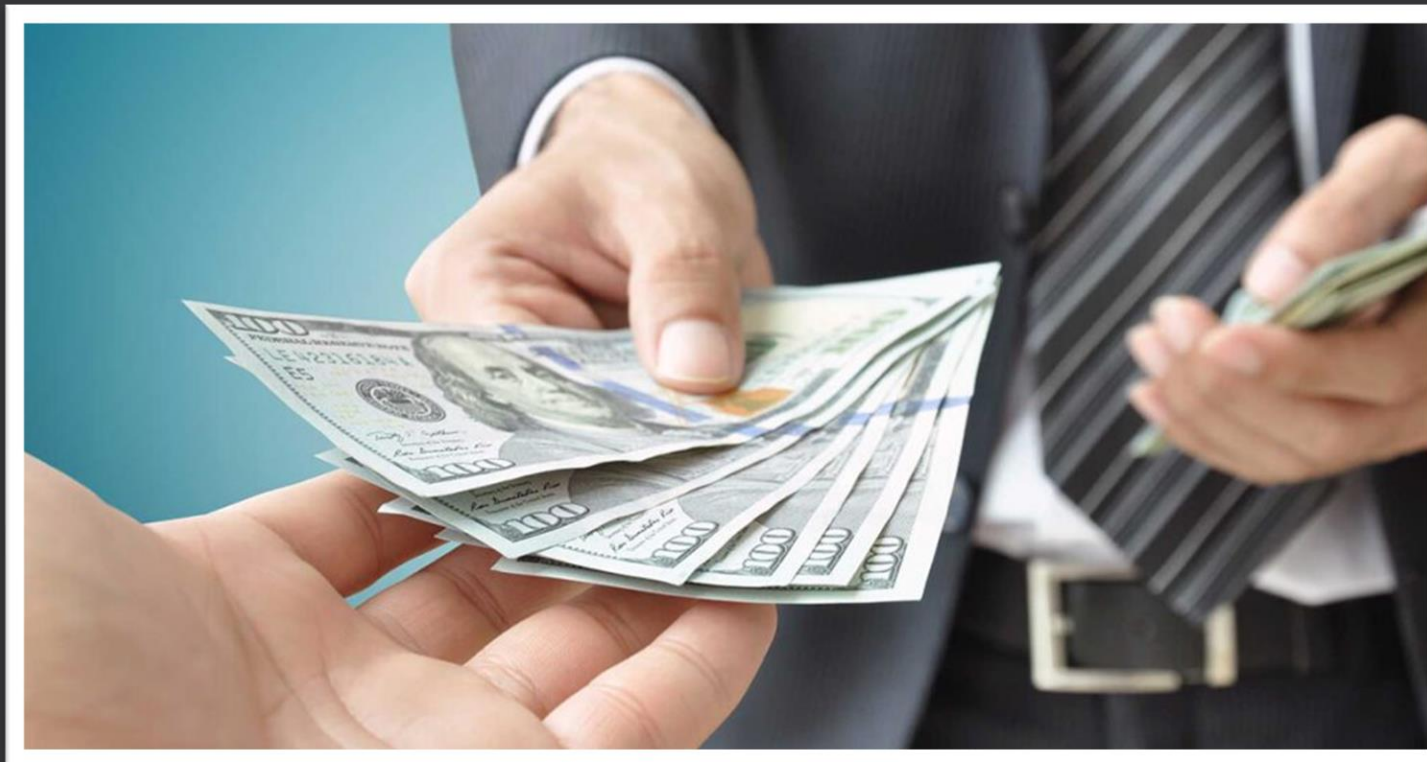


# Classificação de assinatura eletrônica de empréstimos com base no histórico financeiro



# Componentes da equipe

- Carlos Souza
- Rafael Kanda
- Saile Santos

# Problema

Com base no dataset: E-Signing of Loan-Based on Financial History realizar a classificação e previsão se um dado indivíduo concluiu o processo de assinatura eletrônica ou não.



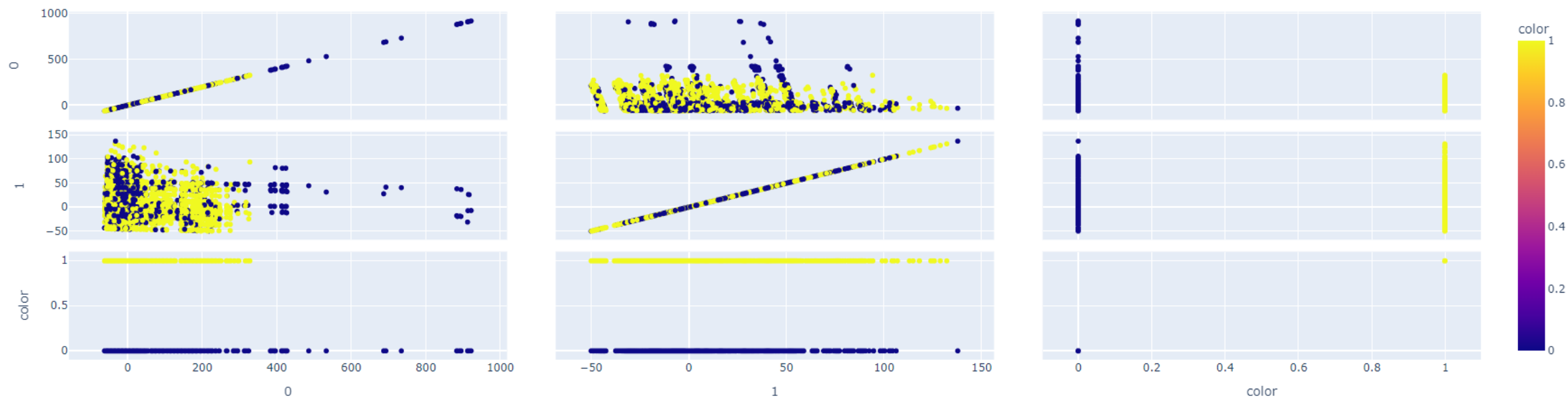
# Preparo do dataset

- Eliminação de atributos com correlações insignificantes;
- Aplicação de OneHotEncoder para lidar com colunas em tipo String;
- Agrupamento de colunas com mesmas informações;
- Redução da variação.



# Aplicação do PCA

- Análise da representatividade dos componentes;
- Restrição de 2 componentes;
- Variância capturada: 95.88%



# Análise do notebook da solução

- Verificação do funcionamento;
- Tratamento do dataset;
- Separação dos dados;
- Criação de função de ativação;
- Classes para criação da rede;
- Compilação e treinamento;
- Perda e acurácia nos treinos e validação;
- Métricas de avaliação.



# Resultados do notebook

	Precision	Recall	F1-Score
0	0.5361	0.5699	0.5525
1	0.5988	0.5655	0.5817
Accuracy			0.5676

# Modificação do notebook

- Implementação do dataset modificado;
- Alteração do tamanho dos filtros iniciais para 40 (antes 32);
- Alteração do otimizador para o RMSprop (anterior 'adam');
- Alteração do número de épocas para 20 (antes 10);
- Alteração do batch\_size para 40 (anterior era 32).

```
[22] pca = PCA(n_components=2)
      components = pca.fit_transform(data)

      total_var = pca.explained_variance_ratio_.sum() * 100
      print('Total da variância capturada: ', total_var)

      labels = {
          str(i): f"PC {i+1} ({var:.1f}%)"
          for i, var in enumerate(pca.explained_variance_ratio_ * 100)
      }
```

```
[60] net = ResMLP(initial_filters=40, block_list=[2, 2, 2], num_classes=2)

      net.compile(optimizer='rmsprop',
                  loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
                  metrics=['sparse_categorical_accuracy'])

      history = net.fit(train_ds, epochs=20, batch_size=40, validation_data=test_ds)

      net.summary()
```



# Resultados do notebook modificado

	Precision	Recall	F1-Score
0	0.5411	0.6156	0.5760
1	0.6145	0.5399	0.5748
Accuracy			0.5754

# Comparação dos resultados

	Original	Modificado
Precision - 0	0.5361	0.5411
Precision - 1	0.5988	0.6145
Recall - 0	0.5699	0.6156
Recall - 1	0.5655	0.5399
F1-Score - 0	0.5525	0.5760
F1-Score - 1	0.5817	0.5748
Accuracy	0.5676	0.5754

# Conclusão

As alterações realizadas não foram suficientes para conseguir resultados muito diferentes do original, havendo apenas uma ligeira melhora em relações as métricas verificadas, com exceção de uma que possuiu um desempenho pior do que a original.

# Links dos notebooks

Dataset:

- [https://drive.google.com/file/d/1NSFD\\_7AtVZn4CXdnEPwKPhqS2Wx8bIIo/view?usp=share\\_link](https://drive.google.com/file/d/1NSFD_7AtVZn4CXdnEPwKPhqS2Wx8bIIo/view?usp=share_link)

Solução original:

- [https://drive.google.com/file/d/1HI6a6Wb27wJVDmDO0678PdmTiYuvCLfA/view?usp=share\\_link](https://drive.google.com/file/d/1HI6a6Wb27wJVDmDO0678PdmTiYuvCLfA/view?usp=share_link)

Solução modificada:

- [https://colab.research.google.com/drive/19jV6jTKgo-Emq2Lc7vaG917joCQP1fdR?usp=share\\_link](https://colab.research.google.com/drive/19jV6jTKgo-Emq2Lc7vaG917joCQP1fdR?usp=share_link)

