

# Projeto Final de Programação

Silvio Alonso

Junho 2021

# Conteúdo

<b>1</b>	<b>Contexto</b>	<b>3</b>
<b>2</b>	<b>Especificação</b>	<b>3</b>
2.1	Público alvo . . . . .	3
2.1.1	Persona . . . . .	3
2.2	Requisitos . . . . .	4
2.2.1	AUT-4 Upload de dados via arquivo csv . . . . .	5
2.2.2	AUT-5 Utilização de dataset exemplo . . . . .	5
2.2.3	AUT-6 Configuração de parâmetros do modelo . . . . .	5
2.2.4	AUT-7 Apresentação dos resultados do modelo . . . . .	5
2.2.5	AUT-8 Apresentação de gráfico de contorno do modelo . . . . .	6
2.2.6	AUT-9 Download dos resultados . . . . .	6
2.2.7	Resumo das User Stories . . . . .	7
2.3	Mockups . . . . .	7
<b>3</b>	<b>Arquitetura</b>	<b>11</b>
3.1	Tecnologias adotadas . . . . .	12
<b>4</b>	<b>Planejamento</b>	<b>13</b>
4.1	Estimativa de esforço . . . . .	13
4.2	Sprints . . . . .	13
<b>5</b>	<b>Qualidade</b>	<b>14</b>
5.1	Inspeção . . . . .	14
5.2	Testes . . . . .	16
5.2.1	Testes unitários . . . . .	16
<b>6</b>	<b>Conclusão</b>	<b>16</b>

## 1 Contexto

Atualmente notamos uma grande adoção de sistemas que utilizam aprendizado de máquina (Machine Learning - ML) como parte de seu funcionamento. ML é a prática de fazer com que os computadores funcionem sem serem explicitamente programados. A maioria das abordagens de ML gera regras com base em um conjunto de exemplos (dados de treinamento) com o objetivo de fazer previsões. O aprendizado de máquina é um subcampo da ciência da computação com o potencial de transformar organizações e a sociedade [1], tanto que é considerado uma das três principais tendências que as organizações devem seguir para obter vantagem competitiva sobre seus concorrentes [2].

A adoção de técnicas de ML é feita a partir de um processo composto por muitas etapas, em que várias das atividades requeridas são realizadas a partir de intervenção humana. Nesse cenário surge o conceito de AutoML (Automated Machine Learning), como um processo para automatizar tarefas realizadas na aplicação de ML em problemas reais [3]. O alto grau de automação buscado pela adoção do AutoML também busca reduzir a barreira de entrada da utilização de técnicas de ML, tornando-as acessível para não especialistas.

## 2 Especificação

Nesse projeto, apresenta-se uma aplicação focada na otimização de hiperparâmetros para o algoritmo Random Forest Regressor. Esta aplicação foi inicialmente descrita num blog <sup>1</sup>, porém o código ali disponibilizado dificulta sua evolução por não seguir padrões de Engenharia de Software. Este projeto visa desenvolver a documentação e o código necessário para uma implementação robusta dessa aplicação de AutoML.

Nesta seção, descreveremos algumas das especificações utilizadas no desenvolvimento da aplicação, visando gerar o máximo de valor para seus usuários.

### 2.1 Público alvo

A partir da definição do público alvo do nosso sistema podemos definir requisitos mais aderentes às necessidades de nossos usuários. Para o caso do aplicativo de AutoML proposto, focado na otimização de hiperparâmetros, definimos que seu foco será em usuários iniciando o estudo de técnicas de ML. Portanto, o objetivo principal do aplicativo será reduzir a barreira de entrada do tema para estudantes, especificamente estudantes que não vem de possuem uma forte base em ciência da computação.

#### 2.1.1 Persona

Para facilitar a empatia com nosso público alvo, definimos uma persona que servirá de base para o levantamento de requisitos feito a seguir. A persona descrita

---

<sup>1</sup><https://towardsdatascience.com/how-to-build-an-automl-app-in-python-e216763d10cd>

nesta seção consolida depoimentos dados por pessoas pertencentes ao público alvo em conversas informais.

Nome: Edu Estudante

Descrição: Edu se formou há dois anos em administração e, após uma experiência no mercado, decidiu iniciar uma pós graduação em ciência de dados, empolgado com as muitas oportunidades profissionais abertas nessa área. Uma das áreas de ciência de dados que mais interessam Edu é a de ML, apesar do pouco contato com aplicações reais de suas técnicas.

Frustrações: Edu se sente frustrado com a dificuldade de aplicar técnicas de ML em problemas reais. O processo de ajustar os modelos parece muito custoso e difícil de compreender. As ferramentas comuns parecem ser feitas apenas para especialistas na área.

Objetivos: Edu gostaria de conseguir aplicar técnicas de ML a um problema sem recorrer a ferramentas de difícil interação ou que requerem um conhecimento muito avançado na área.

Motivação: Edu consegue se motivar ao ver que está conseguindo obter progresso na direção de seu objetivo. A apresentação de resultados que apelam para o visual, como gráficos, também facilitam sua compreensão.

## 2.2 Requisitos

Para a definição dos requisitos será o utilizado o formato de Estórias de Usuário (*User Stories*), agrupadas por Épicos.

Três Épicos foram definidos:

- **AUT-1 Inserção de dados** - agrupa as Estórias ligadas à inclusão de datasets no sistema
- **AUT-2 Configuração de parâmetros** - agrupa as Estórias que tem a ver com a parametrização da utilização do modelo
- **AUT-3 Apresentação de resultados** - agrupa as Estórias que tem a ver com a apresentação dos resultados obtidos pelo modelo, a partir dos parâmetros utilizados

Para cada User Story foram definidos: o Épico ao qual a User Story pertence, uma descrição (seguindo o template: como *usuário*, gostaria de *ação* para *benefício*), o valor de negócio estimado e os critérios de aceitação mínimos que devem ser atendidos após a conclusão das tarefas da User Story.

### 2.2.1 AUT-4 Upload de dados via arquivo csv

**Épico:** Inserção de dados

**Descrição:** Como *estudante de ML*, gostaria de *realizar o upload do meu dataset para entender o funcionamento do modelo para o meu dataset próprio*.

**Valor de negócio:** 3

**Critérios de aceitação:**

1. O usuário deve conseguir realizar o upload de arquivos de até 200MB

### 2.2.2 AUT-5 Utilização de dataset exemplo

**Épico:** Inserção de dados

**Descrição:** Como *estudante de ML*, gostaria de *ver o funcionamento do sistema com um dataset de exemplo para entender suas possibilidades mesmo sem possuir um dataset próprio*.

**Valor de negócio:** 1

**Critérios de aceitação:**

1. O usuário deve ter a opção de utilizar um dataset de exemplo caso não queira utilizar um dataset próprio

### 2.2.3 AUT-6 Configuração de parâmetros do modelo

**Épico:** Configuração de parâmetros

**Descrição:** Como *estudante de ML*, gostaria de *facilmente alterar os parâmetros utilizados pelo modelo para entender como essas alterações afetam os resultados*.

**Valor de negócio:** 2

**Critérios de aceitação:**

1. O usuário deve conseguir alterar os parâmetros do modelo a partir da interface
2. O sistema deve recarregar os resultados apresentados sempre que um parâmetro é alterado

### 2.2.4 AUT-7 Apresentação dos resultados do modelo

**Épico:** Apresentação de resultados

**Descrição:** Como *estudante de ML*, gostaria de *ver os resultados obtidos modelo para verificar se a modelagem utilizada está adequada*.

**Valor de negócio:** 3

**Critérios de aceitação:**

1. O sistema deve apresentar algumas linhas do dataset, incluindo seu cabeçalho
2. O sistema deve apresentar o nome do cabeçalho da variável Y que será calculada pelo modelo
3. O sistema deve apresentar o valor coeficiente  $R^2$  calculado pelo modelo
4. O sistema deve apresentar o valor do erro - MSE ou MAE, dependendo do parâmetro selecionado pelo usuário - calculado pelo modelo
5. O sistema deve apresentar a variação do valor do coeficiente  $R^2$  com o número máximo de features (max\_features) e com o número de estimadores (n\_estimators)

#### 2.2.5 AUT-8 Apresentação de gráfico de contorno do modelo

**Épico:** Apresentação de resultados

**Descrição:** Como *estudante de ML*, gostaria de *ver o gráfico de contorno do modelo para analisar de forma visual o comportamento do modelo*.

**Valor de negócio:** 2

**Critérios de aceitação:**

1. O usuário deve ser apresentado a um gráfico interativo representando a variação do coeficiente  $R^2$  com o número máximo de features (max\_features) e o número de estimadores (n\_estimators)

#### 2.2.6 AUT-9 Download dos resultados

**Épico:** Apresentação de resultados

**Descrição:** Como *estudante de ML*, gostaria de *baixar os resultados do modelo para poder trabalhar com eles em outras ferramentas*.

**Valor de negócio:** 1

**Critérios de aceitação:**

1. O usuário deve ter a opção de realizar o download de um arquivo csv contendo os resultados obtidos pelo modelo

### 2.2.7 Resumo das User Stories

O resumo dos Épicos e suas respectivas User Stories pode ser visualizado na Figura 1.

Epic		
▼	✚ AUT-1 Inserção de dados	
	📌 AUT-4 Upload de dados via arquivo csv	TO DO
	📌 AUT-5 Utilização de dataset exemplo	TO DO
▼	✚ AUT-2 Configuração de parâmetros	
	📌 AUT-6 Configuração de parâmetros do modelo	TO DO
▼	✚ AUT-3 Apresentação de resultados	
	📌 AUT-7 Apresentação dos resultados do modelo	TO DO
	📌 AUT-8 Apresentação de gráfico de contorno do modelo	TO DO
	📌 AUT-9 Download dos resultados	TO DO

Figura 1: Épicos e User Stories

## 2.3 Mockups

Para apoiar o desenvolvimento, alguns protótipos de baixa fidelidade foram desenvolvidos utilizando a ferramenta Balsamiq <sup>2</sup>.

Na figura 2 podemos visualizar o estado inicial da aplicação, sem que o usuário tenha feito nenhuma interação com ela. Na parte da esquerda da tela, observa-se o componente que possibilita realizar o upload de arquivos com datasets (AUT-4) e alguns parâmetros que podem ser configurados pelo usuário (AUT-6). Estes parâmetros estendem-se para a parte inferior da tela e podem ser acessados a partir da interação com o scroll vertical presente nessa área da tela. No componente principal da tela, temos algumas informações sobre a aplicação e o botão que possibilita a utilização de um dataset de exemplo (AUT-5), para possibilitar o teste da aplicação mesmo que o usuário não possua um dataset próprio.

---

<sup>2</sup><https://balsamiq.com>

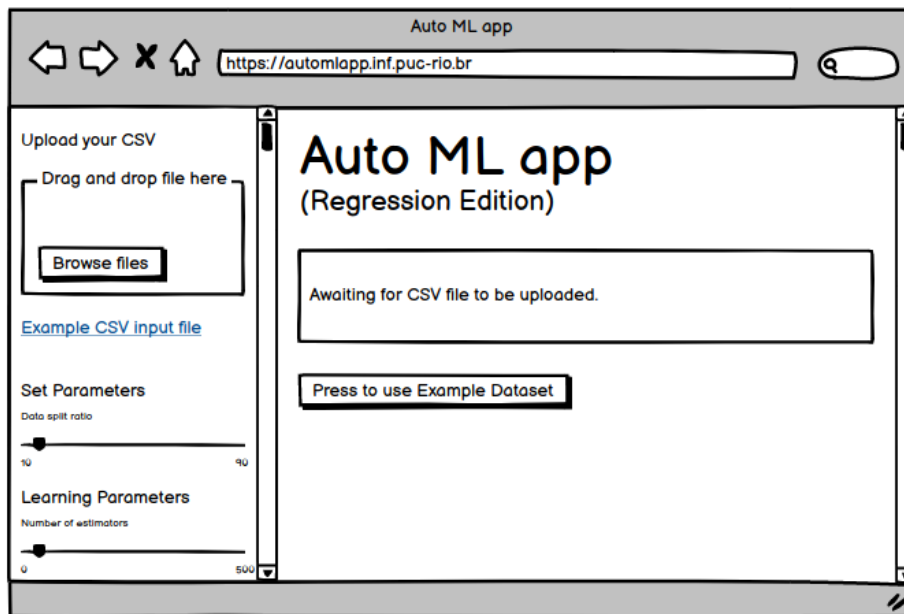


Figura 2: Tela 1

Na figura 3 apresenta-se o comportamento da aplicação após o comando de utilização de um dataset, seja via upload de arquivos ou via clique no botão que aciona a utilização do dataset de exemplo. Nesse momento, a aplicação apresenta dados básicos sobre o dataset, como a coluna Y que será estimada pelo modelo e as primeiras linhas do dataset (AUT-7).



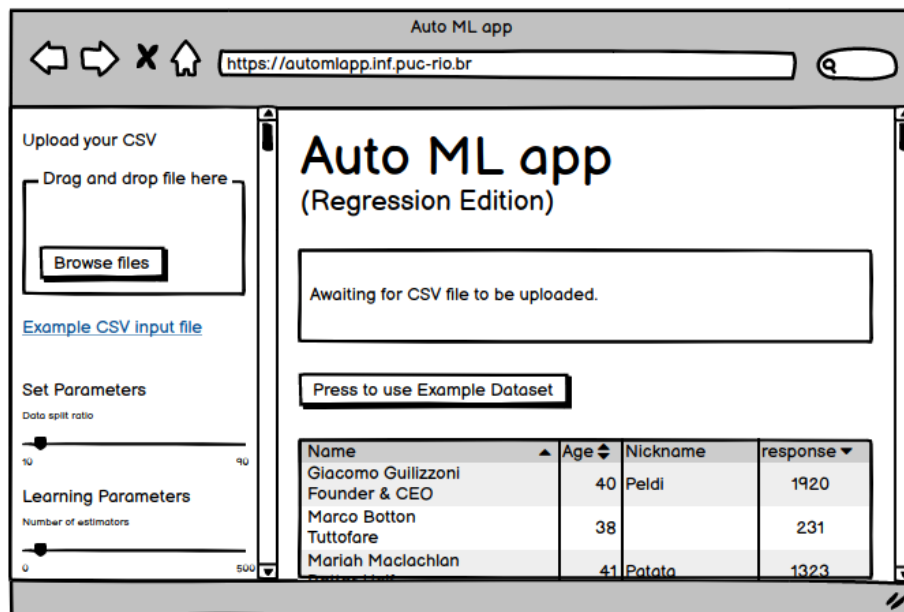


Figura 3: Tela 2

Na tela 3, apresentada na figura 4, são mostrados alguns dos resultados obtidos pelo modelo (AUT-4). Nota-se que o scroll da parte principal da tela já sofreu interação por parte do usuário, sendo esses componentes apresentados uma continuação da parte principal da tela da figura 3.

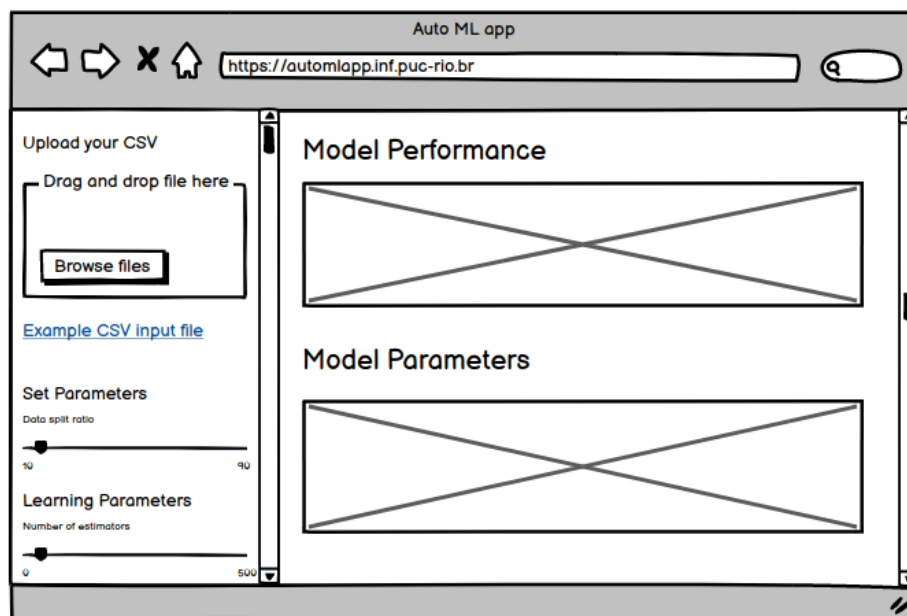


Figura 4: Tela 3

A figura 5 apresenta o fim da tela principal, sendo a continuação da tela 3 após mais interação com o scroll principal. É possível visualizar nessa tela o gráfico interativo (AUT-8) e o link para realizar o download dos resultados do modelo (AUT-9).

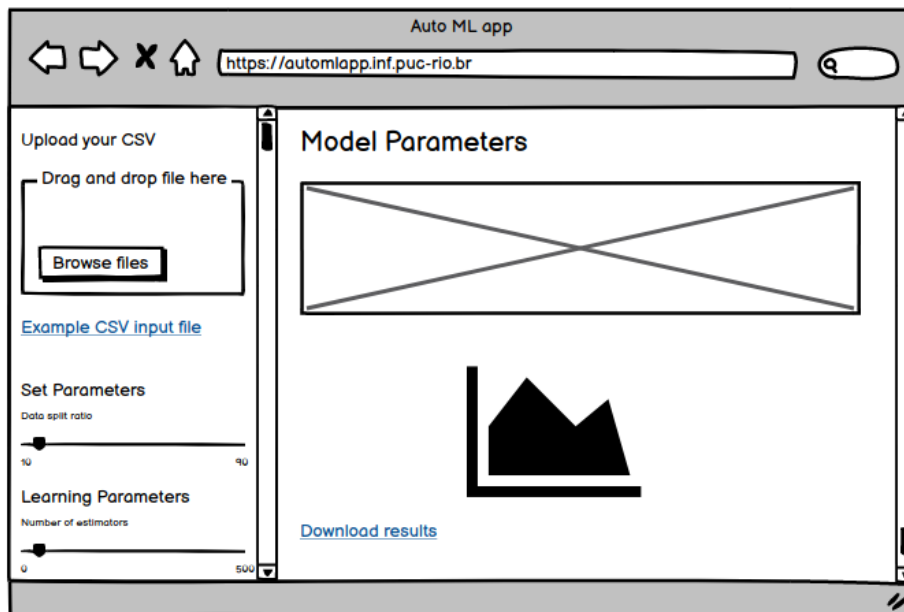


Figura 5: Tela 4

Alguns pontos menos críticos da interface, como a disposição de todos os componentes utilizados na configuração dos parâmetros do modelo, foram omitidos da prototipação. Para estes componentes será seguido o mesmo padrão utilizado nos que foram apresentados nas telas desta seção.

### 3 Arquitetura

A aplicação apresenta uma arquitetura bem simples, baseada no pacote Python Streamlit <sup>3</sup> focado em transformar projetos de ciência de dados em aplicativos web. Este pacote permite a criação de um aplicativo web a partir de um script Python, sem a necessidade de qualquer tipo de camada adicional.

Uma visão da arquitetura é apresentada na figura 6. A partir do browser de um desktop, considerando a persona descrita, o usuário acessa via internet o servidor contendo o projeto Streamlit - Python e interage com as páginas web servidas por ele.

---

<sup>3</sup><https://streamlit.io/>

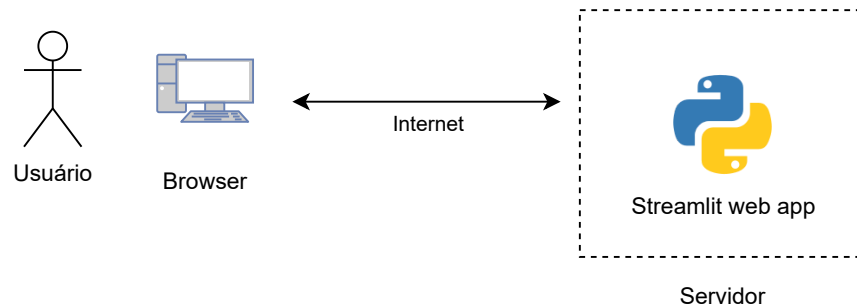


Figura 6: Arquitetura da solução

Os requisitos atuais do produto não requerem que nenhum dado seja armazenado de forma permanente pela aplicação, excluindo a necessidade de uma camada de persistência. A utilização do pacote Streamlit também remove a necessidade de algumas camadas comuns a esse tipo de aplicação, como um servidor de páginas web estáticas. Essas particularidades nos levam a conseguir cumprir os requisitos do produto com uma arquitetura extremamente simples.

### 3.1 Tecnologias adotadas

Toda a aplicação foi construída utilizando a linguagem de programação Python (versão 3.7.9), muito utilizada no desenvolvimento de aplicações de ciência de dados. Na tabela 1, são apresentados os pacotes Python utilizados na construção da aplicação.

Pacote	Versão	Descrição
streamlit	0.71.0	Possibilita a criação web apps em Python, a partir da inserção de markups html [VER] nos scripts - <a href="https://streamlit.io/">https://streamlit.io/</a>
pandas	1.1.3	Oferece diversos métodos para a manipulação de datasets - <a href="https://pandas.pydata.org/">https://pandas.pydata.org/</a>
numpy	1.19.2	É um pacote para realizar computação científica em Python - <a href="https://numpy.org/">https://numpy.org/</a>
plotly	4.14.1	Possibilita a criação de gráficos interativos - <a href="https://plotly.com/python/">https://plotly.com/python/</a>
scikit-learn	0.23.2	Possibilita a realização de análises preditivas em um dataset - <a href="https://scikit-learn.org/stable/">https://scikit-learn.org/stable/</a>

Tabela 1: Pacotes Python utilizados

Para o versionamento do código foi utilizada a tecnologia Git <sup>4</sup> e código está armazenado em um repositório <sup>5</sup> público do Github.

<sup>4</sup><https://git-scm.com/>

<sup>5</sup><https://github.com/silvioalonso/msc-final-project-auto-ml>

## 4 Planejamento

O planejamento do desenvolvimento da aplicação foi feito seguindo em boa medida a utilização do SCRUM como método de desenvolvimento. Apesar de todo o trabalho ter sido realizado por apenas um desenvolvedor, diversos outros artefatos da metodologia, como sprints e kanban, foram utilizados. Todo o processo de desenvolvimento foi apoiado pela ferramenta Jira <sup>6</sup>, amplamente utilizada no mercado para esse propósito.

### 4.1 Estimativa de esforço

Considerando algumas particularidades que dificultam a utilização de Story Points, como poucos sprints para o ajuste da velocidade, e não ter muitas vantagens na sua utilização em um projeto com apenas um desenvolvedor, optou-se por realizar a estimativa das User Stories em dias de desenvolvimento. A tabela 2 apresenta um resumo da estimativa de esforço feita para as User Stories.

User Story	Estimativa de esforço
AUT-4 Upload de dados via arquivo csv	2 dias
AUT-5 Utilização de dataset exemplo	1 dia
AUT-6 Configuração de parâmetros do modelo	2 dias
AUT-7 Apresentação dos resultados do modelo	2 dias
AUT-8 Apresentação de gráfico de contorno do modelo	1 dia
AUT-9 Download dos resultados	1 dia

Tabela 2: Estimativa de esforço

### 4.2 Sprints

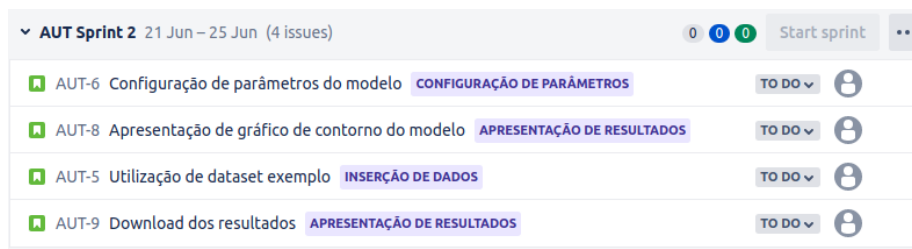
Foram definidos dois sprints de cinco dias cada para a execução do desenvolvimento da aplicação. A prioridade das User Stories para execução foi definida levando-se em consideração o valor de negócio gerado pela implementação da User Story e a estimativa do esforço necessário para sua conclusão. As figuras 7 e 8 apresentam as User Stories definidas para serem desenvolvidas nos sprints 1 e 2, respectivamente.



Figura 7: Sprint 1

<sup>6</sup><https://www.atlassian.com/software/jira>

No primeiro sprint, temos uma estimativa de esforço total de quatro dias e seis pontos de valor de negócio. A folga nesse sprint serve para realizar tarefas ligadas infraestrutura, como criação do repositório e configuração do ambiente de desenvolvimento.



▼ AUT Sprint 2 21 Jun – 25 Jun (4 issues)		0 0 0	Start sprint	...
AUT-6	Configuração de parâmetros do modelo	CONFIGURAÇÃO DE PARÂMETROS	TO DO	
AUT-8	Apresentação de gráfico de contorno do modelo	APRESENTAÇÃO DE RESULTADOS	TO DO	
AUT-5	Utilização de dataset exemplo	INSERÇÃO DE DADOS	TO DO	
AUT-9	Download dos resultados	APRESENTAÇÃO DE RESULTADOS	TO DO	

Figura 8: Sprint 2

Já o segundo sprint totalizou cinco dias de esforço estimado para desenvolvimento e também seis pontos de valor de negócio.

## 5 Qualidade

Para garantir a qualidade da aplicação, foram realizados processos de inspeção e de testes unitários, descritos nesta seção.

### 5.1 Inspeção

Um primeiro processo para a verificação da conformidade na aplicação desenvolvida foi a realização de uma inspeção. Seguindo o princípio do SCRUM, que sugere pela realização de todo o ciclo de desenvolvimento de software durante o sprint, cada User Story executada teve sua inspeção realizada no mesmo sprint, buscando não haver um acúmulo de atividades dessa natureza.

Para cada User Story executada, foi realizada uma inspeção manual de seus Critérios de Aceitação. O resumo dessa atividade é apresentado a seguir:

AUT-4 O usuário deve conseguir realizar o upload de arquivos de até 200 MB  
Status: Inconclusivo

Observação: Apesar de realizar o upload de arquivos de diversos tamanhos, não foi possível aferir se arquivos de até 200 MB podem ser utilizados, já que o tamanho do arquivo aceito depende das configurações do servidor, o que não está contemplado no escopo desse projeto.

AUT-5 O usuário deve ter a opção de utilizar um dataset de exemplo caso não queira utilizar um dataset próprio

Status: Atendido

AUT-6 O usuário deve conseguir alterar os parâmetros do modelo a partir da interface

Status: Atendido

AUT-6 O sistema deve recarregar os resultados apresentados sempre que um parâmetro é alterado

Status: Não atendido

Observação: Após a alteração de um parâmetro pelo usuário, o sistema faz uma atualização total da interface, voltando para o estado inicial em que não possui nenhum dataset carregado. Para observar como a alteração do parâmetro afetou os resultados, o usuário precisa refazer o upload do dataset.

AUT-7 O sistema deve apresentar algumas linhas do dataset, incluindo seu cabeçalho

Status: Atendido

AUT-7 O sistema deve apresentar o nome do cabeçalho da variável Y que será calculada pelo modelo

Status: Atendido

AUT-7 O sistema deve apresentar o valor do coeficiente  $R^2$  calculado pelo modelo

Status: Atendido

AUT-7 O sistema deve apresentar o valor do erro - MSE ou MAE, dependendo do parâmetro selecionado pelo usuário - calculado pelo modelo

Status: Atendido

AUT-7 O sistema deve apresentar a variação do valor do coeficiente  $R^2$  com o número máximo de features (max\_features) e com o número de estimadores (n\_estimators)

Status: Atendido

AUT-8 O usuário deve ser apresentado a um gráfico interativo representando a variação do coeficiente  $R^2$  com o número máximo de features (max\_features) e o número de estimadores (n\_estimators)

Status: Atendido

AUT-9 O usuário deve ter a opção de realizar o download de um arquivo csv contendo os resultados obtidos pelo modelo

Status: Atendido

## 5.2 Testes

Considerando a baixa complexidade da aplicação e seu desenvolvimento se dar em apenas uma camada, apenas testes unitários foram desenvolvidos para garantir sua qualidade.

### 5.2.1 Testes unitários

Testes unitários foram desenvolvidos para cobrir o funcionamento da função do modelo de ML utilizado. Utilizando a biblioteca Pytest <sup>7</sup> diversos cenários da implementação do modelo Random Forest Regressor podem ser testados automaticamente sempre que alterações na função são realizadas.

## 6 Conclusão

Após a conclusão do desenvolvimento do software, podemos fazer algumas avaliações sobre as decisões tomadas durante o projeto. Apesar de não termos conseguindo fazer validações com usuários reais, o sistema cumpriu em boa medida os requisitos definidos em sua especificação. Além disso, a metodologia Scrum, mesmo que apenas parcialmente aplicada, garantiu o ritmo necessário para que o desenvolvimento cumprisse a necessidade de prazo imposta. Por fim, talvez a maior dúvida técnica fosse a utilização da biblioteca Streamlit na criação de aplicativos web. Esta se mostrou muito adequada para esse tipo de projeto, possibilitando um desenvolvimento rápido, sem a necessidade de configurações extensas ou a criação de múltiplas camadas de infraestrutura.

---

<sup>7</sup><https://docs.pytest.org/>



## Referências

- [1] Nicola Jones. Computer science: The learning machines. *Nature News*, 505(7482):146, 2014.
- [2] Gartner’s Gartner. Hype cycle for emerging technologies identifies three key trends that organizations must track to gain competitive advantage, gartner’s 2016 hype cycles highlight digit. *Bus. Ecosyst.(2016)*, 1:90018–3, 2016.
- [3] Xin He, Kaiyong Zhao, and Xiaowen Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.