CREDIT CARD FRAUD DETECTION

1.COMPREENSÃO DO ESTUDO

O objetivodo trabalho é aplicar técnicas de aprendizado supervisionado para detectar fraudes em transações financeiras com cartão de crédito. Inicialmente, são construídos e comparados diversos modelos de classificação, como Árvores de Decisão, Regressão Logística, KNN, Floresta Aleatória, SVC, SGDClassifier e Naive Bayes, utilizando métricas como recall, precisão e F1-score para selecionar os modelos mais promissores a serem estudados ao longo do trabalho. **Devido ao alto desbalanceamento dos dados, a métrica recall é priorizada para maximizar a detecção de fraudes reais**, adotando técnicas de balanceamento como undersampling e oversampling para melhorar o desempenho dos modelos.

Além da comparação inicial de modelos, **o foco principal do trabalho está na análise detalhada e no desempenho comparativo dos classificadores de Regressão Logística e Floresta Aleatória.** Esses modelos são treinados e avaliados com diferentes estratégias de balanceamento de classes, empregando métricas como acurácia, precisão, recall e F1-score para medir sua eficácia. A análise de curvas ROC e a métrica AUC também são exploradas para compreender melhor o comportamento desses classificadores.

Adicionalmente, **o trabalho realiza um processo de seleção de variáveis (feature selection)** para identificar os atributos mais relevantes na detecção de fraudes, aprimorando a eficiência e a precisão dos modelos. **Também são utilizadas validação cruzada e otimização de hiperparâmetros para ajustar os parâmetros dos classificadores**, visando maximizar a performance preditiva. Esse refinamento resulta em uma abordagem mais robusta e confiável para a detecção de fraudes, garantindo que os modelos estejam configurados para alcançar o melhor equilíbrio entre precisão e generalização.

1.1 MÉTRICAS

Para o estudo e análise comparativa dos resultados que serão adquiridos, serão consideradas as métricas recall e precisão. Durante a apresentação dos resultados, também será exibida a acurácia obtida; no entanto, essa não será a principal métrica, **visto que, conforme analisado abaixo, o conjunto de dados apresenta um alto desbalanceamento dos dados da variável alvo. Logo, a acurácia não é a melhor métrica nesses casos.** Como o estudo tem como finalidade identificar todos os casos realmente fraudulentos, a métrica de recall será a mais importante no critério de escolha, sendo sempre balanceada e analisada em conjunto com um valor coerente de precisão.

2.COMPREENSÃO DO ESTUDO

O presente conjunto de dados que será trabalhado nesse projeto, está disponível no seguinte **Credit Card Fraud Detection**

O conjunto de dados contém transações feitas por cartões de crédito em setembro de 2013 por titulares de cartão europeus. [1]

Este conjunto de dados apresenta **transações que ocorreram em dois dias, onde temos 492 fraudes de 284.807 transações. O conjunto de dados é altamente desbalanceado, a classe positiva (fraudes) é responsável por 0,172% de todas as transações.** [1]

INFORMAÇÕES DA BASE:

- 'Time'(tempo): contém os segundos decorridos entre cada transação e a primeira transação no conjunto de dados;
- 'Amount'(valor): é o valor da transação;
- 'Class' (classe): é a variável de resposta e assume valor 1 em caso de fraude e 0 caso contrário;
- **Devido a questões de confidencialidade, não podemos fornecer os recursos originais e mais informações básores sobre os dados. Características V1, V2 ... V28 são os princípais componentes obtidos com "PCA".**

Elaborado por: Yago Castro dos Reis

3.ANÁLISE INICIAL

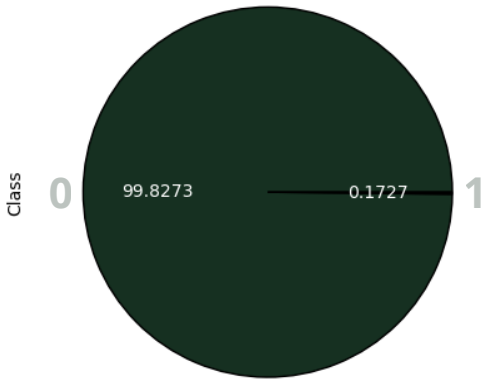
Nesta etapa, será realizada uma análise preliminar do conjunto de dados para compreender suas principais características e identificar possíveis desafios que possam influenciar o desempenho dos modelos de aprendizado supervisionado. O foco inicial é verificar a qualidade e a estrutura dos dados, avaliar a distribuição da variável alvo e definir estratégias para lidar com problemas, como o desbalanceamento de classes.

O conjunto de dados utilizado neste projeto contém **284.807 registros e 31 colunas.**

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

	Time	Amount	Class
count	284807.00	284807.00	284807.00
mean	94813.86	88.35	0.00173
std	47488.15	250.12	0.042
min	0.00	0.00	0.00
25%	54201.50	5.60	0.00
50%	84692.00	22.00	0.00
75%	139320.50	77.17	0.00
max	172792.00	25691.16	1.00

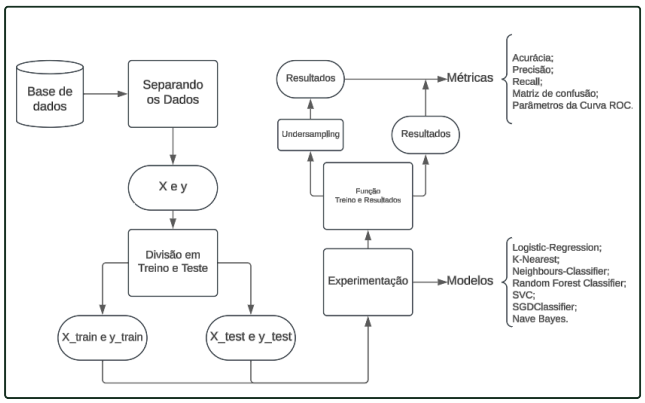
O resumo estatístico inicial sugere uma ampla variação nas variáveis, especialmente em Amount e Time, reforçando a necessidade de normalização ou padronização em etapas posteriores.



Esse desbalanceamento exige o uso de técnicas específicas, como o balanceamento das classes, para garantir a eficácia dos modelos de classificação. **Caso contrário, o modelo pode apresentar alta acurácia geral, mas com baixa eficiência na identificação de fraudes.**

4.ESCOLHA DO MODELO DE ESTUDO

Nesta seção, serão analisados diferentes algoritmos de classificação disponíveis na biblioteca scikit-learn do Python, voltados para o aprendizado supervisionado. O principal objetivo é avaliar o desempenho desses modelos quando aplicados ao conjunto de dados, com ênfase nas métricas curva ROC, recall e precisão. Com base nesses indicadores, serão selecionados os dois modelos com o melhor desempenho geral para a aplicação em questão. Além disso, ao longo das seções do trabalho, será explorado o comportamento desses modelos utilizando diferentes técnicas e abordagens, examinando como as métricas de avaliação variam conforme as estratégias adotadas.



Esta primeira tentativa é com o conjunto de dados original, porém, **vamos perceber que, por questões de tempo e pela quantidade de dados, é mais vantajoso aplicar o undersampling para reduzir a quantidade de dados. Veremos que o resultado obtido é semelhante.**

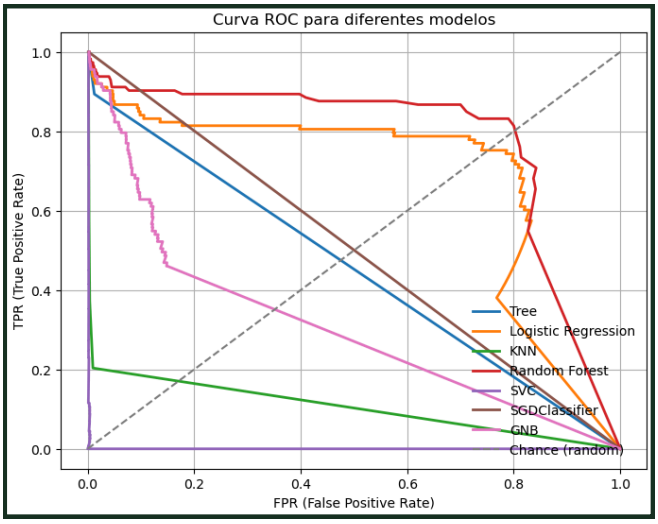
Treinamento dos Modelos

43,7 min

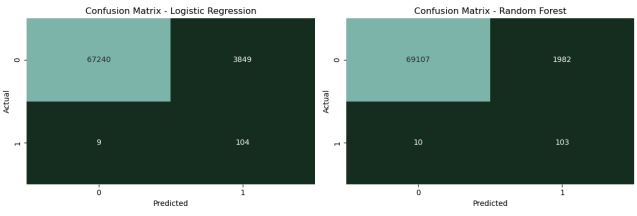
com undersampling

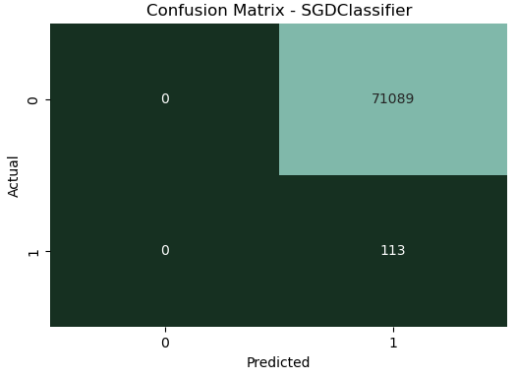
31.2 s

PRIMEIROS RESULTADOS



Modelos	Modelos	Acurácia	Precisão	Recall	F1-Score
1	SGDClassifier	0.0016	0.0016	1.0000	0.0032
2	Logistic Regression	0.9458	0.0263	0.9204	0.0511
3	Random Forest	0.9720	0.0494	0.9115	0.0937
4	Tree	0.8884	0.0125	0.8938	0.0248
5	GNB	0.9862	0.0805	0.7345	0.1452
6	KNN	0.6548	0.0026	0.5752	0.0053
7	SVC	0.4827	0.0017	0.5487	0.0034



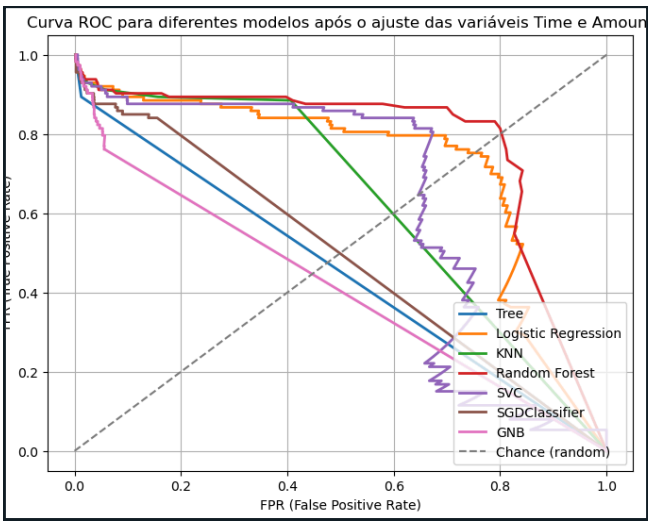


Conforme é possível verificar nos resultados apresentados anteriormente, os modelos Logistic Regression e Random Forest apresentaram um melhor desempenho. Apesar de o modelo SGDClassifier ter apresentado um Recall mais alto, ele não será selecionado, pois seu comportamento se classifica como overfitting nos dados, e possivelmente não terá o mesmo desempenho para dados futuros.

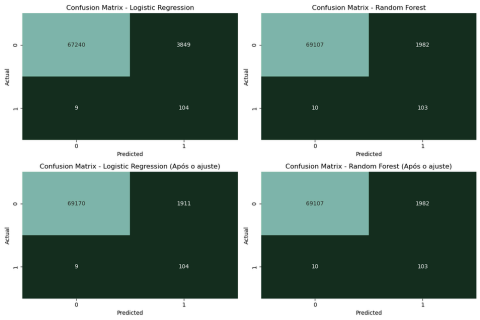
4.1.AJUSTANDO OS VALORES

Nesta etapa, serão ajustadas algumas variáveis para verificar novamente o comportamento do treinamento nos algoritmos.

RESULTADOS



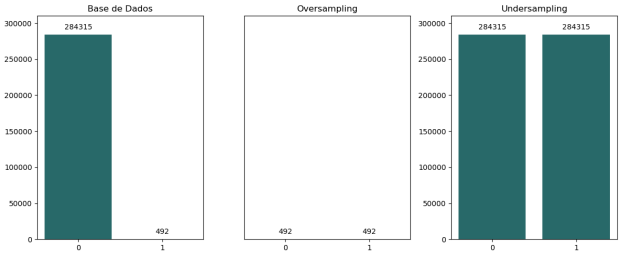
Modelos	Modelos	Acurácia	Precisão	Recall	F1-Score
1	Logistic Regression	0.9730	0.0516	0.9204	0.0977
2	Random Forest	0.9720	0.0494	0.9115	0.0937
3	KNN	0.9765	0.0583	0.9115	0.1096
4	SGDClassifier	0.9280	0.0197	0.9115	0.0386
5	SVC	0.9842	0.0834	0.8938	0.1527
6	Tree	0.8884	0.0126	0.8938	0.0248
7	GNB	0.9622	0.0357	0.8761	0.0686



Com o ajuste dos dados, foi possível identificar que **os dois modelos que apresentaram os melhores valores na métrica alvo (Recall) foram o Logistic Regression e o Random Forest,** conforme destacado na tabela anterior. Assim, esses dois modelos serão analisados ao longo do código, com o objetivo de avaliar seu comportamento utilizando diferentes técnicas e parâmetros.

5.EXPLORANDO AS DIFERENTES TÉCNICAS DE BALANCEMENTO DOS DADOS

Essa seção tem como objetivo analisar cada técnica de balanceamento de dados e selecionar aquela que, com base nas métricas de avaliação consideradas, apresentar o melhor resultado. Posteriormente, essa técnica será utilizada na próxima seção.



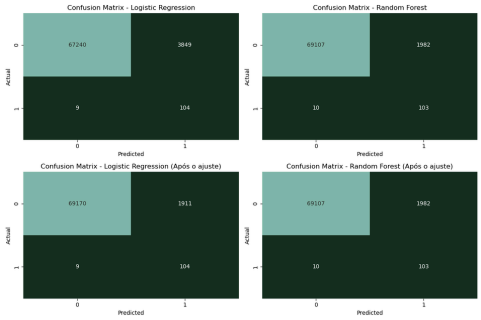
OBSERVAÇÃO: Sobre o oversampling, como a repetição dos dados pode ser um problema, é possível utilizar o parâmetro shrinkage. Quando esse parâmetro é aplicado, ele realiza uma 'suavização' nos dados.

Variável	Valor	Repetições
Time	0.3947	3518
Time	0.4936	2324
Amount	0.000039	78993
V21	27.2028	3517

Como critério de comparação ou resultado inicial, será desenvolvido um modelo sem a aplicação de nenhuma técnica de balanceamento.

RESULTADOS

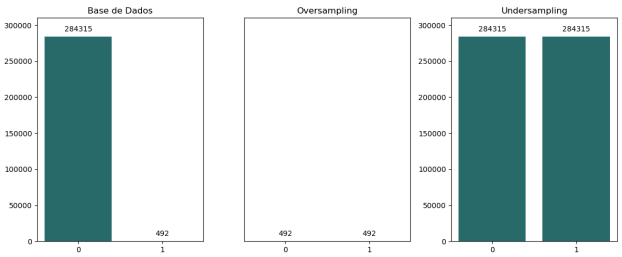
Modelos	Acurácia	Precisão	Recall	F1 score	Matriz
RF_init	99.9579	0.9429	0.8049	0.8684	[[71073, 6], [24, 99]]
RF_RU	97.6194	0.0609	0.8862	0.1140	[[69398,1681], [14, 109]]
RF_RU_CC	71.1595	0.0056	0.9431	0.0112	[[50551, 20528], [7, 116]]
RF_RU_NM	4.6642	0.0018	0.9756	0.0035	[[3201, 67878], [3, 120]]
RF_RO	99.9522	0.9588	0.7561	0.8455	[[71075, 4], [30, 93]]
RF_RO_SM	99.9424	0.8661	0.7886	0.8255	[[71064, 15], [26, 97]]
RF_RO_ADA	99.9382	0.8559	0.7724	0.8120	[[71063, 16], [28, 95]]
LR_init	99.9157	0.8462	0.6260	0.7196	[[71065, 14], [46, 77]]
LR_RU	97.0675	0.0503	0.8943	0.0953	[[69004, 2075], [13, 110]]
LR_RU_CC	95.7894	0.0343	0.8618	0.0660	[[68098, 2981], [17, 106]]
LR_RU_NM	63.4757	0.0044	0.9431	0.0088	[[45080, 25999], [7, 116]]
LR_RO	97.6784	0.0624	0.8862	0.1165	[[69440, 1639], [14, 109]]
LR_RO_SM	97.5057	0.0583	0.8862	0.1093	[[69317, 1762], [14, 109]]
LR_RO_ADA	91.7207	0.0187	0.9106	0.0366	[[65195, 5884], [11, 112]]



Com o ajuste dos dados, foi possível identificar que **os dois modelos que apresentaram os melhores valores na métrica alvo (Recall) foram o Logistic Regression e o Random Forest,** conforme destacado na tabela anterior. Assim, esses dois modelos serão analisados ao longo do código, com o objetivo de avaliar seu comportamento utilizando diferentes técnicas e parâmetros.

5.EXPLORANDO AS DIFERENTES TÉCNICAS DE BALANCEMENTO DOS DADOS

Essa seção tem como objetivo analisar cada técnica de balanceamento de dados e selecionar aquela que, com base nas métricas de avaliação consideradas, apresentar o melhor resultado. Posteriormente, essa técnica será utilizada na próxima seção.



OBSERVAÇÃO: Sobre o oversampling, como a repetição dos dados pode ser um problema, é possível utilizar o parâmetro shrinkage. Quando esse parâmetro é aplicado, ele realiza uma 'suavização' nos dados.

Variável	Valor	Repetições
Time	0.3947	3518
Time	0.4936	2324
Amount	0.000039	78993
V21	27.2028	3517

Como critério de comparação ou resultado inicial, será desenvolvido um modelo sem a aplicação de nenhuma técnica de balanceamento.