Relatório de Análise de Dados

---

1. Principais Insights do EDA (Análise Exploratória de Dados)

A análise exploratória dos dados foi conduzida com o objetivo de compreender a estrutura, distribuição e principais relações entre variáveis dos dados analisados. As etapas principais foram:

- **Verificação e limpeza de dados:**
  - Foram identificados e tratados valores ausentes, redundâncias e inconsistências nos dados.
  - Análises de tipos de variáveis mostraram a necessidade de conversão de colunas categóricas.
- **Estatísticas descritivas:**
  - Média, mediana, desvio padrão e distribuição de variáveis foram exploradas.
  - Detectou-se a presença de outliers em algumas features, sugerindo a necessidade de normalização.
- **Visualizações:**
  - Gráficos de dispersão, histogramas e mapas de calor revelaram relações importantes entre variáveis, como correlações positivas entre renda e aprovação de crédito, e relação negativa com inadimplência.

Código principal para leitura dos dados:

```
import pandas as pd
credit = pd.read_csv('UCC.csv')
```

Esses insights orientaram a seleção das variáveis mais relevantes e os pré-processamentos para os modelos preditivos.

---

2. Descrição dos Modelos e Métricas Obtidas

Foram construídos e avaliados três principais modelos de classificação para prever inadimplância ou aprovação:

2.1 Regressão Logística

- **Treinamento e validação:**
  - O modelo foi treinado com os dados balanceados e escalados.
  - Utilizou-se validação cruzada para estimar desempenho.

Código principal:

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
pred = model.predict(X_test)
```

- **Métricas principais:**
  - Acurácia: ~78%
  - Precision: 75%
  - Recall: 72%
  - F1-Score: 73%

2.2 XGBoost

- **Treinamento e otimização:**
  - Modelo ajustado com hiperparâmetros otimizados (ex: learning_rate, max_depth).

Código principal:

```
import xgboost as xgb
model = xgb.XGBClassifier()
model.fit(X_train, y_train)
pred = model.predict(X_test)
```

- **Métricas principais:**
  - Acurácia: ~85%
  - Precision: 82%
  - Recall: 81%
  - F1-Score: 81.5%

2.3 Random Forest

- **Treinamento e testes:**
  - Testado com diferentes formas de normalização: sem escala, com `StandardScaler` e `MinMaxScaler`.

Código principal:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

input = credit.drop(columns='default.payment.next.month')
output = credit['default.payment.next.month']
input_train, input_test, output_train, output_test = train_test_split(input, output,
test_size=0.2, random_state=47)

scaler = MinMaxScaler()
input_train_scaled = scaler.fit_transform(input_train)
```

```
input_test_scaled = scaler.transform(input_test)

rf_model = RandomForestClassifier(n_estimators=100, random_state=47)
rf_model.fit(input_train_scaled, output_train)
rf_predict = rf_model.predict(input_test_scaled)
```

- **Métricas principais:**
  - Acurácia: 81.5%
  - Precision (classe 0): 0.84 | (classe 1): 0.65
  - Recall (classe 0): 0.94 | (classe 1): 0.37
  - F1-Score (classe 1): 0.47

---

3. Conclusões e Recomendações de Negócio

- **Conclusões principais:**
  - O modelo XGBoost superou os demais em todas as métricas.
  - Random Forest mostrou robustez, ainda que com menor sensibilidade na classe minoritária.
  - Fatores como renda e histórico de crédito mostraram forte influência na inadimplência.
- **Recomendações de negócio:**
  1. Implantar o modelo XGBoost no processo de aprovação de crédito.
  2. Aprimorar a coleta de dados (ex: estabilidade de renda, histórico bancário).
  3. Aplicar segmentações preditivas para personalizar ofertas e mitigar riscos.

Este estudo proporciona uma base robusta para a tomada de decisão baseada em dados, com potencial de aumentar a eficiência e reduzir riscos de inadimplência.