Análise de Concessão de Crédito usando Técnicas de Ciência de Dados

Curso: Análise e desenvolvimento de sistemas
Disciplina: Ciência de dados
Alunos: Davi Souza e Gustavo Souza
Ano: 2025

Objetivo do Projeto

- Identificar fatores associados à inadimplência de empréstimos utilizando as variáveis do conjunto de dados.
- Construir um modelo capaz de prever se um cliente pode se tornar inadimplente.
- Criar um pipeline seguindo o CRISP-DM que é uma metodologia estruturada e cíclica muito utilizada em Data Science

Entendimento dos Dados

- Base obtida do Kaggle (Lending Club adaptado).
- Variável alvo: loan_condition_cat (0 = bom empréstimo, 1 = inadimplente).
- Diversas variáveis socioeconômicas, financeiras e categóricas.
- O Dataset possui estrutura adequada para classificação supervisionada.

Dados do Banco

	id	year	issue_d	final_d	emp_length_int
0	63398958.0	2015.0	01/11/2015	1122015.0	8.0
1	27610673.0	2014.0	01/10/2014	1012016.0	3.0
2	49925091.0	2015.0	01/05/2015	1102015.0	10.0
3	28102260.0	2014.0	01/10/2014	1012016.0	9.0
4	57324697.0	NaN	01/08/2015	1012016.0	10.0
5	61402817.0	2015.0	01/11/2015	1012016.0	10.0
6	38700393.0	2015.0	01/01/2015	1012016.0	2.0
7	59955204.0	2015.0	01/10/2015	1012016.0	10.0
8	6156565.0	2013.0	01/07/2013	1012016.0	6.0
9	27511428.0	2014.0	01/09/2014	1122015.0	3.0

10 rows × 23 columns

Informação das Colunas

```
Data columns (total 23 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    7906 non-null   float64
 1   year                  7917 non-null   float64
 2   issue_d               7923 non-null   object
 3   final_d               7900 non-null   float64
 4   emp_length_int        7912 non-null   float64
 5   home_ownership_cat    7921 non-null   float64
 6   income_category       7929 non-null   object
 7   annual_inc            7934 non-null   float64
 8   income_cat            7923 non-null   float64
 9   loan_amount           7281 non-null   float64
 10  term_cat              7912 non-null   float64
 11  application_type_cat  7946 non-null   float64
 12  purpose_cat           7929 non-null   float64
 13  interest_payment_cat  7924 non-null   float64
 14  loan_condition_cat    7921 non-null   float64
 15  interest_rate         7285 non-null   float64
 16  grade_cat             7914 non-null   float64
 17  dti                   7921 non-null   float64
 18  total_pymnt           7923 non-null   float64
 19  total_rec_prncp       7924 non-null   float64
 20  recoveries            7942 non-null   float64
 21  installment           7919 non-null   float64
 22  region                7918 non-null   object
dtypes: float64(20), object(3)
memory usage: 1.4+ MB
```

Suposições Iniciais e Hipóteses

Exproratórias

- H1: Taxa de juros varia entre regiões.
- H2: Quanto maior a Renda maior o empréstimos.
- H3: O índice de DTI difere entre finalidades.

Explicativas

- H4: Taxas de juros explicadas pela grade.
- H5: Condição final depende de DTI, juros e renda.

Preditivas

- H6: Modelos supervisionados conseguem prever inadimplência com alta precisão.
- H7: principais variáveis: grade, interest_rate, dti.

Descrição das Variáveis Estatísticas

Conclusões observadas

- A maioria dos empréstimos ocorreu entre 2013 e 2015, caracterizando uma base relativamente recente e concentrada.
- Os empréstimos apresentam valores de pequeno a médio porte, com média em torno de R$ 13 mil.
- A maior parte dos contratos possui prazo de 36 meses, com um grupo menor de 60 meses.
- As taxas de juros situam-se em torno de 13%, refletindo níveis médios no período analisado.
- O comprometimento médio da renda dos clientes gira em torno de 18%, considerado dentro de uma faixa financeiramente aceitável.

	count	mean	std	min	25%	50%	75%	max
id	7906.0	3.261400e+07	2.301002e+07	68817.00	8966704.500	34894352.50	5.523986e+07	6.861689e+07
year	7917.0	2.014021e+03	1.259601e+00	2007.00	2013.000	2014.00	2.015000e+03	2.015000e+03
final_d	7900.0	1.046885e+06	4.541793e+04	1012009.00	1012016.000	1012016.00	1.092015e+06	1.122015e+06
emp_length_int	7912.0	5.917941e+00	3.538755e+00	0.50	3.000	6.05	1.000000e+01	1.000000e+01
home_ownership_cat	7921.0	2.092665e+00	9.486822e-01	1.00	1.000	3.00	3.000000e+00	3.000000e+00
annual_inc	7934.0	7.411045e+04	4.682275e+04	5000.00	45000.000	64942.00	9.000000e+04	1.036000e+06
income_cat	7923.0	1.194623e+00	4.357032e-01	1.00	1.000	1.00	1.000000e+00	3.000000e+00
loan_amount	7281.0	1.466940e+04	8.383141e+03	1000.00	8000.000	13000.00	2.000000e+04	3.500000e+04
term_cat	7912.0	1.302578e+00	4.594033e-01	1.00	1.000	1.00	2.000000e+00	2.000000e+00
application_type_cat	7946.0	1.000503e+00	2.243229e-02	1.00	1.000	1.00	1.000000e+00	2.000000e+00
purpose_cat	7929.0	4.862908e+00	2.392702e+00	1.00	3.000	6.00	6.000000e+00	1.300000e+01
interest_payment_cat	7924.0	1.473877e+00	4.993486e-01	1.00	1.000	1.00	2.000000e+00	2.000000e+00
loan_condition_cat	7921.0	7.688423e-02	2.664245e-01	0.00	0.000	0.00	0.000000e+00	1.000000e+00
interest_rate	7285.0	1.326614e+01	4.418158e+00	5.32	9.990	12.99	1.620000e+01	2.899000e+01
grade_cat	7914.0	2.808314e+00	1.326523e+00	1.00	2.000	3.00	4.000000e+00	7.000000e+00
dti	7921.0	1.825733e+01	8.303294e+00	0.00	12.030	17.82	2.418000e+01	4.856000e+01
total_pymnt	7923.0	7.514807e+03	7.875723e+03	0.00	1871.915	4861.08	1.063895e+04	5.680905e+04
total_rec_prncp	7924.0	5.722360e+03	6.605961e+03	0.00	1166.270	3201.29	8.000000e+03	3.500001e+04
recoveries	7942.0	4.494994e+01	3.789841e+02	0.00	0.000	0.00	0.000000e+00	1.187980e+04

Divisão da Base em Treino e Teste

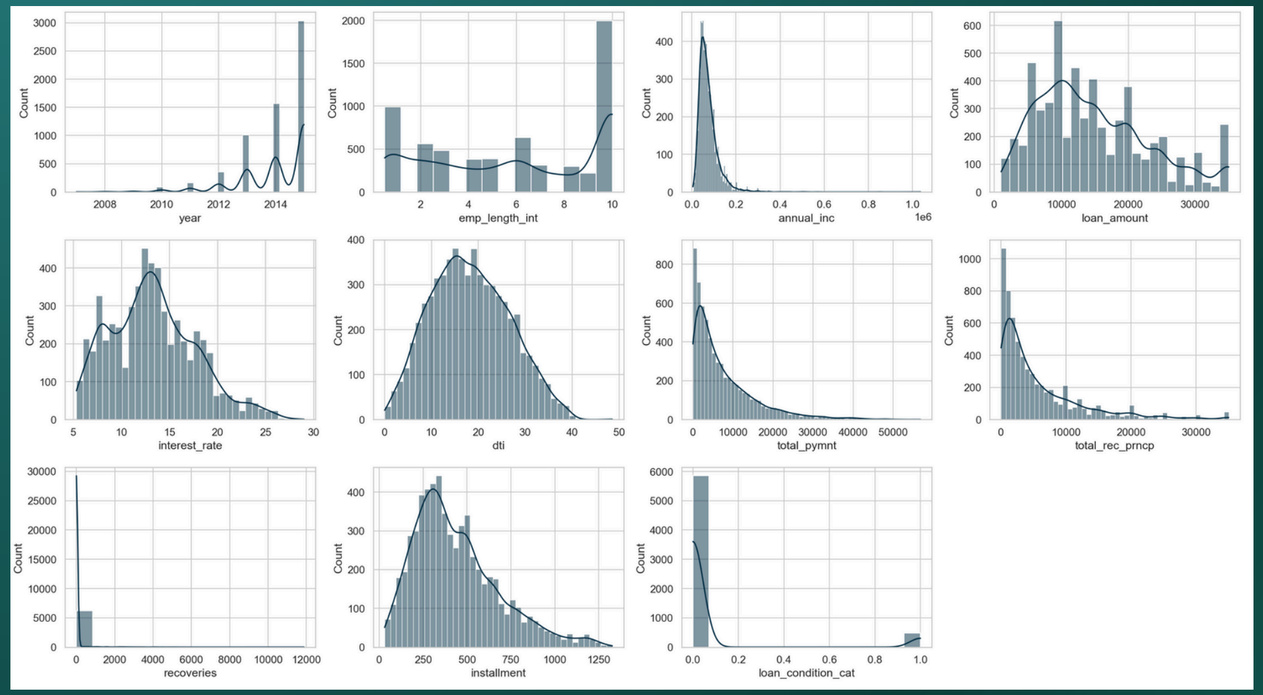
- O objetivo dessa separação é garantir que o modelo seja avaliado com dados que ele nunca viu antes, obtendo assim uma estimativa realista de desempenho.
- A divisão foi feita utilizando train_test_split, mantendo a proporção da variável alvo através do parâmetro stratify=y.
- O conjunto de teste recebeu 20% dos dados.

```
X = df.drop(columns=['loan_condition_cat'])
y = df['loan_condition_cat'].copy()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
```

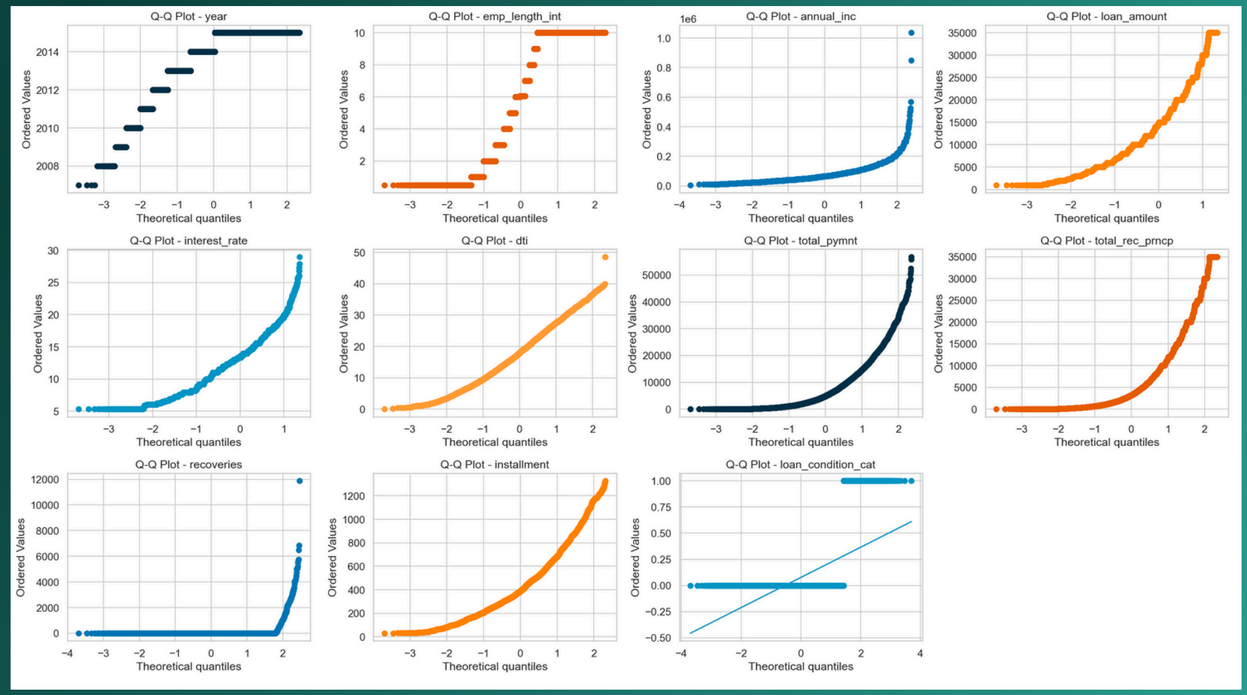
- O tamanho final de cada conjunto foi: preditores de treino com 6.336 amostras, alvo de treino com 6.336 amostras.
- Preditores de teste com 1.585 amostras e alvo de teste também com 1.585 amostras.

Distribuição das Variáveis

- Variáveis não seguem distribuição normal.
- Assimetrias positivas predominantes.
- Presença de valores extremos.



Histograma

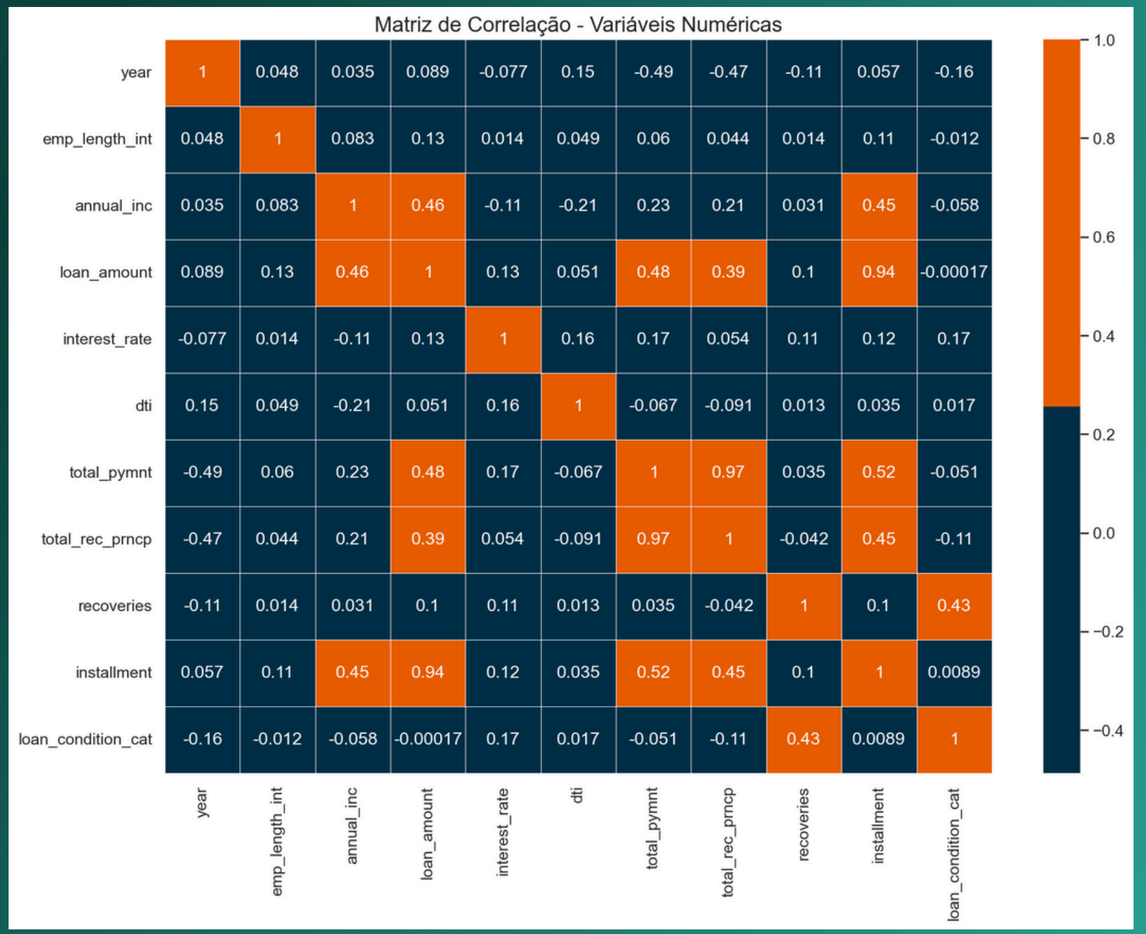


Q-Q plot

Correlações Importantes

- Forte relação entre loan_amount e installment.
- total_pymnt e total_rec_prncp altamente correlacionados.

O gráfico facilita identificar padrões, como grupos de variáveis fortemente relacionadas, além de confirmar que não há correlações excessivamente altas que indiquem multicolinearidade severa

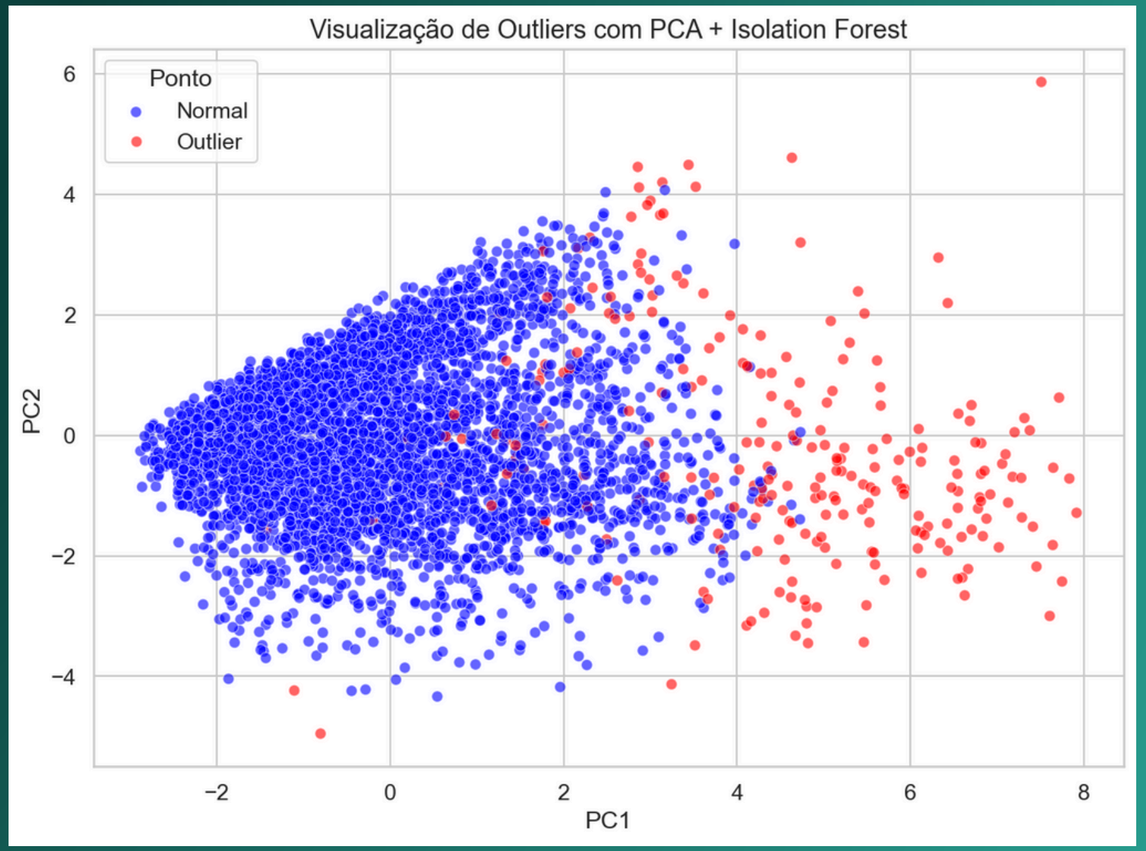


heatmap

Detecção de Outliers

Método utilizado: Isolation Forest.

- Outliers em annual_inc, total_pymnt, recoveries.
- year é a variável mais estável.
- Estratégia: manter outliers para não perder informação real.



Os dados foram convertidos para duas dimensões usando PCA, e cada ponto representa um registro do conjunto de dados. Os pontos azuis são valores normais e os vermelhos são outliers detectados pelo Isolation Forest.

Pré-processamento

```
num_cols = X_train.select_dtypes(include=['int64', 'float64']).columns.tolist()
cat_cols = X_train.select_dtypes(include=['object', 'category']).columns.tolist()
num_knn_cols = ["interest_rate"]
num_mean_cols = [col for col in num_cols if col not in num_knn_cols]
numeric_mean_transformer = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="mean"))
])
numeric_knn_transformer = Pipeline(steps=[
    ("imputer", KNNImputer(n_neighbors=5))
])
categorical_transformer = Pipeline(steps=[
    ("imputer", SimpleImputer(strategy="most_frequent"))
])
preprocessor = ColumnTransformer(
    transformers=[
        ("num_mean", numeric_mean_transformer, num_mean_cols),
        ("num_knn", numeric_knn_transformer, num_knn_cols),
        ("cat", categorical_transformer, cat_cols)
    ]
)
preprocessor.fit(X_train)

X_train_imputed = pd.DataFrame(
    preprocessor.transform(X_train),
    columns=num_mean_cols + num_knn_cols + cat_cols
)
X_test_imputed = pd.DataFrame(
    preprocessor.transform(X_test),
    columns=num_mean_cols + num_knn_cols + cat_cols
)
y_train_imputed = y_train.copy()
y_test_imputed = y_test.copy()
```

- Imputação média para numéricas e moda para categóricas.
- Uso adequado de fit/transform para evitar data leakage.
- Zero valores faltantes após tratamento.

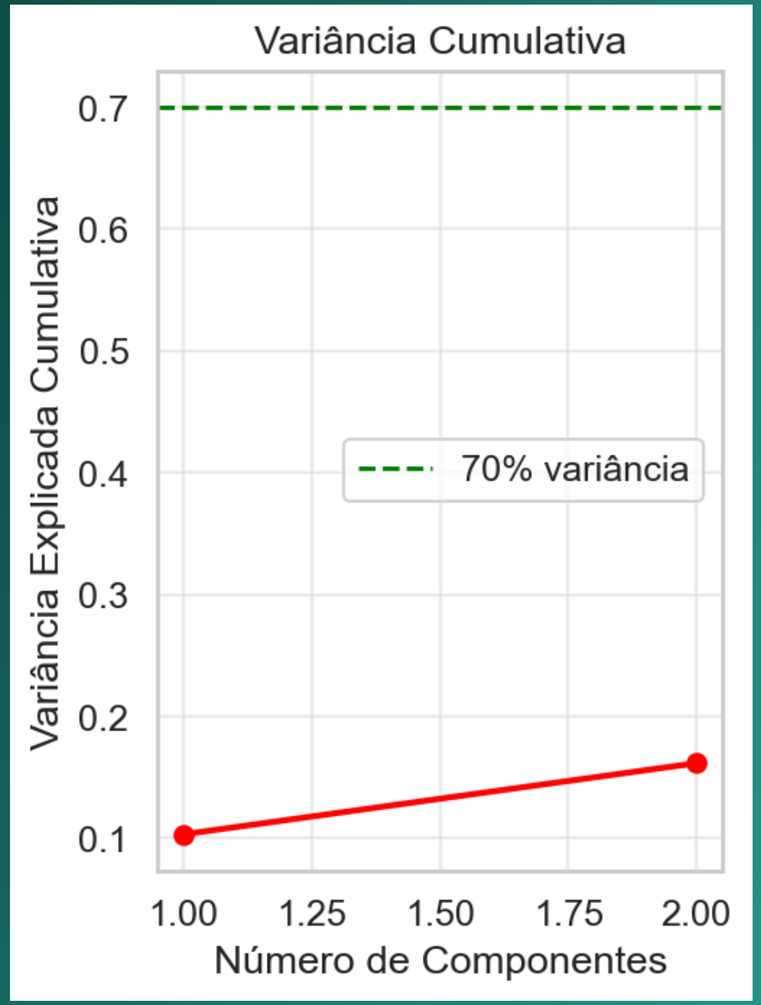
Durante o pré-processamento dos dados, primeiro foi separado as variáveis numéricas e categóricas.
o imputador usou apenas dados de treino.
Depois, foi aplicado o mesmo preenchimento ao conjunto de teste usando somente o transform, mantendo a consistência das estatísticas.
Por fim, foi reunida novamente as colunas imputadas e foram preservervadas variáveis-alvo sem modificações.

Transformações

```
num_cols_imputed = X_train_imputed.select_dtypes(include=['int64', 'float64']).columns
cat_cols_imputed = X_train_imputed.select_dtypes(include=['object', 'category']).columns

transformation_pipeline = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_cols_imputed),
        ("cat", OneHotEncoder(sparse_output=False, handle_unknown='ignore'), cat_cols_imputed)
    ]
)

X_train_transformed = transformation_pipeline.fit_transform(X_train_imputed)
X_test_transformed = transformation_pipeline.transform(X_test_imputed)

print(f"X_train_transformed: {X_train_transformed.shape}")
print(f"X_test_transformed: {X_test_transformed.shape}")
```

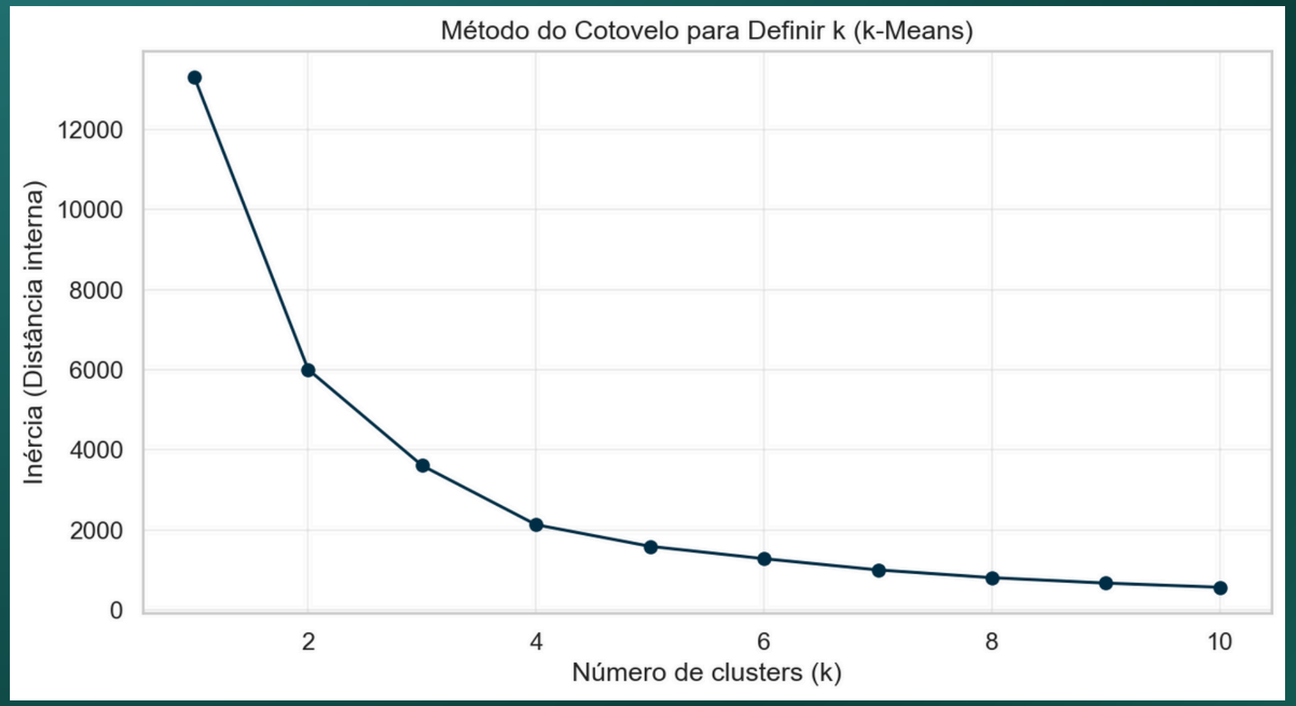
Redução Dimensional com PCA

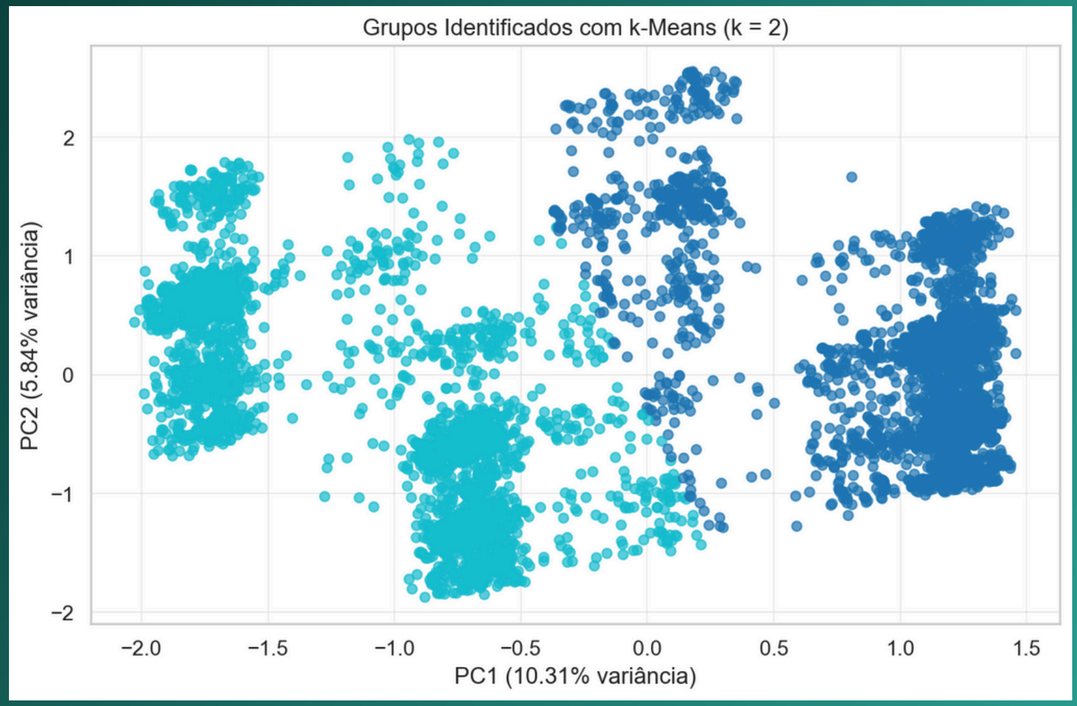
- PCA (2 componentes) representa apenas 17% da variância total.
- Apesar de baixa retenção, útil para visualização.
- Densidade indica formação de grupos estruturados.



Agrupamento (K-Means)

- Método do cotovelo sugeriu k = 3.
- Optou-se por k = 2, alinhado à classificação binária.
- Separação visual coerente no espaço PCA.





Modelagem

Modelos testados:

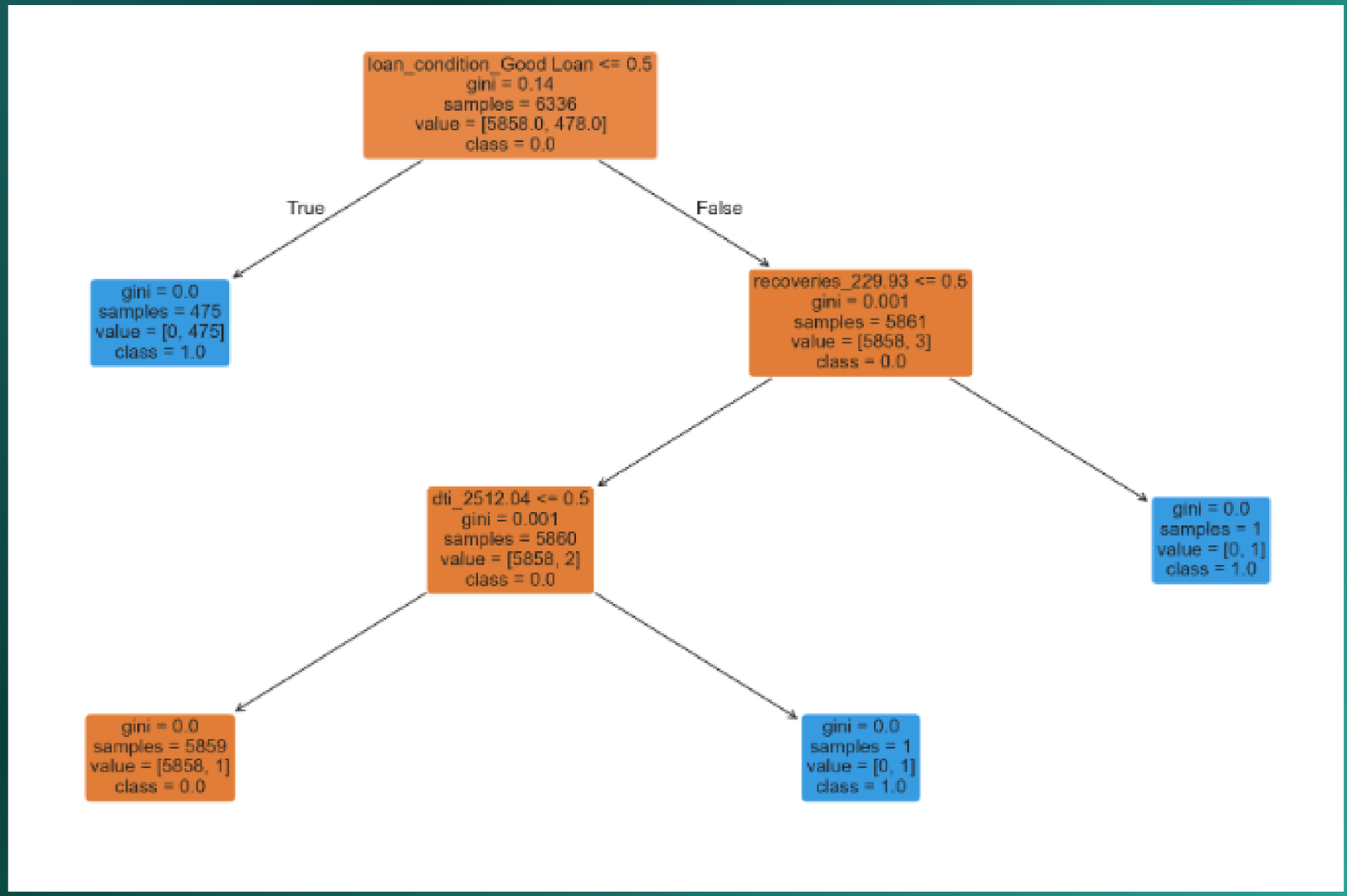
- SGDClassifier
- DecisionTreeClassifier
- KNN

Resultados:

- SGD obteve acurácia média 99,68%.
- DecisionTree obteve acurácia média 99,68%.
- KNN teve baixo desempenho (~41%).
- Modelos extremamente estáveis entre dobras.

Interpretação da Árvore de Decisão

- **Variáveis mais usadas: dtl e recoveries.**
- **grade e interest_rate não foram relevantes.**
- **H7 e H5 parcialmente rejeitadas.**



Conclusão das Hipóteses

H5: A condição final do empréstimo está maior associada ao dti, juros e renda. Conclusão: A árvore usa dtl (semelhante a dti) mas não usa juros (interest_rate) nem renda (income). Portanto, a H5 é parcialmente falsa, a condição do empréstimo parece estar mais associada a dtl e recoveries, não ao trio proposto.

H6: Modelos supervisionados podem prever inadimplência com alta precisão. Conclusão: A árvore mostra gini = 0.001 em nós profundos e folhas puras (gini = 0.0), indicando que o modelo consegue separar muito bem as classes de inadimplência. Isso suporta a H6, confirmando que modelos de árvore de decisão podem alcançar alta precisão na previsão de inadimplência.

H7: As variáveis mais relevantes para previsão devem ser: grade, interest_rate e dti. Conclusão: Na árvore fornecida, as variáveis usadas são dtl_2512.04 e recoveries_229.93, enquanto grade e interest_rate não aparecem. Isso não suporta a H7, pelo menos neste modelo, as variáveis mais relevantes são relacionadas a dívida (dtl) e valores recuperados, não às três listadas

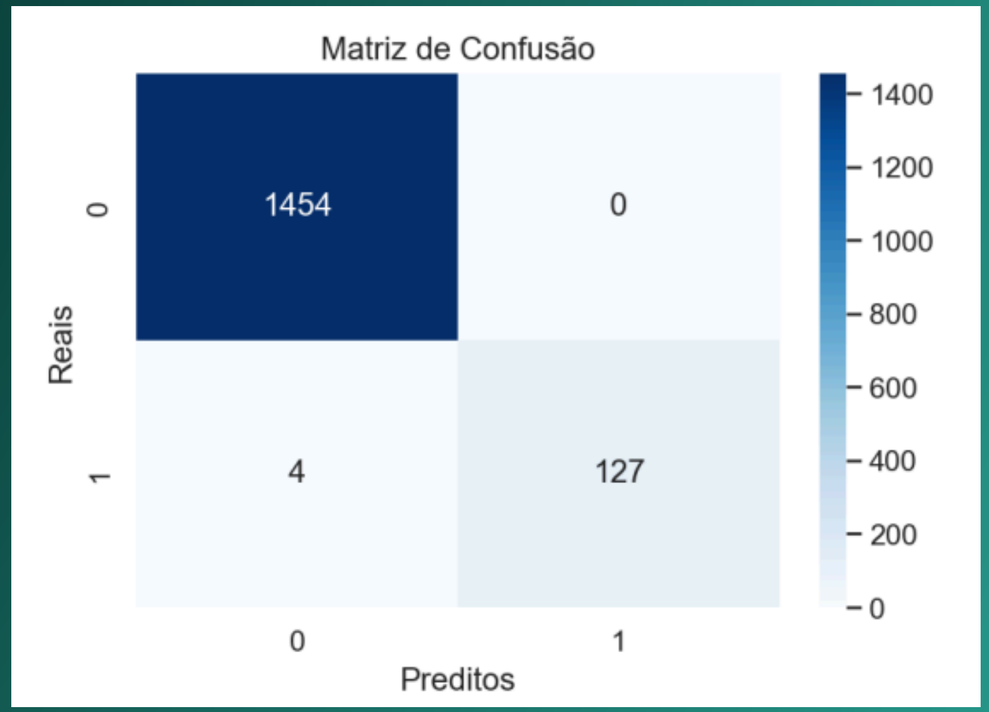
Avaliação Final no Conjunto de Teste

- **Acurácia próxima de 100%.**
- **Excelente separação das classes.**
- **Apenas 4 falsos negativos (muito baixo).**

```python
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

acuracia = accuracy_score(y_test, y_pred)
print(f"Acurácia: {acuracia}")

print(classification_report(y_test, y_pred))
```

```
Acurácia: 0.9974763406940063
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      1454
         1.0       1.00      0.97      0.98       131

    accuracy                           1.00      1585
   macro avg       1.00      0.98      0.99      1585
weighted avg       1.00      1.00      1.00      1585
```

Perfomace perfeita



Esses resultados reforçam a confiabilidade do modelo e confirmam que ele generaliza bem para dados novos.

Conclusões

Modelo supervisionado alcançou excelente performance com H6 confirmada.
Variáveis mais relevantes foram diferentes das esperadas.
O pipeline criado é robusto e aplicável para predição real.

Recomendações:

Ajustar política de crédito considerando dtl e recoveries.
Utilizar score contínuo para segmentação de risco.
Ampliar análises considerando dados mais recentes.