

Case - Modelagem de Crédito - SCR Dados

Introdução

Neste case, queremos visualizar como você resolveria um problema de dados, de complexidade intermediária com SQL e Python.

O desafio consiste em pegar uma base mais produtiva e gerar uma base que possa ser utilizada de forma mais rápida em outros contextos.

O tempo médio de solução deste case é **6h**.

Case

O **Banco Central (BC)** mantém o **Sistema de Informações de Créditos (SCR)**, um instrumento de registro gerido pelo BC e alimentado mensalmente pelas instituições financeiras, que enviam quanto cada CPF possui de saldo devedor ou limite em diversas modalidades de crédito.

Quando um banco consulta o SCR, obtém as modalidades que um cliente possui dívida, e isto auxilia muito nas decisões de conceder ou não um crédito.

Links interessantes para maiores informações:

1. [Link para página do SCR \(https://www.bcb.gov.br/estabilidadefinanceira/scr\)](https://www.bcb.gov.br/estabilidadefinanceira/scr)
2. [Instruções para preenchimento do Documento 3040 \(https://www.bcb.gov.br/content/estabilidadefinanceira/Leiaute_de_documentos/scrdoc3040/SCR_InstrucoesDePreenchimento_Doc3040.pdf\)](https://www.bcb.gov.br/content/estabilidadefinanceira/Leiaute_de_documentos/scrdoc3040/SCR_InstrucoesDePreenchimento_Doc3040.pdf)
3. [Link para referência de modalidades \(https://www.bcb.gov.br/content/estabilidadefinanceira/Leiaute_de_documentos/scrdoc3040/SCR3040_Leiaute.xls\)](https://www.bcb.gov.br/content/estabilidadefinanceira/Leiaute_de_documentos/scrdoc3040/SCR3040_Leiaute.xls) (ver a planilha Anexo, a tabela Anexo 3)

O que você está recebendo

Você está recebendo o arquivo **base_bacen.sqlite**. Este é um banco de dados SQLite com uma única tabela, **scr**. Esta tabela possui o schema abaixo:

coluna	tipo	descricao
chave_cpf	string	Hash que identifica um cliente.
data_consulta_dado_bacen	string	Mês da consulta, no formato YYYY-MM
codigo_modalidade_operacao	int	Modalidade de crédito, no formato xxyy , onde xx indica o domínio e yy indica o subdomínio. Veja a tabela abaixo.
valor_credito_vencer_ate_30_dia	float	Saldo a vencer no prazo de 0 a 30 dias.
valor_credito_vencer_31_60_dia	float	Saldo a vencer no prazo de 31 a 60 dias.
valor_credito_vencer_61_90_dia	float	Saldo a vencer no prazo de 61 a 90 dias.
valor_credito_vencer_acima_90_dia	float	Saldo a vencer num prazo acima de 90 dias.
valor_credito_vencido_15_30_dia	float	Saldo vencido, de 0 a 30 dias.
valor_credito_vencido_31_60_dia	float	Saldo vencido, de 31 a 60 dias.
valor_credito_vencido_61_90_dia	float	Saldo vencido, de 61 a 90 dias.
valor_credito_vencido_acima_90_dia	float	Saldo vencido a mais de 90 dias.

Neste case, estamos simulando uma tabela com estes campos. Estamos enviando as seguintes modalidades:

Modalidade	Domínio	Subdomínio	Descrição
202	02	02	consignado
101	01	01	cheque_especial
213	02	13	cheque_especial
204	02	04	cartao_revolver
210	02	10	cartao_revolver
218	02	18	cartao_revolver

Modalidade	Domínio	Subdomínio	Descrição
406	04	06	cartao_revolver
1304	13	04	cartao_transactor
499	04	99	outros

O que queremos que você prepare

Um arquivo csv, chamada **book_scr.csv**, contendo uma tabela que terá suas colunas utilizadas como variáveis explicativas de modelos de crédito.

Esta tabela deve ter o seguinte layout:

1. Cada linha representa um CPF / data_consulta_dado_bacen;
2. Colunas descrevendo o saldo de cada faixa de atraso para cada produto; por exemplo: cartao_trasactor_saldo_vencer_0_30_dias
3. Criação de faixas de atraso maiores; por exemplo, 0 a 60 dias, para cada produto;
4. Quaisquer combinações de variáveis que você considerar útil para construção de um modelo de crédito.

Exemplo (Primeiras colunas apenas):

chave_cpf	data_consulta_dado_bacen	cheque_especial_valor_credito_vencer_ate_30_dia	cheque_especial_valor_credito_vencer_31_60_dia	cheque
10003023342	2022-06	0	0	
10003023342	2022-07	0	0	
10003023342	2022-08	0	0	
10004431332	2022-06	0	0	
1000721890	2022-06	0	0	

O que esperamos

1. Demonstre seu conhecimento de SQL e Python

Utilize, dentro do possível, seu conhecimento de SQL e Python.

Pontos de atenção:

1. Não recomendamos o uso de ORMs.
2. Dê preferência a tratar os dados em SQL: a tabela original possui mais de 5 bilhões de linhas, não sendo possível tratar em Pandas.

2. Apresente seu case organizado

Queremos ver seu código. Não é necessário preparar apresentação PowerPoint.

Organize seus scripts, notebooks e queries para que seja possível entender o que está sendo feito.

3. Sugestão de próximos passos

Dado que seu case foi aceito, o que você sugere de próximos passos?

O que será valorizado

1. Utilize funções e classes para facilitar seu trabalho

Você está recebendo uma tabela pequena, amostral. Como indicado antes, a tabela original possui mais de 5 bilhões de linhas, e geramos mais de 2000 colunas.

É possível, mas inviável escrever na mão o SQL. Assim, na medida do possível, crie funções e classes para facilitar seu trabalho, conforme a necessidade. Não complique mais que o necessário.

Dicas

1. Utilize as funcionalidades do Python para gerar o SQL

Se surgir uma nova modalidade, como você incluiria ela em minutos dentro do processo?

2. Analise a base que você está trabalhando

Considere que a base pode conter erros e coisas desconhecidas. Sugira o que fazer em cada situação.