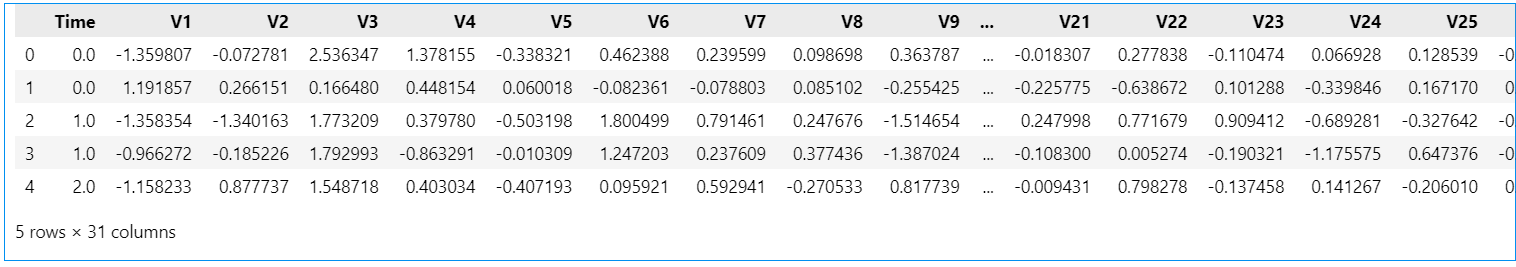
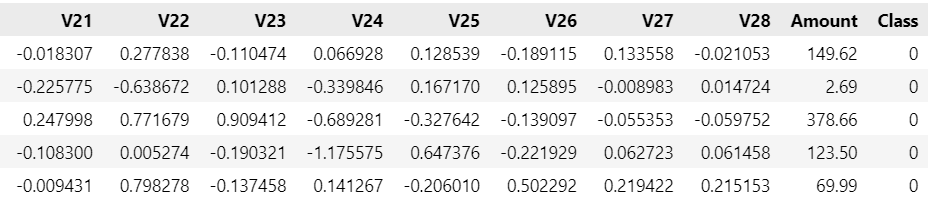
**信用卡欺诈检测实验报告**

**小组成员： 21302010011沈远哲**

**20307110256刘懿德**

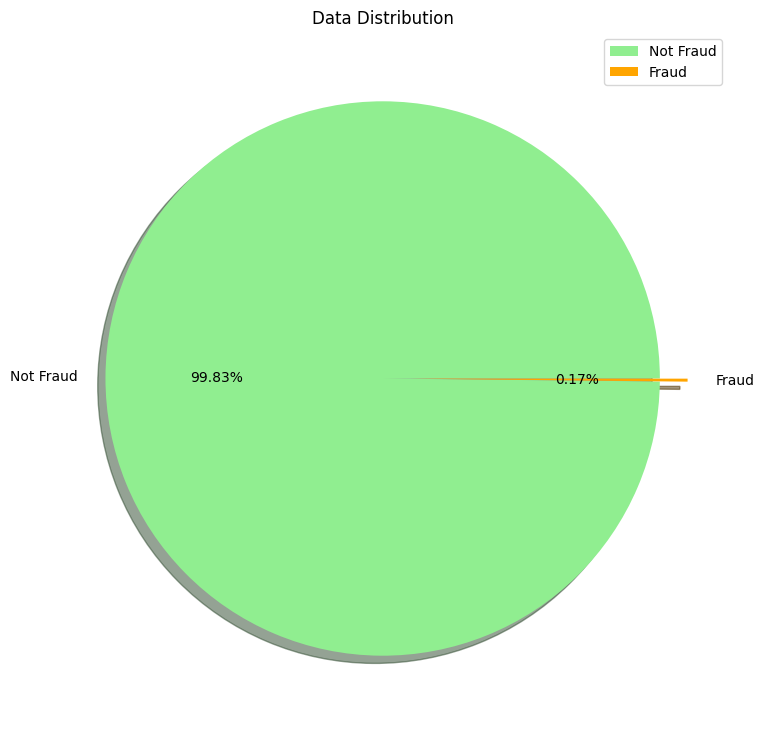
**一、数据分析**

首先查看数据内容：

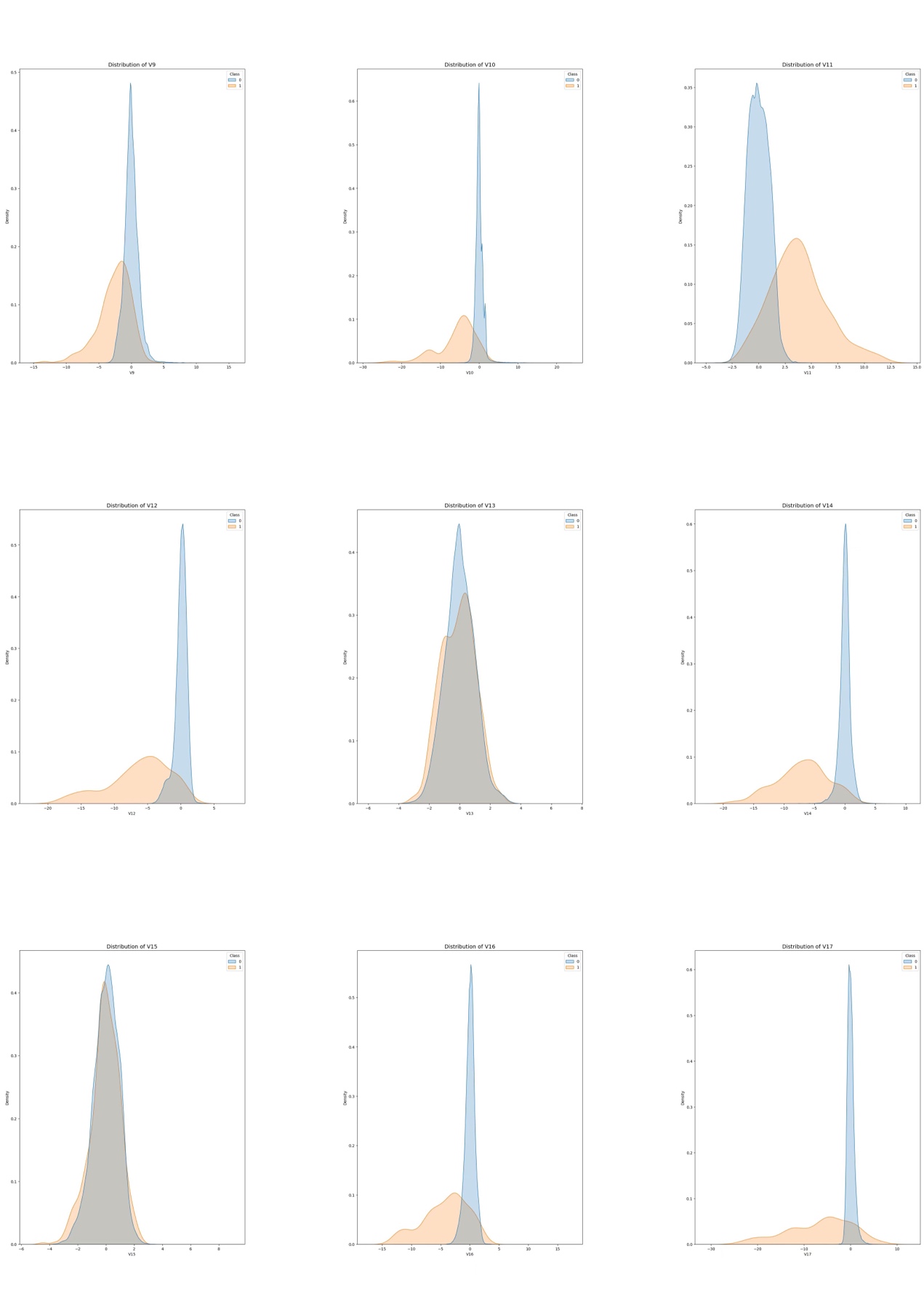
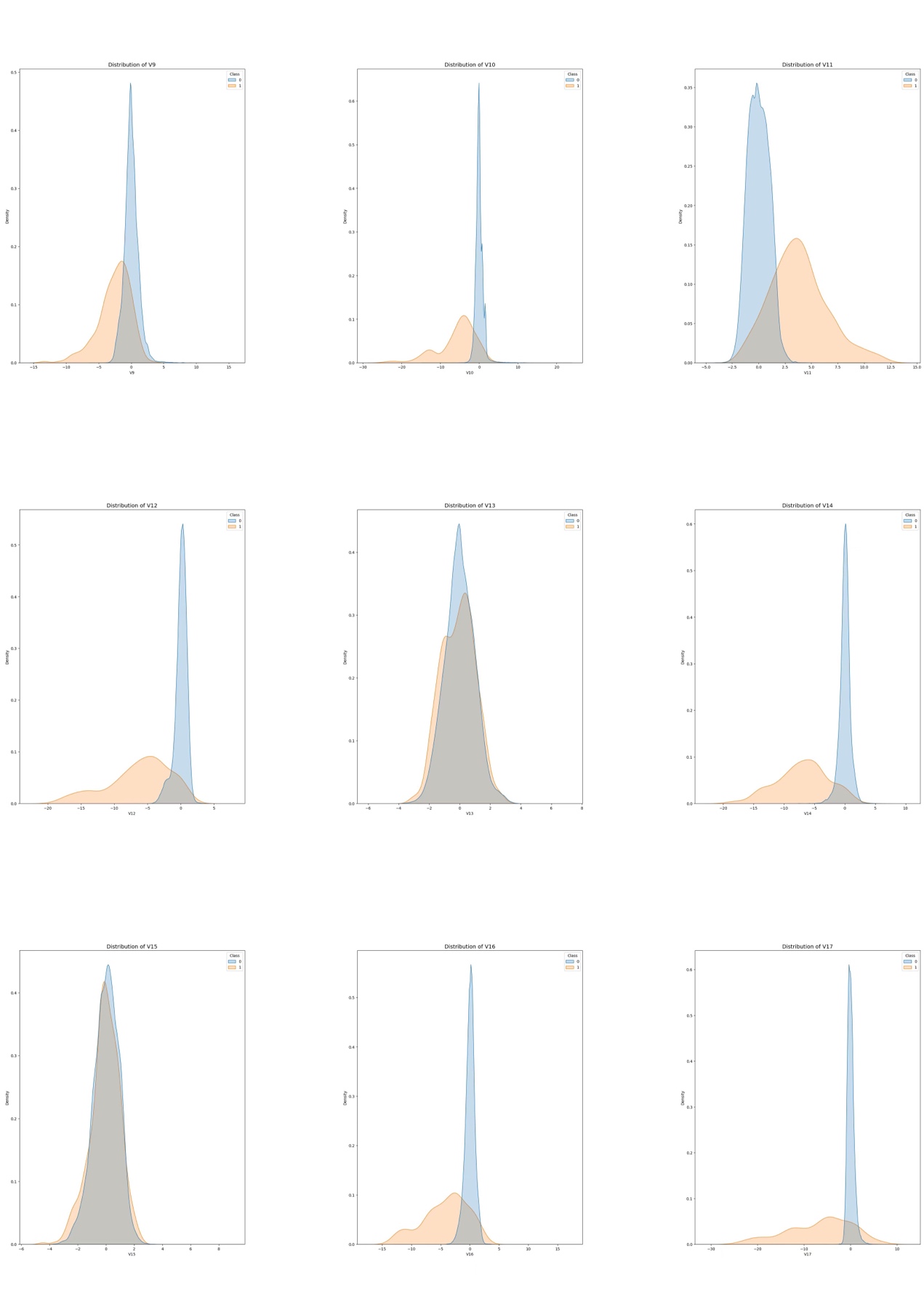
 

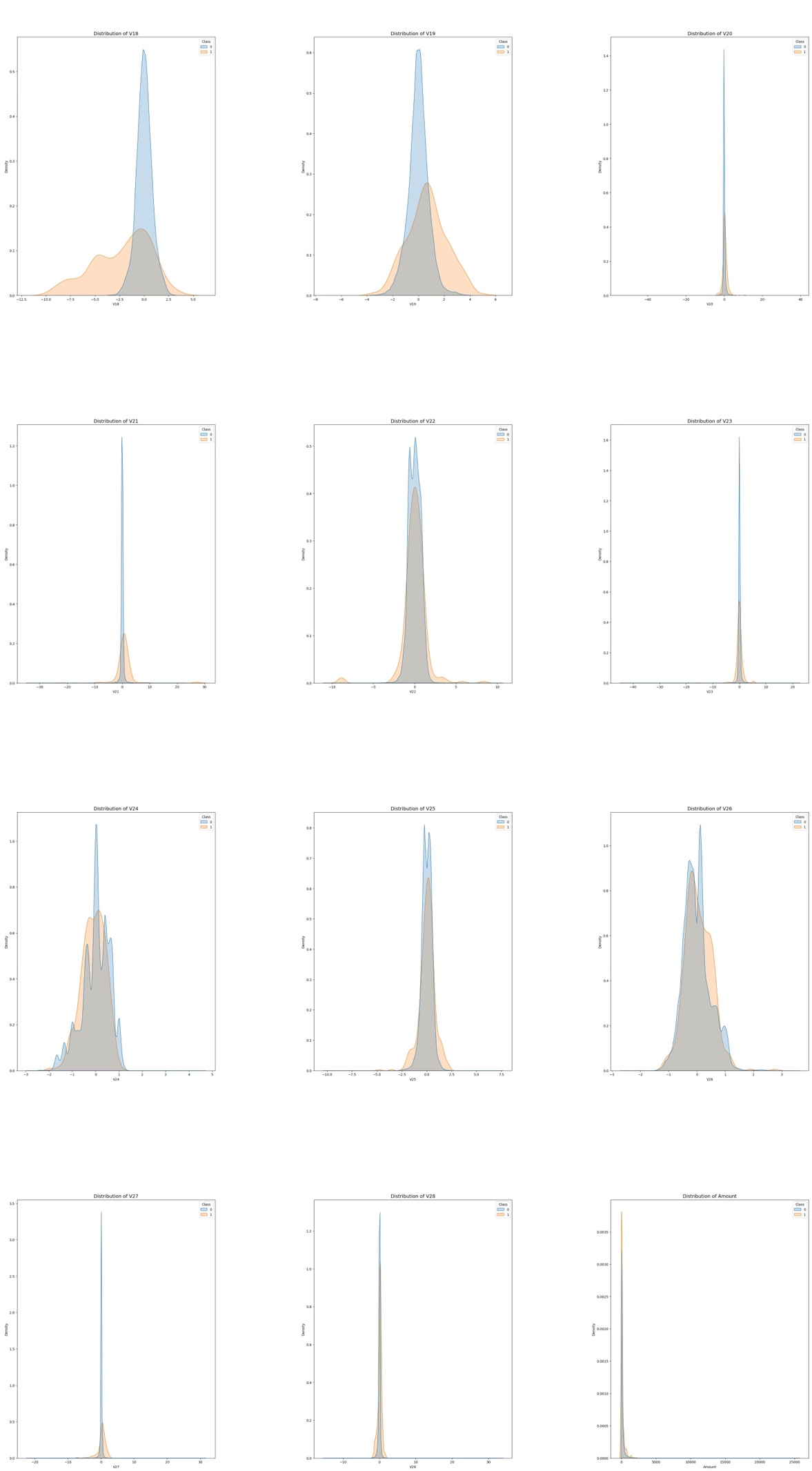
共有284807条数据，特征量除了Time和Amount外，都经过了标准化处理，且均无缺省值。

接下来查看数据分布情况：



样本数据极其不均衡，异常样本只占0.17%。



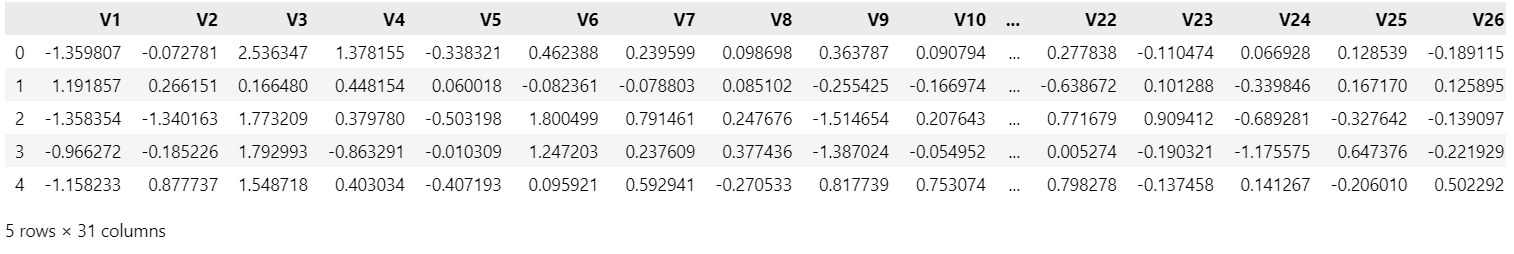


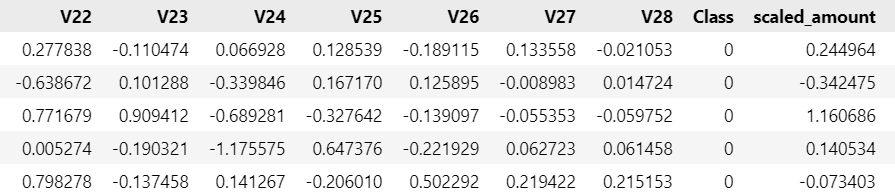
特征分布不均衡，不同特征重要性不同，存在部分特征对区分是否欺诈没有帮助。

**二、数据预处理**

首先需要剔除无关特征量，在数据集中显然Time是应该去除的特征量，然后对Amount列进行标准化，最后将所有列数据都映射到（-1，1）区间，避免量纲不一致的干扰。

最后按照80%，20%随机划分训练集和测试集，训练集中1类样本数量为396，测试集中为98，基本符合比例。





**三、各类数据处理方法和模型对比**

**一）原样本集训练测试**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC |
| XGBoost | **0.9750** | **0.7959** | **0.8764** | **0.9847** |
| LogisticRegression | 0.8500 | 0.5204 | 0.6456 | 0.9825 |
| RandomForest | 0.9740 | 0.7653 | 0.8571 | 0.9476 |
| KNeighbors | 0.9367 | 0.7551 | 0.8362 | 0.9284 |
| SVM | 0.8041 | **0.7959** | 0.8000 | 0.9523 |

可以看到XGBoost在精确度（Precision）、召回率（Recall）、F1分数（F1 Score），以及ROC曲线下面积四个指标中均表现最好。

**二）对样本集训练测试**

**1）样本分布不均衡处理**

为了解决样本分布不均衡的问题，目前主要有Random Under-Sampling，Condensed Nearest Neighbour (CNN)，Edited Nearest Neighbours (ENN)等剔除部分多数类样本的欠采样方法，SMOTE等在少数类样本近邻之间进行插值采样的过采样方法，以及综合了以上两种方法的混合采样。

**1.对训练集使用Random Under-Sampling欠采样方法后，0类样本数量降至1类样本数量，再进行训练：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC |
| XGBoost + RUS | 0.0475 | **0.9286** | 0.0904 | **0.9872** |
| LogisticRegression + RUS | **0.1673** | 0.8878 | **0.2816** | 0.9791 |
| RandomForest + RUS | 0.0562 | **0.9286** | 0.1060 | 0.9845 |
| KNeighbors + RUS | 0.0719 | 0.8776 | 0.1329 | 0.9630 |
| SVM + RUS | 0.1341 | 0.8878 | 0.2329 | 0.9832 |

可以看到在使用 RUS 欠采样方法后，对比直接使用原样本集的方法，召回率均有明显提升，ROC AUC 面积也都有所提升，但由于精确率相较不做样本处理的方法大幅下降，误杀率的大幅提高，导致 F1 分数有大幅下降。我们认为造成变化的原因主要在于训练集0类样本数量的大幅减少使模型更容易误判0为1，使精度下降，同时对1类样本的预测正确率上升，使召回率大大提高。不过由于对0类样本的选取完全随机且比例过低，信息损失较大，因此总体性能损失很大，我们认为 RUS 预处理方法效果较差，不如直接训练。

**2.对训练集使用****CNN欠采样方法后，****0类样本数量为1061，1类样本数量为396，再进行训练：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC |
| XGBoost + CNN | 0.8283 | **0.8367** | 0.8325 | 0.9791 |
| LogisticRegression + CNN | 0.8152 | 0.7653 | 0.7895 | **0.9823** |
| RandomForest + CNN | **0.8454** | **0.8367** | **0.8410** | 0.9696 |
| KNeighbors + CNN | 0.7570 | 0.8265 | 0.7902 | 0.9476 |
| SVM + CNN | 0.8041 | 0.7959 | 0.8000 | 0.9523 |

可以看到在使用CNN欠采样方法后，对比直接使用原样本集的方法，召回率均有所上升，但由于精确率的下降，F1分数有所下降。我们认为造成变化的原因在于训练集的0类样本数量大幅度减少使模型更容易误判0为1，使精确率下降，同样的，预测更多的1类会使召回率上升。但是由于CNN方法保留的样本具有代表性，信息损失较少，因此性能损失也不大，考虑到ROC AUC面积持平，以及此处检测出信用卡欺诈相对而言更重要，我们认为使用CNN方法与直接训练的效果相当（虽然CNN预处理数据比较费时间）。

**3.对训练集使用SMOTE过采样方法后，1类样本数量升至0类样本数量，再进行训练：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC |
| XGBoost + SMOTE | 0.7304 | 0.8571 | 0.7887 | 0.9907 |
| LogisticRegression + SMOTE | 0.0636 | **0.9184** | 0.1189 | 0.9798 |
| RandomForest + SMOTE | **0.8925** | 0.8469 | **0.8691** | **0.9954** |
| KNeighbors + SMOTE | 0.4970 | 0.8571 | 0.6292 | 0.9332 |
| SVM + SMOTE | 0.0798 | 0.9122 | 0.1619 | 0.9796 |

可以看到在使用SMOTE过采样方法后，对比RUS欠采样方法，召回率有所下降，而相较CNN欠采样方法，召回率则是有所上升。而精确率较RUS方法有所提升，较CNN方法除随机森林模型外均大幅降低。因此F1分数总体介于RUS和CNN方法之间。我们认为造成变化的原因在于SMOTE方法相较于RUS方法有更完整的0类样本信息，而相较于CNN方法，其在高纬空间上线性生成的1类样本必然存在和0类样本重复的样本空间，因此总体训练效果介于RUS和CNN方法之间。虽然随机森林模型有良好的表现，但由于样本量增多以及随机森林的长训练时间，我们认为使用RandomForest + SMOTE方法略逊色于XGboost直接训练或是CNN欠采样加随机森林的效果。

**4.对训练集使用****SMOTE + ENN混合采样方法后，0类样本数量为227053，1类样本数量为227453，再进行训练：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC |
| XGBoost + SMOTEENN | 0.7328 | 0.8673 | 0.7944 | 0.9888 |
| LogisticRegression+SMOTEENN | 0.0588 | 0.9184 | 0.1106 | 0.9810 |
| RandomForest + SMOTEENN | **0.8235** | 0.8571 | **0.8400** | **0.9910** |
| KNeighbors + SMOTEENN | 0.4599 | 0.8776 | 0.6035 | 0.9382 |
| SVM + SMOTEENN | 0.0818 | **0.9286** | 0.1503 | 0.9871 |

可以看到在使用SMOTE + ENN混合采样方法后，对比其他非过采样方法，召回率均有所上升，但精确率大幅度下降，导致F1分数大幅度下降。我们认为造成变化的原因在于训练集的1类样本数量大幅度增加使模型更容易误判0为1，使精确率下降，但同样的，预测更多的1类会使召回率上升。考虑到ROC AUC面积持平，以及此处检测出信用卡欺诈相对而言更重要，我们认为使用CNN方法与直接训练的效果相当（虽然CNN比较费时间）。

**2)** **特征分布不均衡处理**

我们我们选择了两种综合效果最好的模型——原样本集合+XGBoost和CNN欠采样样本集+RandomForest对比了三种处理特征的方法——手动选择相关特征，PCA降维，Autoencoder降维。这里手动选择特征相关图中有区分度的特征进行训练（选择V1, V2, V3, V4, V5, V6, V7, V9, V10, V11, V12, V14, V16, V17, V18, V19共16列变量），PCA和Autoencoder均降至16维以进行对比。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC |
| XGBoost | 0.9750 | 0.7959 | **0.8764** | **0.9847** |
| XGBoost + MS | **0.9870** | 0.7755 | 0.8686 | 0.9712 |
| XGBoost + PCA | 0.9500 | 0.7755 | 0.8539 | 0.9780 |
| XGBoost + AE | 0.9615 | 0.7653 | 0.8523 | 0.9730 |
| RandomForest + CNN | 0.8454 | **0.8367** | 0.8410 | 0.9696 |
| RandomForest + CNN + MS | 0.8367 | **0.8367** | 0.8367 | 0.9573 |
| RandomForest + CNN + PCA | 0.8172 | 0.7755 | 0.7958 | 0.9573 |
| RandomForest + CNN + AE | 0.8571 | 0.7959 | 0.8254 | 0.9568 |
| AE(Unsupervised Top1000) | 0.0680 | 0.6939 | 0.1248 | 0.9356 |

总体来看，PCA和AE的压缩效果基本一致，略低于手动选择特征。所以实验表明，除去除显性无关的特征外，其他特征选择都会丢失部分信息，降低性能，得不偿失。

**四、****oneAPI加速效果**

综合以上分析，我们选择了综合效果（训练时间少，模型效果好）最好的样本集和模型——RandomForest + CNN，同时对参数进行格搜索调优，并记录了是否使用oneAPI的训练和推理时间以进行对比。格搜索范围：

n\_estimators: [50, 100, 200],

max\_depth:[None,10, 20, 30],

min\_samples\_split: [2, 5, 10],

min\_samples\_leaf: [1, 2, 4],

max\_features: [sqrt,log2],

class\_weight: [None, balanced, balanced\_subsample],

random\_state: [42]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 Score | ROC AUC | Training(s) | Inference(s) |
| Local + RFCNN | 0.7961 | 0.8367 | 0.8159 | 0.9769 | 399.66 | 0.20 |
| Local + RFCNN + oA | 0.8454 | 0.8367 | 0.8410 | 0.9741 | 95.08 | 0.12 |
| Server + RFCNN | 0.7961 | 0.8367 | 0.8159 | 0.9769 | 207.65 | 0.17 |
| Server + RFCNN+ oA | 0.8367 | 0.8367 | 0.8367 | 0.9767 | 208.69 | 0.04 |

总的来看，在本地使用oneAPI modin加速效果最为理想，训练时间减少至非加速状态的1/4，推理时间减少至非加速状态的1/2，但在服务器上训练时间没有明显变化，可能因为服务器端机器性能不一致，推理时间减少至非加速状态的1/4。

**五、使用oneAPI感想**

oneAPI作为一个综合性的软件开发工具集，旨在支持跨多种硬件架构的高性能计算，对许多常用的机器学习库都提供了加速技术。在本次实验中，由于需要处理数十万条数据，且需要对多种机器学习模型和数据处理方法进行对比实验，在没有使用oneAPI的加速服务前每次训练模型都需要花费不少时间，每次微调参数都需要重新训练一次模型，需要等待一段不短的时间。而在使用oneAPI提供的加速服务后，可以看到机器学习模型的训练和推理时间均成几倍地缩减，大大提高了我们的实验效率。而且oneAPI使用也极其简单，由于其对许多常用机器学习库都做了适配，我们只需要从加速库中导入相关包并在开始处调用即可。正因为oneAPI有着广阔的适配库，极其简单的使用方法，几乎不需要对源代码进行改动，为我们实验效率的提高，实验方法的多样化提供了不小的帮助。

**六、结论**

总体来看，仅对数据集进行一些标准化处理后使用XGboost训练效果已经非常好了。因为数据已经经过PCA降维压缩的原因，各数据列与最终样本分类的结果均存在一定的相关性，选择数据特征并未起到很大作用，反而会丢失信息。而在平衡样本集方面，简单的随机选择或插值增加虽然很快的平衡了样本集，但破坏了数据的分布，造成最终预测效果非常不好。比较好的方案是在大样本集里选择一部分有代表性的、处于边界处的点，在减少样本数量的同时保证数据分布不发生变化，再使用一些ensemble的方法，例如Randomforest等来加强预测效果，可以得到比较好的训练时间和训练效果。

最后，使用OneAPI的加速库能显著减少训练时间和推理时间，这不仅加速了实验进程，也使大量的ensemble提高性能成为了可能。