Report

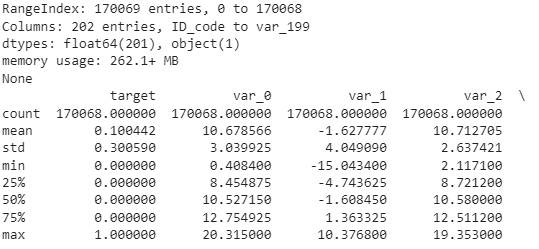
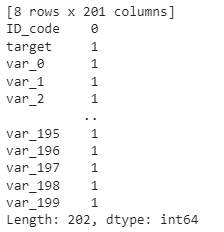
# 1. Dataset

Santander Customer Transaction Dataset:

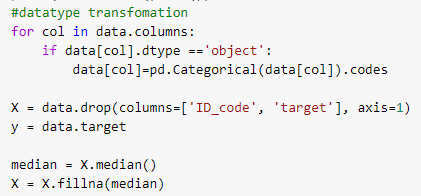
This dataset contains 200,000 bank transaction records with 200 features each, including IDs, timestamps and anonymous features. The task of this dataset is to predict whether the next transaction is fraudulent or not.

Links: <https://www.kaggle.com/c/santander-customer-transaction-prediction/data>

# 2. Data cleaning and pre-processing



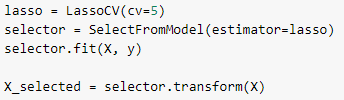
First, we loaded the dataset with the pandas module, then printed the Information, etc. of the dataset, and checked if there were any null values in the data, and printed the sum of the null values.



Next, we first convert the non-numeric features to numeric, then use the median() method to calculate the median of each feature column, and finally use the fillna() method to fill the empty values in the dataset with the median of that column.

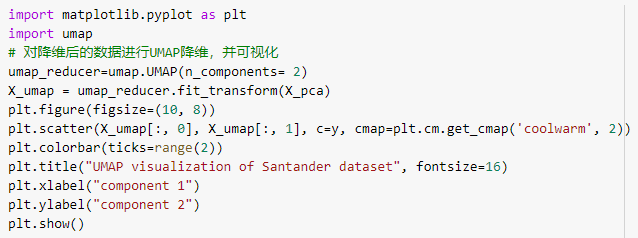
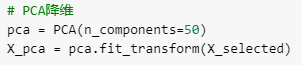
# 3. Feature Selection and Dimensionality Reduction

3.1 Feature Selection



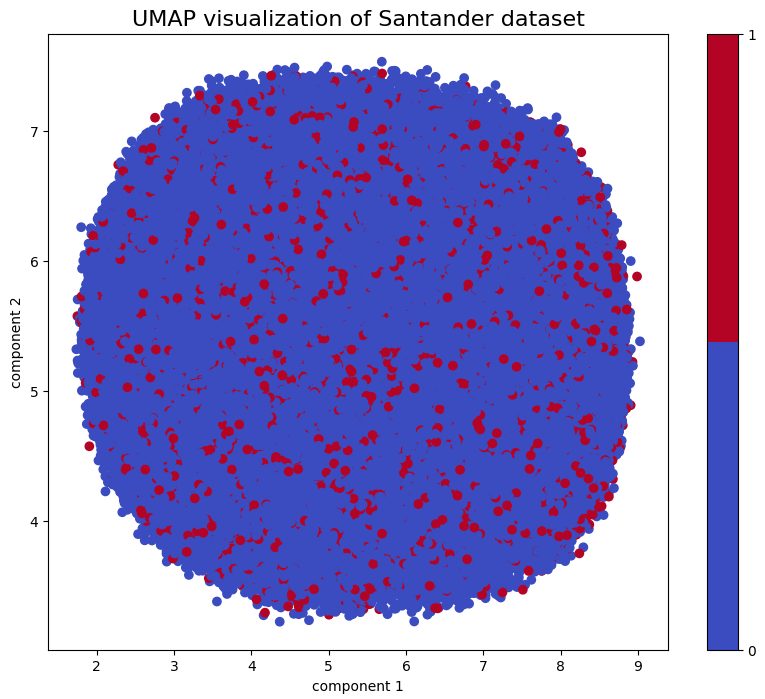
In this part, we selected the best features using the SelectFromModel class. SelectFromModel is the class used to select the important features in the model, we selected LassoCV as the model here. LassoCV is a cross-validated version of Lasso regression, which can be used to select the best regularization parameter alpha by cross-validation. By using LassoCV for feature selection, we can discard features that are not or weakly correlated with the target variables, avoiding model overfitting and improving the explanatory and generalization capabilities of the model. Also, using LassoCV, we can calculate the relative importance of each feature, which helps to understand the relationship between features in the dataset and their influence on the target variables. The feature matrix X\_selected after feature selection is used as the input for the next step of dimensionality reduction.

3.2 Dimensionality Reduction



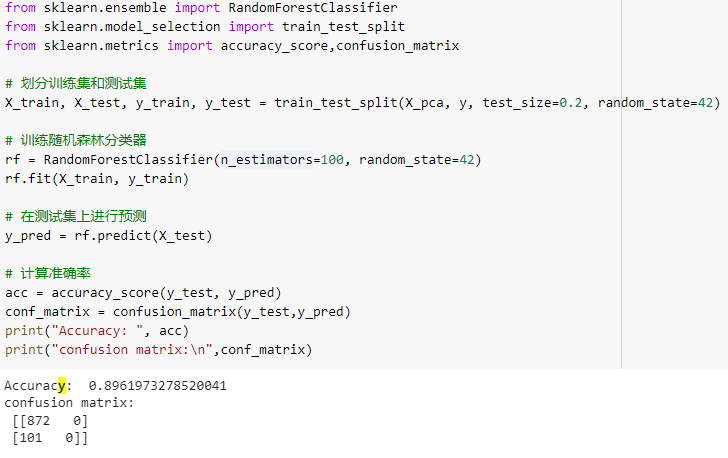
In this part, we first reduce the data to 50 features using the PCA method. Principal component analysis (PCA) is a popular technique for analyzing large datasets containing a high number of dimensions/features per observation, increasing the interpretability of data while preserving the maximum amount of information, and enabling the visualization of multidimensional data.

Then the data after PCA processing is reduced to two dimensions for visualization using the UMAP algorithm. UMAP (Uniform Manifold Approximation and Projection) is a novel manifold learning technique for dimension reduction. UMAP is constructed from a theoretical framework based in Riemannian geometry and algebraic topology. The visualization results are shown in the figure below.



With this image, we can see that the data samples are divided into two clusters, with one cluster (blue) accounting for most of the entire data set and the other cluster (red) being smaller. This result can indicate the clustering structure of the data and may also provide clues for further building classification models. It is important to note that UMAP only presents the relative position of the data in the low-dimensional space, and does not indicate its exact value. Therefore, UMAP is commonly used as a data exploration or presentation tool, rather than as a final feature extraction or classification algorithm.

# 4. Model building and prediction



After completing feature dimensionality reduction, we can use the random forest model to model and predict the data. We use the 50 features of the PCA algorithm after dimensionality reduction as input, divided the dataset into a training set and a test set, and used the RandomForestClassifier class in sklearn to train a random forest classifier. Noted that before training the model, we have performed feature selection and dimensionality reduction on the data in order to speed up the training and improve the model performance.

Then, we performed predictions on the test set and calculated the accuracy and the confusion matrix of the model. By using the reduced-dimensional features, we can significantly reduce the model training time and memory usage, and potentially improve the generalization ability and accuracy of the model.

# 5. Summary & Future Work

In this project, we selected a high-dimensional dataset "Santander Customer Transaction", we firstly cleaned and pre-processed the dataset, then reduced its dimensions and visualized it with PCA and UMAP algorithms, and finally the model was trained with random forest algorithm and achieved 89.6% accuracy on the test set.

In general, for the analysis of high-dimensional data sets, we need to perform feature selection and dimensionality reduction work, and select suitable algorithms and models for training and evaluation. In practice, we need to select and adjust algorithms and models flexibly according to specific situations to achieve better analysis results.

In the future, we can do some other improvements.

For example:

1. Feature engineering: based on feature selection, feature engineering is further considered. Methods such as constructing new features and removing redundant or useless features can be tried to better describe the data and capture the intrinsic features of the data.
2. Model selection: Other machine learning algorithms or deep learning models can be tried for classification, such as support vector machines, neural networks, etc. In addition, integrated learning techniques can be used to combine different algorithms to further improve the performance of the classifier.
3. Hyperparameter tuning: When training the model, techniques such as cross-validation can be used to tune the hyperparameters to achieve better performance, such as the n\_estimators parameter of random forest, the n\_components parameter of PCA, etc.