']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_state=0)
```

Decision Trees

- Decision trees are statistical data mining technique that express independent attributes and a dependent attributes logically AND in a tree shaped structure.
- Classification rules, extracted from decision trees, are IF-THEN expressions and all the tests have to succeed if each rule is to be generated.
- Decision tree usually separates the complex problem into many simple ones and resolves the sub problems
- Decision trees are predictive decision support tools that create mapping from observations to possible consequences.

Random Forest

- Random forest model is an ensemble of decision trees.
- Ensemble methods aggregates their predictions in determining the class label for a data point.
- Ensembles perform well when individual members are dissimilar, and random forests obtain variation among individual trees using two sources for randomness as follows:
  - Obtain a bootstrap sample of N cases.
  - At each node, randomly select a subset of attributes. Determine the best split at the node from this reduced set of b attributes
  - Grow the full tree without pruning
  - Random forests are computationally efficient since each tree is built independently of the others. With large number of trees in the ensemble, they are also noted to be robust to over fitting and noise in the data.

Artificial Neural Network Model

- **Fraud detection methods** based on neural network are the most popular ones.
- The advantages of neural networks over other techniques are that these models are able to learn from the past and thus, improve results as time passes.
- They can also extract rules and predict future activity based on the current situation.
- By employing neural networks, effectively, banks can detect fraudulent use of a card, faster and more efficiently.
- Among the reported credit card fraud studies most have focused on using neural networks.

Code