Análise de crédito baseada em floresta aleatória

Raul Victor de O. P.*, Roni Vial C. Jr.*, Elton M. S.*, Claudio Henrique de O.* e Sandro C. M.*

**Especialização em Ciência de Dados, Universidade Ateneu (UniAteneu), Fortaleza-CE, Brasil,*
E-mail:{raul.ofc.fs@gmail.com, roni.ciribelli@outlook.com, eltonsaraiva@hotmail.com,
claudiohenriquedm@gmail.com, sandro.mesquita@professor.uniateneu.edu.br}

Resumo

A avaliação de crédito torna-se essencial em períodos cruciais para a estabilidade financeira. A busca por sistemas confiáveis reflete o empenho em unir aspirações e realidade. Soluções de análise de crédito usando aprendizado de máquina são um avanço promissor em direção à acessibilidade, para aqueles que buscam aprovação de crédito. Neste sentido, o presente trabalho tem como objetivo apresentar uma aplicação de análise de crédito baseado em aprendizagem de máquina. O produto consta com o uso do *framework* `Django` para comunicar a interface gráfica com núcleo em `Python`, para acessar o modelo de aprendizado de máquina baseado em floresta aleatória.

***Palavras-chave*—** Ciência de dados, aprendizagem de máquina, Django, empréstimo.

1 Introdução

Em uma época em que a estabilidade financeira é fundamental, a análise de crédito desempenha um papel vital na vida das pessoas. Logo, a busca por um sistema de avaliação preciso e confiável reflete o compromisso de construir pontes sólidas entre sonhos e realidade. Neste sentido, produtos baseados em aprendizagem de máquina que oferecem análise de crédito representam um interessante passo em direção à acessibilidade para todos os que aspiram uma carta de aprovação de crédito, na Índia nesta ocasião. Com cada avaliação, o caminho para um mercado imobiliário mais inclusivo e sustentável é pavimentando, em que o potencial de cada indivíduo encontra sua devida consideração. O presente trabalho tem como objetivo apresentar um sistema de avaliação de crédito baseado em aprendizado de máquina, a partir do treinamento de um modelo.

O resto do trabalho está organizado como segue. Na seção 2, a ferramenta para a aplicação *web* `Django` utilizada no produto é apresentada. A modelagem baseada em floresta aleatória é abordada na seção 3. Na seção 4 é apresentado o fluxograma utilizado no projeto. Finalmente, as conclusões são apresentadas na seção 5.

2 *Framework* para aplicação *web*: `Django`

O `Django` é um *framework* web Python de alto nível que permite o rápido desenvolvimento de sites seguros e de fácil manutenção [1] . Construído por desenvolvedores experientes, o `Django` cuida de grande parte do trabalho de desenvolvimento web, para que você possa se concentrar em escrever seu aplicativo sem precisar reinventar a roda. É gratuito e de código aberto, tem uma comunidade próspera e ativa, ótima documentação e muitas opções de suporte gratuito e pago.

Aplicativos web feitos com `Django` geralmente agrupam o código que manipula cada uma dessas etapas em arquivos separados:

- Localizador Uniforme de Recursos, do inglês *Uniform Resource Locator* (URL): Embora seja possível processar solicitações de cada URL por meio de uma única função, é muito mais simples fazer a manutenção do código escrevendo uma função *view* separada para manipular cada

10.5281/zenodo.10850225

recurso. Um mapeador de URLs é usado para redirecionar solicitações Protocolo de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol* (HTTP) para a *view* apropriada com base na URL da solicitação. O mapeador de URLs também pode corresponder padrões específicos de *strings* (cadeia de caracteres) ou dígitos que aparecem em um URL e transmiti-los a uma função *view* como dados [1].

- *View*: Uma *view* é uma função manipuladora de solicitações, que recebe solicitações HTTP e retorna respostas HTTP. As *views* acessam os dados necessários para satisfazer solicitações por meio dos *models* e encarregam a formatação da resposta aos *templates* [1].

- *Models*: Modelos são objetos em `Python` que definem a estrutura dos dados de um aplicativo, e fornecem mecanismos para gerenciar (adicionar, modificar e excluir) e consultar registros no banco de dados [1].

- *Templates*: Um *template* é um arquivo de texto que define a estrutura ou o layout de um arquivo (como uma página Linguagem de Marcação de Hipertexto, do inglês *HyperText Markup Language* (HTML)), com espaços reservados usados para representar o conteúdo real. Uma *view* pode criar dinamicamente uma página HTML usando um *template* HTML, preenchendo-a com dados de um *model*. Um *template* pode ser usado para definir a estrutura de qualquer tipo de arquivo; não precisa ser HTML [1].

O `Django`, por sua vez refere a essa organização como uma arquitetura nomeada Modelo de Visualização, do inglês *Model View Template* (MVT).

Por se tratar de uma arquitetura MVT, o `Django` foi escolhido para compor o presente projeto, uma vez que este possui várias facilidades. O `Django`, por sua vez, foi utilizado para a criação de uma página que realiza a leitura dos campos de um usuário e faz uma consulta no modelo de referência, em que é retornado a aprovação ou não do crédito. O modelo de referência para consulta é descrito na seção 3.

3 Modelagem

Na presente seção, é descrita a técnica de modelagem utilizada assim como a sua avaliação. A seção atual está organizada como segue. Primeiramente, na subseção 3.1, o algoritmo de árvore de decisão é apresentado para melhor compreensão da modelagem com floresta aleatória, que é apresentada em seguida, na mesma subseção. Na subseção 3.2, a avaliação do modelo utilizado no produto é apresentada em termos de acurácia, precisão, sensibilidade e F1.

3.1 Floresta aleatória

Árvore de decisão é um método de aprendizagem supervisionado não paramétrico para classificação ou regressão [2], [3]. O método da árvore de decisão é comumente conhecido por se tratar de uma técnica de aprendizagem indutiva. Em síntese, uma árvore é construída através de nós de decisão e terminal(ou folha), e ramificações que conectam outros nós, sejam de decisão ou folha, como a figura 1 ilustra. O primeiro nó (ou nó raiz) é exibido no topo, conectado por ligações ou ramificações sucessivas (direcionais) a outros nós. Esses são conectados de forma semelhante até alcançar os nós folhas, que não têm mais ligações [2]. A classificação de um padrão particular é iniciada no nó raiz, que questiona o valor de uma propriedade específica do padrão. As diferentes ligações do nó raiz correspondem aos diferentes possíveis valores. Baseado na resposta, é seguido um caminho apropriado para um nó subsequente. O próximo passo é realizar a decisão no nó subsequente apropriado, que pode ser considerado o nó raiz de uma subárvore [2] (veja figura 1). Dessa maneira é continuado até alcançar um nó folha, onde não houver mais nenhum questionamento. No caso, cada nó folha indica um rótulo de classe e o padrão de teste é atribuído à classe do nó folha alcançado [2].

10.5281/zenodo.10850225

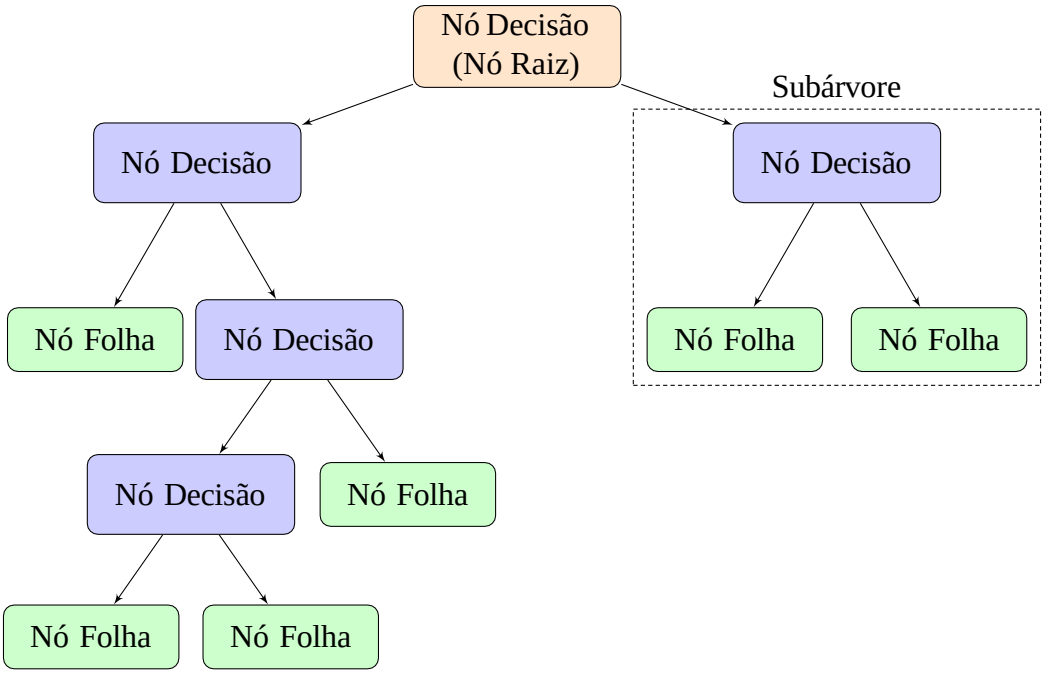

Figura 1: Representação da estrutura de uma árvore de decisão.

Nesse sentido, o percurso formado do nó raiz até a folha representa uma regra de classificação. A seleção da sequência de atributos mais adequados emprega conceitos como entropia H e ganho G [2], [3]. A medida de qualidade da separação para o calculo do ganho de informação de Shannon, a entropia H, que representa a medida da impureza ou o nível de desordem em um conjunto de dados de treinamento, é definida como segue

$$H(x) = -\sum_{k=1}^{\Sigma} p_k \log_2 p_k, \tag{1}$$

com $H(x)$ denotando a impureza e p_k a proporção de amostras pertencentes a classe k.

Em síntese, a construção de uma árvore de decisão é orientada pelo objetivo de reduzir a entropia. Quando $H \to 0$, significa que o grau de impureza (desordem) entre os dados de treino e ajustados se aproxima da nulidade, indicando bons ajustes por parte do modelo. Enquanto $H \to 1$ implica dizer que há bastante impureza no dado de treinamento, indicando mau ajuste e nesse caso é necessário dividir as instâncias para reduzir a impureza [2]. Logo, para medir o ganho de informação de um atributo A em relação a um conjunto x é dado por

$$G(x,A) = H(x) - \sum_{v \in V(A)} \frac{x_v}{x} H(x_v), \tag{2}$$

em que $V(A)$ é o conjunto de todos os valores possíveis para o atributo A e x_v o subconjunto de x no qual o atributo A possui valor v. O ganho de informação é a redução esperada da entropia, servindo como uma medida quantitativa que avalia o quão eficaz um determinado atributo separa os exemplos de treinamento de acordo com a classificação alvo. Em suma, ele indica a capacidade de um atributo em discriminar as classes [2], [3]. Com as informações a respeito do algoritmo de árvore de decisão, é possível compreender melhor sobre floresta aleatória.

Floresta aleatória é um algoritmo de aprendizado supervisionado que pode ser usado para problemas de regressão e classificação. Tal técnica é conhecida por se tratar de uma aprendizagem por agrupamento (*ensemble learning*), que reúne um conjunto de modelos de árvores de decisão. No geral, como representado na figura 2, o algoritmo faz isso ao selecionar amostra de dados de maneira aleatória, adquirir as predições para cada árvore e eleger a melhor solução por voto, para problemas de classificação. É válido ressaltar que a robustez do modelo está diretamente relacionada a quantidade de árvores de decisão [4].

10.5281/zenodo.10850225

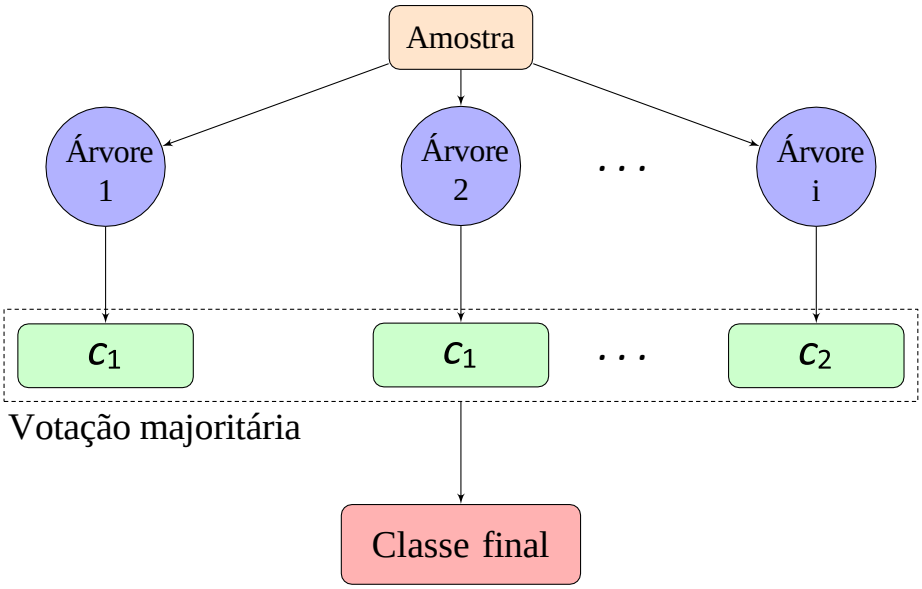

Figura 2: Representação básica da estrutura de uma floresta aleatória.

Como ilustrado na figura 2, em caso de classificação, a técnica de aprendizagem por agrupamento floresta aleatória funciona da seguinte maneira:

1. Amostras de um conjunto de dados são selecionadas aleatoriamente;
2. Para cada amostra, é criada uma árvore de decisão, e a previsão de cada árvore é obtida;
3. É realizada uma votação para cada resultado de predição;
4. O resultado da predição mais votada é selecionado como o resultado final.

Grandes quantidades de atributos (ou campos) do conjunto de dados permite que a modelagem com floresta aleatória seja mais eficaz, pois quanto maior a quantidade de atributos de um conjunto de dados, melhor é a filtragem para a tomada de decisão final, uma vez que esta depende da votação majoritária entre as classes de cada árvore de decisão (observe na figura 2).

3.2 Avaliação de desempenho do modelo de floresta aleatória utilizado no produto

A base de dados utilizada no presente trabalho está disponível em [5]. Inicialmente, a base de dados com 4269 entradas e 12 atributos foi dividida em duas partes, em que 80% da base de dados original foi separada para treinamento e 20% para teste (avaliação). Os doze atributos mencionados são evidenciados e descritos na tabela 1.

O motivo da escolha do modelo de floresta aleatória foi devido a grande quantidade de campos que o modelo pode acessar a partir do conjunto de dados utilizado para fazer uma previsão, pois quanto mais campos houver, mais complexo a floresta aleatória se torna e isso é eficaz de certo modo para a filtragem do resultado final.

Para a modelagem com floresta aleatória, foram utilizadas 100 árvores e critério entropia (veja a equação (1)) para a estimação do ganho entre os nós de decisão.

10.5281/zenodo.10850225

Tabela 1: Descrição da base de dados utilizada para a modelagem.

Campo	Descrição	Tipo da entrada	Tratamento no tipo da entrada
Número de dependentes	Quantidade de dependentes do cliente.	`int`	–
Ensino superior (grau de instrução)	Cliente possui ensino superior.	`bool`	`int` (binário)
Trabalhador avulso (situação)	Cliente é trabalhador avulso.	`bool`	`int` (binário)
Renda anual	Renda anual do cliente.	`float`	–
Valor do empréstimo	Valor desejado para o empréstimo.	`float`	–
Prazo do empréstimo	Prazo estipulado para o pagamento do empréstimo em anos.	`int`	–
Pontuação CIBIL / crédito	Pontuação de crédito.	`int`	–
Valor de ativos residenciais	Valor de ativos residenciais.	`float`	–
Valor de ativos comerciais	Valor de ativos comerciais.	`float`	–
Valor de ativos de luxo	Valor de ativos de luxo.	`float`	–
Valor de ativos bancários	Valor de ativos bancários.	`float`	–
Alvo: Aprovação do crédito	Campo alvo / resultado final da pesquisa por aprovação de crédito.	`int` (binário)	`string`

Na figura 3, são apresentados os resultados da avaliação de desempenho do modelo em questão.

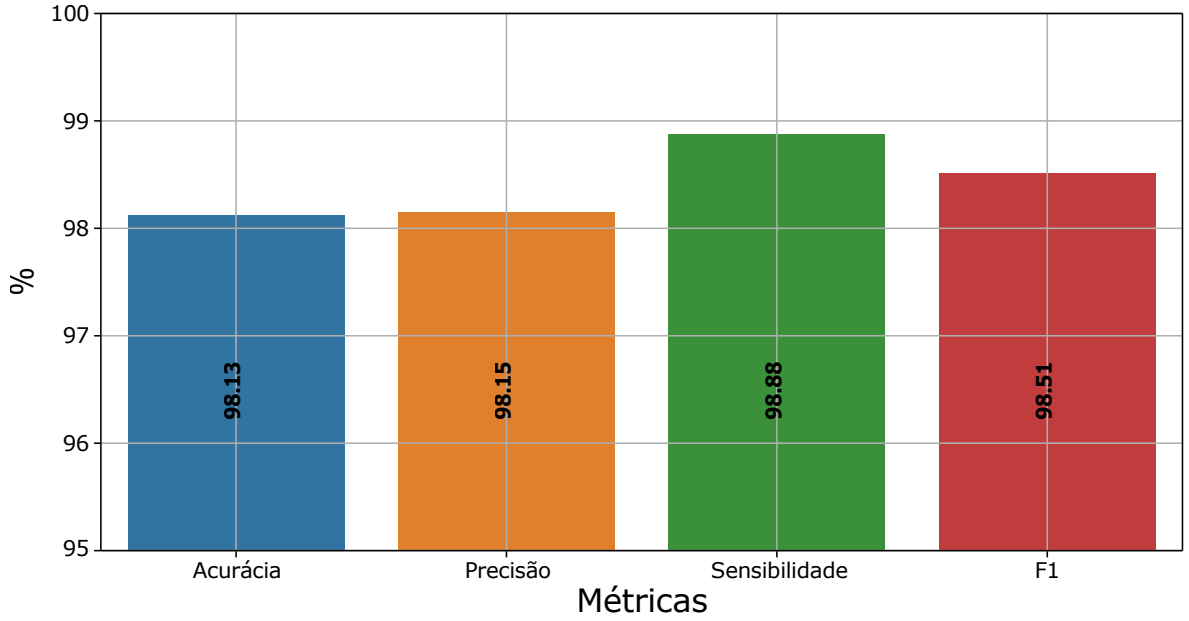

Figura 3: Avaliação do modelo de floresta aleatória em termos de acurácia, precisão, sensibilidade e F1. Note que o eixo da porcentagem está entre 95% e 100% para melhor visualização dos resultados.

A métrica de acurácia revela muito pouco a respeito do desempenho do modelo, por se tratar de um resultado generalista. Logo, com as métricas de precisão, sensibilidade e F1, é possível avaliar melhor o modelo em termos desempenho de classificação em cada classe. A partir desses resultados é possível

10.5281/zenodo.10850225

observar que o modelo classifica 98,15% (em laranja) como correto, dentre todas as classificações de classe crédito aprovado que o modelo realizou. O modelo, por outro lado, classifica 98,88% (em verde) corretamente, dentre todas as situações de classe aprovado como valor esperado. A avaliação em termos de pontuação F1 torna-se dispensável por se tratar de uma média harmônica entre as métricas precisão e sensibilidade.

4 Fluxograma do projeto

A presente seção tem como objetivo expor as etapas aplicadas no projeto. Através de um fluxograma na figura 4, as etapas do processo de análise de crédito são apresentadas.

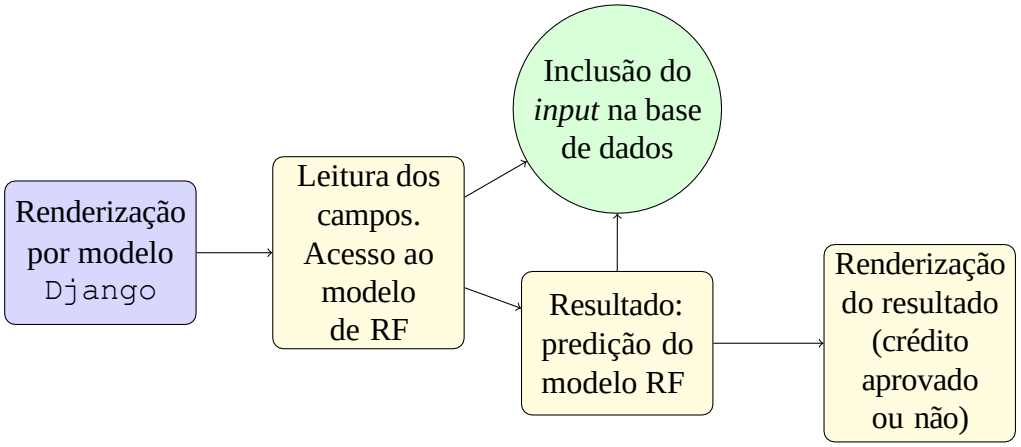

Figura 4: Fluxograma do produto.

A partir da figura 4, segue uma breve explicação de cada passo, respectivamente:

1. Renderização dos campos utilizando um MVT fornecido pelo `Django` *framework*. O HTML do formulário é representado na figura 5;

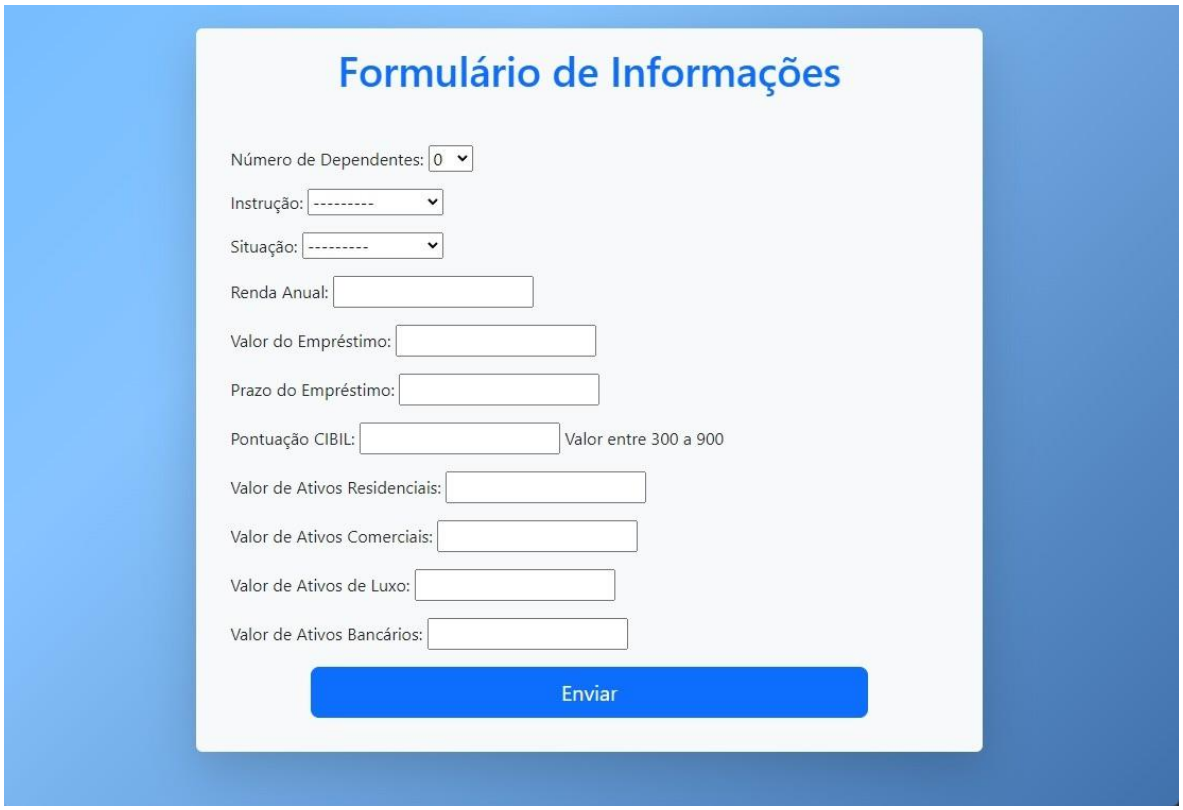

Figura 5: Renderização do formulário de informações requisitadas para a análise de aprovação de crédito. Note os onze campos solicitados para preenchimento.

10.5281/zenodo.10850225

2. A leitura dos campos é realizada a partir do preenchimento dos mesmos por parte do usuário;
3. Inclusão dos dados fornecidos pelo o usuário assim como a predição (resultado) realizada pelo modelo de Floresta Aleatória, do inglês *Random Forest* (RF);
4. Renderização final do resultado dado pelo modelo de RF. O HTML do formulário preenchido é representado na figura 6.

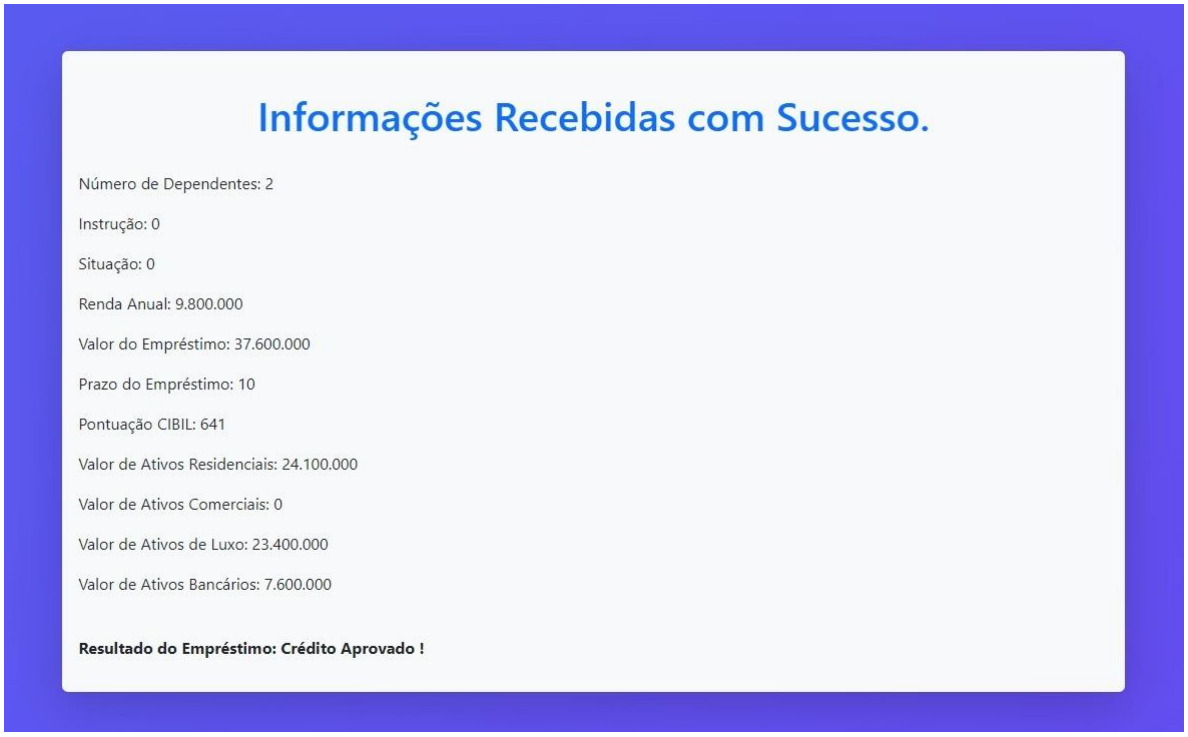

Figura 6: Renderização do formulário preenchido para a análise de aprovação de crédito e o resultado final.

A partir do fluxograma mostrado na figura 4, é possível entender, de maneira simplificada, o funcionamento do projeto. No geral, o modelo de floresta aleatória é consultado sempre que houver preenchimento de dados no formulário de informações retornando um resultado, que é a previsão baseada nos resultados apresentados na seção 3, com 98,15% de precisão e 98,88% de sensibilidade.

5 Conclusão

Foi criado um sistema de consulta de crédito, em que um modelo baseado em RF foi aplicado para gerar o resultado da consulta. O modelo baseado em floresta aleatória utilizado revela desempenho satisfatório, apresentando 98,15% de precisão e 98,88% de sensibilidade. O *framework* `Django` foi utilizado para prover a conexão, renderização e coleta dos dados do usuário.

Fica como perspectiva a implementação de melhorias na modelagem a partir de uma busca de parâmetros ótimos, assim como melhorias na apresentação do produto, ou seja, na interface do formulário de informações em HTML.

Referências

[1] Django Software Foundation, *Django*, versão 2.2, 5 de mai. de 2019. endereço: `https://djangoproject.com`.

[2] R. O. Duda, P. E. Hart e D. G. Stork, *Pattern classification*. John Wiley & Sons, 2002.

[3] C. M. Bishop e N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.

10.5281/zenodo.10850225

[4] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc.", 2022.

[5] A. Vidhya, *Loan Prediction*, data retrieved from Analytics Vidhya, `https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/`, 2016.

10.5281/zenodo.10850225