**Projeto Aplicado – Relatório Final**

|  |  |
| --- | --- |
| **Nome do Aluno** | **Jonas Jahns da Rocha** |
| **Título do Trabalho** | Portal de Informações de licitações |
| **Curso** | MBA em Big Data |
| **Linha de Especialização** | Big Data Analytics – Análise de Dados da Web |
| **Orientador** | **Wallison Willian Guimarães** |
| **Data** | **Setembro/2019** |

**INTRODUÇÃO**

1. Apresentação do desafio e da solução:
   1. Setor do mercado e a justificativa de tal seleção;
   2. Características e restrições de escopo do desafio;
   3. Oportunidade vislumbrada que motivou o desenvolvimento da solução.

a. O setor do mercado escolhido para esse trabalho é o setor público. Vemos um crescente interesse e envolvimento das pessoas nos gastos públicos, porém, o acesso às informações é dificultado, por não serem claras, centralizadas e nem amigáveis. Dentro desse grande universo de dados de informações públicas, um setor chama muita atenção, que é o de licitações públicas. Hoje é alvo de muitas críticas da população e de dor dos gestores de empresas públicas, eles dividem a mesma opinião: “Não há clareza e nem amplo conhecimento da divulgação e do processo de licitação completo”. Sendo assim, o trabalho apresentado abaixo busca diminuir essa dor vivida pelos envolvidos no assunto.

b. Pensando em uma solução simples para o problema, o trabalho busca sanar essa dor, aumentando a divulgação das licitações realizadas pelos órgãos governamentais e empresas de capital misto, que também são submetidas a lei 8.666, que trata de licitações e pregões públicos. Esse trabalho não tem o foco de abrir os dados detalhados de cada licitação, também não faz parte do escopo o acompanhamento do tramite completo do processo licitatório, a ideia é trazer, em um portal centralizado, informações de licitações como título, data de lançamento no sistema, descrição presente na publicação, classificadas com tags informativas que possam facilitar a filtragem e a navegação do usuário, dentre elas canal da licitação, se eletrônico ou presencial, cidade e estado, produto ou serviço.

c. Trazendo um exemplo, durante minha atuação em uma empresa de capital misto, estava vasculhando algumas licitações para servir de modelo a uma licitação que a minha área estava criando. Durante a minha busca por licitações encontrei uma licitação para compra de veículos, onde houveram apenas 2 participantes na licitação e o vencedor conseguiu vender o veículo com valor acima do valor padrão das concessionárias, demonstrando a pouca participação da população nesse tipo de evento, a pouca fiscalização da população e a baixa divulgação das licitações. Além disso, a mesma empresa que arrematou a licitação anterior, participava ativamente de quase todas as licitações disponibilizadas pela instituição, ganhando boa parte dos pregões, reforçando, assim, a falta de comunicação. Concluindo, as licitações são dominadas por players consagrados do mercado, não têm divulgação clara e aberta e são de difícil busca por parte dos usuários.

1. Identificação da(s) pessoa(s) envolvida(s) no desafio:

Para a identificação das pessoas envolvidas no desafio, utilizou-se o mapa de empatia, conforme descrito abaixo:

**Pessoa 1: Empresário dono de empresa de médio porte, com mais de 10 anos de mercado e que quer aumentar o faturamento de sua empresa:**

**Com quem estou sendo empático:**

- Empresário com cerca de 40 anos, dono/gestor de uma empresa de médio porte, com estabilidade no mercado pois já tem mais de dez anos de atuação no seu setor.

- Fico sabendo, através de um amigo, que o governo federal abre diversas licitações no seu estado e que sua empresa estaria qualificada para participar de várias dessas licitações.

**O que precisa fazer:**

- Aumentar o lucro de sua empresa e fazer novos negócios, porém, com o mercado cada dia mais competitivo, tem dúvidas de qual/quais ações deve tomar para aumentar o faturamento da empresa.

- Participar de licitações dentro da área de atuação da sua empresa é um de seus desejos.

**O que ele vê:**

- Um país em crise, porém, um governo com diversas obrigações com a população e com o orçamento mandatório, ou seja, o governo tem dinheiro reservado e pronto para ser investido e é obrigado a investir.

- Várias pessoas falando de licitações e pregões do governo.

- Sua empresa pronta para crescer e investir

**O que ele fala:**

- Há poucas oportunidades para crescer nos setores já consagrados do meu mercado de atuação

- Tenho capital e pessoas para expandir nossos negócios

- Minha empresa é estável e confiável

- Já li muitas notícias sobre licitações, mas não sei nem onde encontra-las

**O que ele faz:**

- Gerencia a empresa de maneira tradicional, como seus antecessores.

- Busca se manter informado através de sites de notícias e telejornais.

- É ambicioso, então, busca sempre novas oportunidades de negócio para a empresa, mesmo que de maneira conservadora.

**O que ele escuta:**

- O país está em crise, não há dinheiro para investir.

- O acesso a crédito é muito baixo.

- Não há oportunidades para baixar mais o preço de seus produtos.

**O que ele pensa e sente:**

**Dores:**

- Ele tem medo de investir em setores muito revolucionários.

- Ele não quer arriscar a credibilidade e estabilidade de sua empresa com grandes aportes ou mudanças.

- Ele sempre ouve falar de licitações, mas não sabe onde encontra-las e nem como participar.

**Desejos:**

- Ele quer aumentar o faturamento e expandir os negócios da empresa.

- Ele quer atuar de maneira segura e assertiva na expansão da empresa.

- Ele quer crescer sem grandes inovações e sem ter que baixar ainda mais o lucro da empresa.

**Pessoa 2:** **Gestor de empresa pública, com carreira consolidada no setor público e que busca gerir os gastos de maneira consciente:**

**Com quem estou sendo empático:**

- Gestor de empresa pública com cerca de 20 anos de carreira, gestor de uma empresa púbica que divulga regularmente licitações nos portais do governo.

- Por causa da baixa participação de empresas nas suas licitações, ele está buscando maneiras de melhorar a divulgação de suas licitações.

**O que precisa fazer:**

- Diminuir os custos de sua empresa, para cumprir o orçamento que está cada vez mais enxuto.

- Gerir de maneira eficiente e inovadora.

- Cumprir a legislação aplicada a todas as suas ações.

**O que ele vê:**

- O governo federal diminuindo as verbas destinadas a todos os setores do país.

- Baixa competitividade em suas licitações.

- Um mercado com diversas empresas buscando novos meios para faturar.

**O que ele fala:**

- A legislação está ajuda na competitividade e regulação do setor público, porém burocratiza o processo.

- Se houvesse mais competição em suas licitações, os preços seriam mais justos.

**O que ele faz:**

- Gere a empresa, seguindo a legislação.

- Busca divulgar a licitação às empresas com quem ele já tem relacionamento.

**O que ele escuta:**

- Empresas estão falindo por não conseguirem vender seus produtos.

- As empresas não conhecem e nem ouvem falar de licitações.

- Há muita reclamação da população com relação as licitações, devido a diversos escândalos que foram noticiados.

**O que ele pensa e sente:**

**Dores:**

- Baixa participação de empresas nas suas licitações.

- Baixa diversidade de empresas nas suas licitações.

- Falta de divulgação dos pregões.

**Desejos:**

- Que várias empresas sólidas participem de suas licitações.

- Divulgar e promover as licitações e seus resultados.

- Diminuir os custos da empresa que está gerindo.

1. Construção da proposta de solução:
   1. Requisitos da construção do protótipo/MVP, com descrição do experimento e as métricas de validação;
   2. Indicadores econômico-financeiros do projeto.

a. A proposta atual de MVP para o trabalho foi alterada em relação ao entregue anteriormente no documento Proposta de Solução. Para isso foram propostos os seguintes itens como requisitos mínimos para entrega do MVP do trabalho:

- Busca de informações de licitações do Diário Oficial da União.

- Análise das licitações e classificação de acordo com algumas tags.

- Apresentação do resultado através de um dashboard de informações

Os experimentos foram executados durante o desenvolvimento da solução, inicialmente utilizou-se o modelo de download dos diários oficiais do governo no formato PDF, para posterior processamento dessas informações. Inicialmente a ideia era buscar o PDF de diários de vários órgãos e instâncias da composição do funcionalismo público (Diário da União, Diário dos Estados e dos Municípios), focando, principalmente, nos de maior relevância como os estados de São Paulo, Rio de Janeiro, Minas Gerais e Rio Grande do Sul, além de cidades grandes como São Paulo, Rio de Janeiro, Belo Horizonte, Fortaleza, Brasília, Porto Alegre, entre outros. Devido a diversas dificuldades encontradas na conversão do PDF para um formato de mais fácil validação, como texto puro ou estruturado, html ou json, a hipótese de utilização do PDF foi descartada.

Sendo assim, buscaram-se alterativas para o processamento dessas informações. Verificou-se que o Diário Oficial da União é disponibilizado no formato HTML, no próprio site do Governo, sendo assim, criou-se um web crawler que lê as primeiras licitações publicadas no dia informado. Foram escolhidas 10 licitações pois elas são de mais fácil acesso, uma vez que não precisamos executar nenhum JavaScript para a obtenção desses dados, as demais licitações publicadas no dia ficam escondidas dentro de uma função JavaScript, sendo ativadas pelo clique do usuário, dificultando a obtenção desses dados com as tecnologias escolhidas.

Círculo Dourado:

O segundo momento da validação foi a classificação manual de licitações em algumas tags como: Tipo do Pregão (presencial ou eletrônico), Cidade e Estado, Tipo do Objeto (Produto, Serviço ou Ambos), Empresa que está realizando a licitação. A partir da classificação manual de vários elementos coletados, utilizou-se um mecanismo de machine learning para a leitura desses dados e criação de um modelo de classificação para novas licitações. Esse modelo foi aplicado para os demais itens da base de dados coletados e seus resultados serão apresentados no tópico específico.

Por fim, foi criado um dashboard, em que se pode verificar a quantidade de licitações processadas e classificadas, realizando o filtro pelas tags criadas no modelo de classificação, além da exibição dos itens unitários que compõem o número filtrado. Para a validação do modelo, foi executado o processo de seleção aleatória de itens dentro do banco de dados, fazendo-se a avaliação manual das tags atribuídas a licitação selecionada, validando assim a assertividade do modelo de classificação escolhido.

b. Para os custos do projeto, serão utilizados os valores padrões de hospedagem e banco de dados da plataforma Amazon AWS. Pensado em quantidade de chamadas simples e tempo de computação diário do MVP da solução, a ideia é que o custo mensal da solução seja de certa de U$ 10,00 por mês. Para os indicadores de sucesso da solução, busca-se medir a quantidade de acessos a solução hospedada no AWS, tendo uma ideia inicial de 100 acessos/mês na plataforma, com um crescimento mensal de 5%.

Após a consolidação de uso da plataforma, será pensando em maneiras de monetizar a solução como anúncios, limitação de quantidade de acessos a ferramenta, venda de pacotes de acesso ilimitados, venda de priorização de exibição das licitações para os órgãos governamentais.

**DESCRIÇÃO DO DESENVOLVIMENTO DO TRABALHO**

1. Criação do coletor de dados da web:
   1. Definição das fontes e tipos de dados a serem coletados;
   2. Criação de um coletor de dados;
   3. Coleta e validação dos dados.

A fonte de dados para a coleta de informações é o site <http://www.in.gov.br/web/guest/inicio>, onde o governo divulga todas as publicações diárias. Os dados são acessados através de requisições HTTP, sendo retornados no formato HTML. Para cada dia consultado, é retornado um HTML contendo 10 links para as licitações, que são acessados um a um para a coleta da informação das licitações. Dentro das licitações são coletados os dados de cabeçalho, descrição alto nível da licitação e nome e cargo da pessoa responsável pela publicação. O coletor de dados foi desenvolvido em Python, com a utilização do Framework Scrapy para a criação dos coletores, Splash para o processamento do JavaScript das páginas e Scrapy-Splash para a integração das duas tecnologias. Esses dados são separados, transformados em um objeto e armazenados em banco durante cada interação do algoritmo com cada uma das páginas acessadas.

1. Modelagem estatística da solução:
   1. Definição do modelo a ser implementado;
   2. Realização de testes preliminares;
   3. Validação do modelo.

A modelagem estatística foi feita através da análise de tags dos textos. Nos experimentos executados foram coletados, incialmente, dados de 1000 licitações de 100 dias diferentes de publicação do site do Diário Oficial da União. Como a classificação é realizada de maneira manual, iniciou-se com a classificação de apenas 2 tags do modelo proposto, reduzindo o volume de trabalho manual/operacional vinculado a classificação manual de cada uma das licitações.

Então, por causa da ótima performance dos modelos Naive Bayes, eles foram escolhidos como base de utilização na modelagem. Além disso, como temos uma base de informações textuais das licitações, foram utilizadas as técnicas de TF-IDF para o treinamento do modelo. Foi realizada a limpeza no texto com a remoção de caracteres não textuais, principalmente quebras de linha. A partir disso é criado um vetor de ocorrência de cada uma das palavras do texto ( TF ) e também a relevância das palavras com relação as licitações ( IDF ). Para essa implementação foi utilizada a biblioteca scikit-learn, que contempla diversos algoritmos para o processamento de texto, aprendizado de máquina, etc.

Com o levantamento acima, foi realizado o treinamento do algoritmo de Random Forest para a criação de um modelo de classificação. A escolha do Random Forest foi realizada pela sua fácil implementação e costumeira utilização em algoritmos de classificação de Machine Learning em várias áreas da Inteligência Artifical.

O Algorítmo usa a base da inteligência das multidões, um estudo consagrado que mostra que em uma grande população de pessoas, a resposta a determinado evento tender a ser correta, em sua média. Sendo assim, o algoritmo cria diversas árvores de decisão para escolher a classificação certa, selecionando várias aleatórias para a classificação, escolhendo a classificação que tiver maior média.

Além disso, o Random Forest foi escolhido também por assumir que os itens são independentes entre si, ou seja, a classificação de uma licitação não influencia na classificação de outra licitação, sendo assim, são consideradas independentes entre si.

Os dados foram divididos entre base de treino (80%) e base de teste (20%), considerando-se a acurácia do algoritmo como a quantidade de licitações que foram classificadas de maneira correta pelo algoritmo, dividido pela quantidade total de itens classificados, nesse ponto considerou-se a acurácia individual de cada uma das tags, não foi realizada a média entre a classificação das tags. Além disso, os experimentos foram realizados de maneira individualizada, sem dependência entre as tags.

Esse modelo classificador teve o seguinte comportamento para as tags abaixo:

- Presencial ou Eletrônico: 85%

- Produto, Serviço ou Ambos: 60%

Sendo assim, podemos utiliza o modelo para a classificação de futuras licitações, sem prejuízos a confiabilidade dos dados.

1. Implementação da solução utilizando algum framework:
   1. Definição do framework;
   2. Implementação e teste da solução.

Os Framewokrs escolhidos para a implementação da solução foram Scrapy e Splash, essa dupla foi escolhida pois são complementares, sendo o Scrapy responsável pela navegação das páginas, de maneira muito performática e o Splash pelo processamento do JavaScript das páginas, algo que não é realizado pelo Scrapy. Além disso, ambos são compatíveis com a linguagem Python, que é uma das líderes em Big Data, Machine Learning e Web Scrapping.

Com o projeto todo realizado em Python, foi possível a utilização do SciKit-Learn para os algoritmos de Aprendizado de Máquina, PyMongo para integração com o MongoDB além da utilização do Docker para a hospedagem local do servidor Splash.

Todas as implementações foram feitas no ambiente local, dentro do Windows 10 que já estava instalado na minha máquina. Não foi necessária a utilização de uma máquina virtual com a instalação de uma distribuição do Linux pois todos os componentes tecnológicos escolhidos para essa solução são compatíveis com a distribuição Windows, com isso, ganhou-se em performance da solução, uma vez que a máquina virtual divide recursos com a máquina hospedeira.

Os testes realizados no ambiente foram um sucesso, tanto de tempo de execução como de armazenamento e integração entre as ferramentas, cabe ressaltar alguns problemas de performance do Splash, como ele é utilizado para processar JavaScript da página, há um pouco de demora nesse processo, sendo a etapa mais lenta de toda a solução.

1. Definição e implementação da estrutura de armazenamento:
   1. Seleção da tecnologia de armazenamento NoSql;
   2. Implementação e teste da estrutura de armazenamento.

Pensando na solução de armazenamento, foi feita uma relação entre a maneira com que os dados estavam apresentados na web e as possibilidades de armazenamento NoSql, sendo assim, como os dados estavam apresentados em HTML, a estruturação deles em Json é uma saída lógica e simples para o armazenamento, sendo assim, foi escolhido o MongoDB como banco de dados NoSql, atendendo a expectativa de armazenamento das estruturas de maneira simples, pois é orientado a documentos, com queries também simples e com simples integração ao Python.

Foi instanciado um banco local no Windows, armazenado os dados na própria estrutura de pastas da distribuição. Esse banco fica visível através do IP da máquina e na porta padrão da distribuição MongoDB, portas 27017.

Para a integração ao Python, foi utilizado Pymongo para a realização da conexão com o banco de dados e a biblioteca JSON para transformação dos objetos coletados em estruturas a serem salvas no banco de dados.

Além disso, foi conectado ao banco de dados uma instância no mongo shell, onde eram realizadas as validações, a tempo de codificação, das informações inseridas, excluídas e consultadas pela aplicação, visando garantir a integridade da solução.

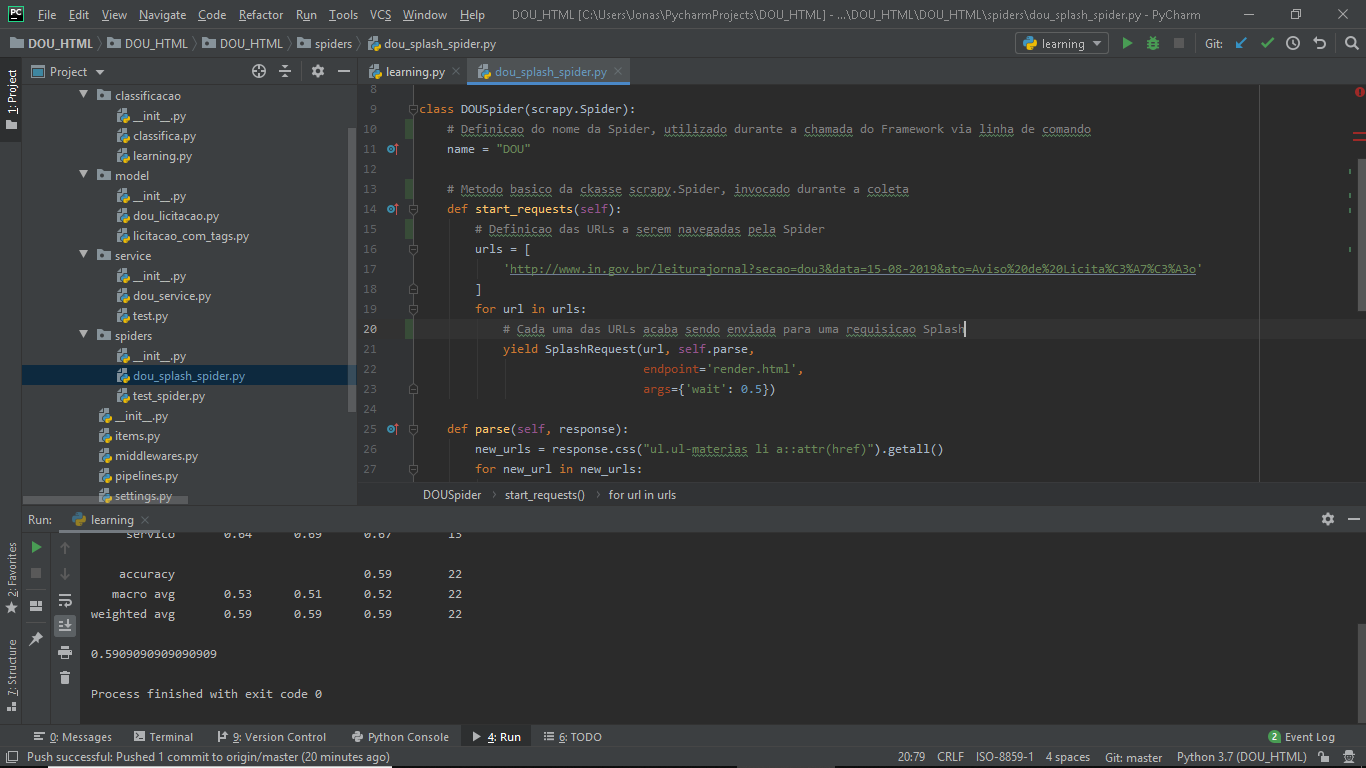
1. Criação de dashboard para a apresentação dos dados:
   1. Definção do formato de apresentação dos dados (texto, gráfico, mapas, etc);
   2. Ligação dos dados ao dashboard;
   3. Teste e validação.

**DESCRIÇÃO DO TRABALHO FINAL**

1. Fase de coleta:
   1. Apresentação das ferramentas;
   2. Apresentação das técnicas;
   3. Apresentação das integrações.

Para a resolução do problema foi escolhida a linguagem Python, tendo integração com ambientes Windows e Linux, por ter sido apresentada como a linguagem mais utilizada em algoritmos de BigData e Machine Learning durante o percorrer do curso e por sua facilitade com tratamento de arquivos e strings, tarefa bem mais árdua em linguagens clássicas como Java e C#.

Inicialmente, as requisições para o site do governo utilizam o framework Scrapy, onde é definida uma “spider” que acessa a página inicial do diário oficial da união:



Devido a estrutura dos projetos Scrapy, todas as Spiders devem herdar a classe base Spider, conforme apresentado acima. Também há a necessidade de sobrescrita de dois pontos:

- O nome da Spider: no caso acima a Spider foi nomeada como DOU

- O método: start\_request: Esse é o método base das Spiders, por ele que se inicia todo o web crawling da ferramenta.

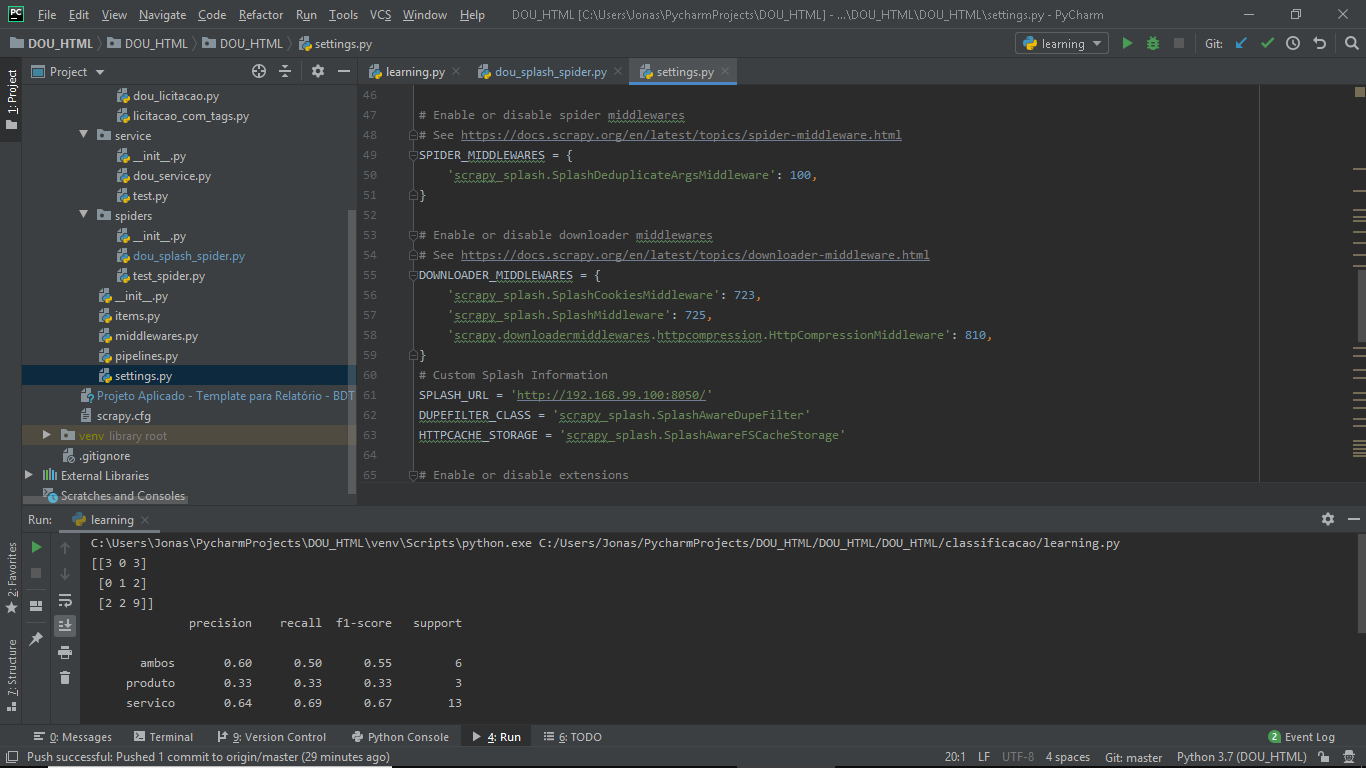
Em um segundo momento, cada uma das respostas é encaminhada para um ambiente Splash, conforme imagem, em que é processado o JavaScript da página, buscando apresentar todas as informações que não estão presentes no HTML inicial da requisição. No método que invoca o ambiente Splash, passamos a URL inicial, a função de call-back “parse”, o end-point da aplicação e também é incluído o tempo de espera entre diferentes requisições, que busca evitar time-outs dentro da aplicação.

Para a utilização conjunta dos frameworks Scrapy e Splash, são necessários:

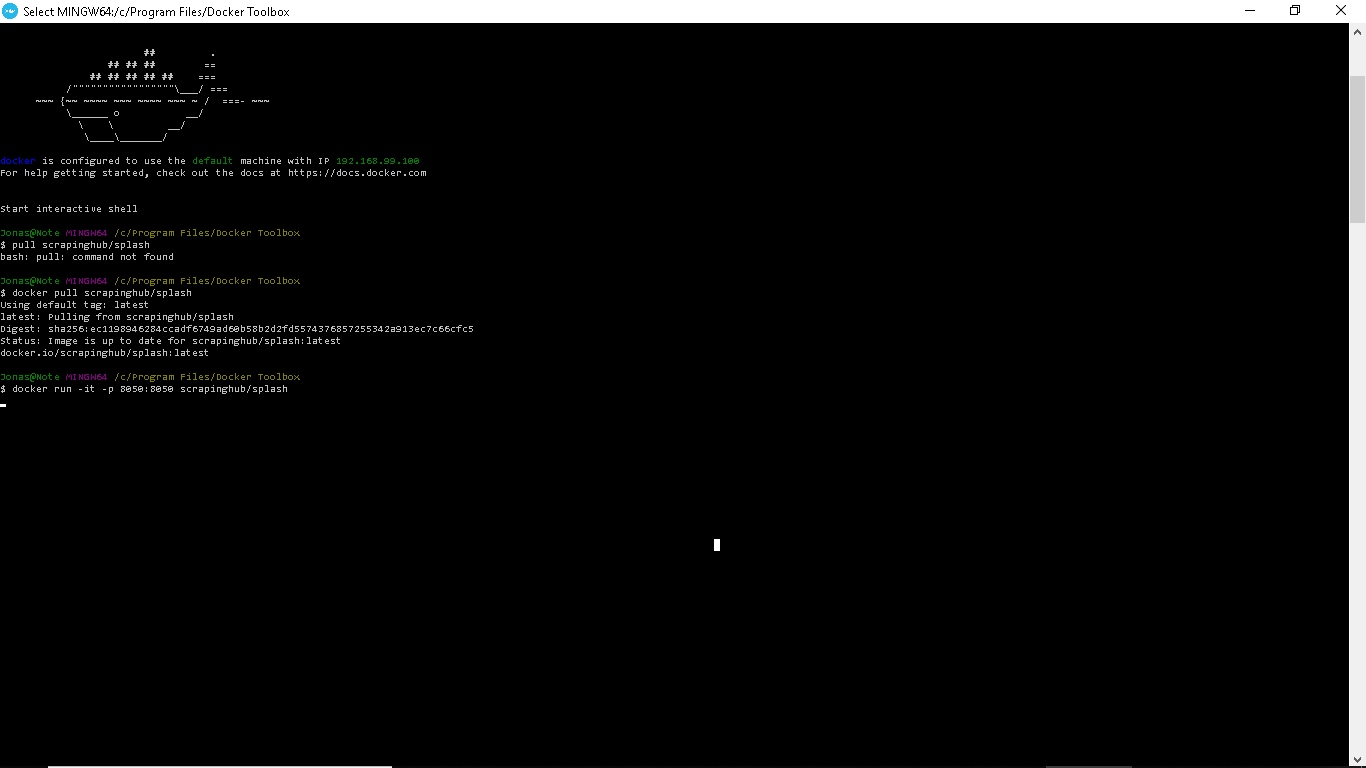
- Alteração do arquivo de settings para ignorar a busca pelas regras dos robôs, arquivo que geralmente está disponível nas páginas porém não está presente dentro do ambiente Splash. A desativação desse ponto trouxe um ganho de velocidade a Spider, pois ela ficava aguardando até time-out por essa página.

- Inclusão, no arquivo settings, do middleware Splash para requisições de dowloads.

- Inclusção, no arquivo de settings, do endereço do host Splash, definições de Cahce e DupeFilter

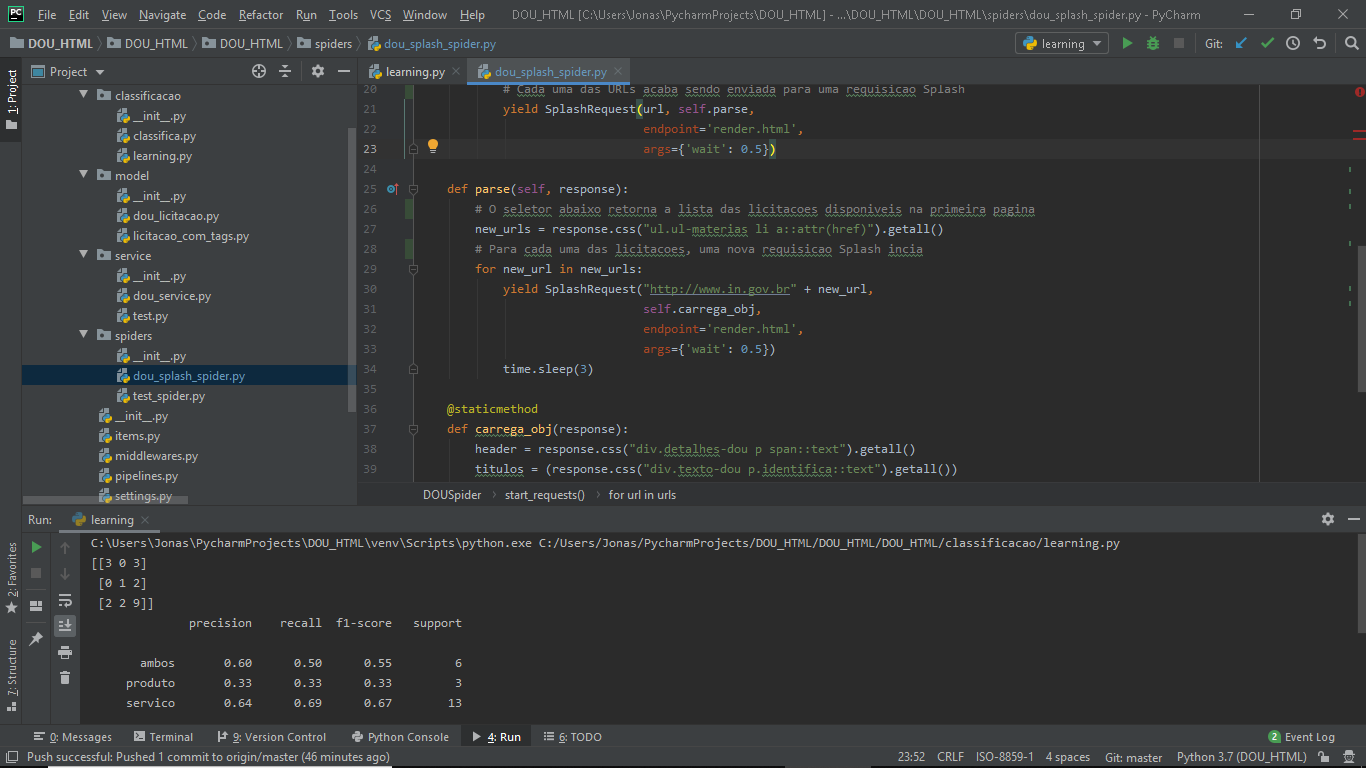


Para que o ambiente Splash estivesse em funcionamento, foi realizada a criação e deploy dele através do Docker Shell para Windows. Primeiramente, é feito a clonagem do projeto através do comando docker pull, em seguida o docker é instanciado utilizando o comando run:

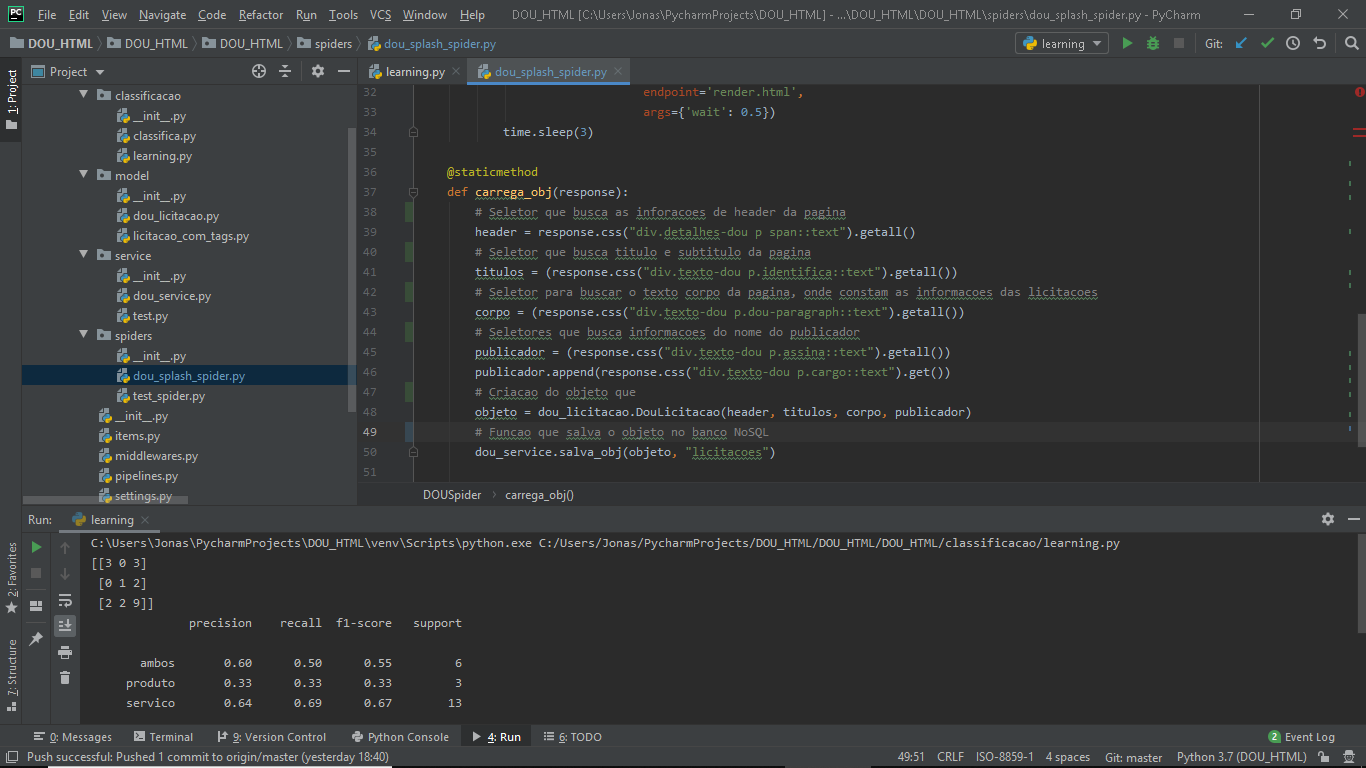


O serviço fica disponível no IP local, utilizando a porta 8050, conforme definido no comando acima.

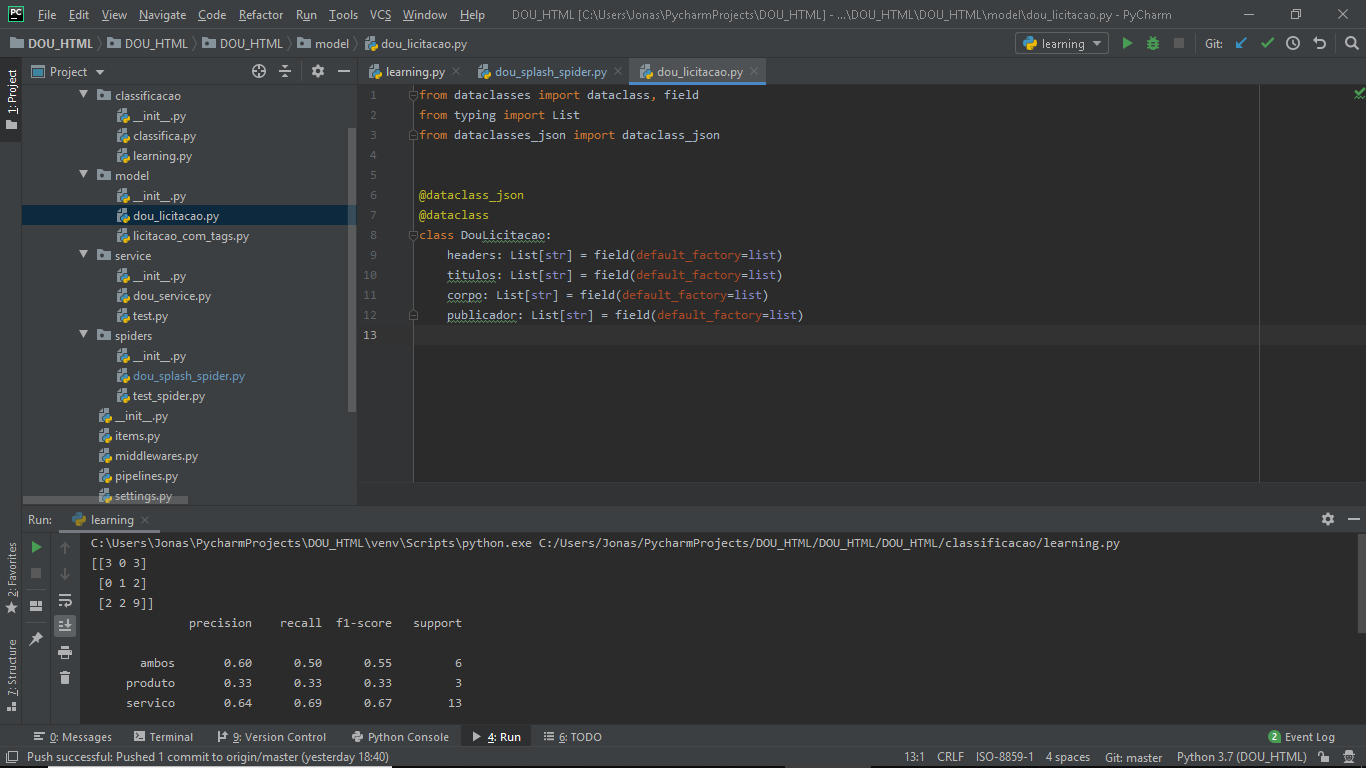
Com os ambientes configurados e integrados, todas as requisições iniciais são encaminhadas para o ambiente Splash, que processa o JavaScript da página e retorna ao processamento da Spider, na função de call-back parse, que é sobrescrita, conforme imagem abaixo:



Para cada página encontrada, uma nova chamada Splash é realizada, que buscam as informações disponíveis nas páginas específicas de cada licitação, a função carrega\_obj é apresentada abaixo:



Nessa função são buscados, através de seletores css, informações do header, título, corpo e dados do publicador da licitação. Esses dados são carregados em um objeto DouLicitacao que possui os atributos conforme imagem a seguir. Por fim, o objeto é encaminhado para a camada da aplicação que faz a persistência de dados no banco MongoDB.



Na imagem acima, é definida a classe DouLicitacao, ele tem como atributos informações pertinentes presentes nas páginas de licitação, além disso, são utilizados os annotations dataclass, que gera automaticamente getters, setters e construtores e dataclass\_json que gera automaticamente a transformação do objeto para Json.

Sendo assim, nessa fase do projeto foram utilizados os framewokrs/tecnologias:

- Python

- Pycharm

- Scrapy

- Splash

- Scrapy-Splash

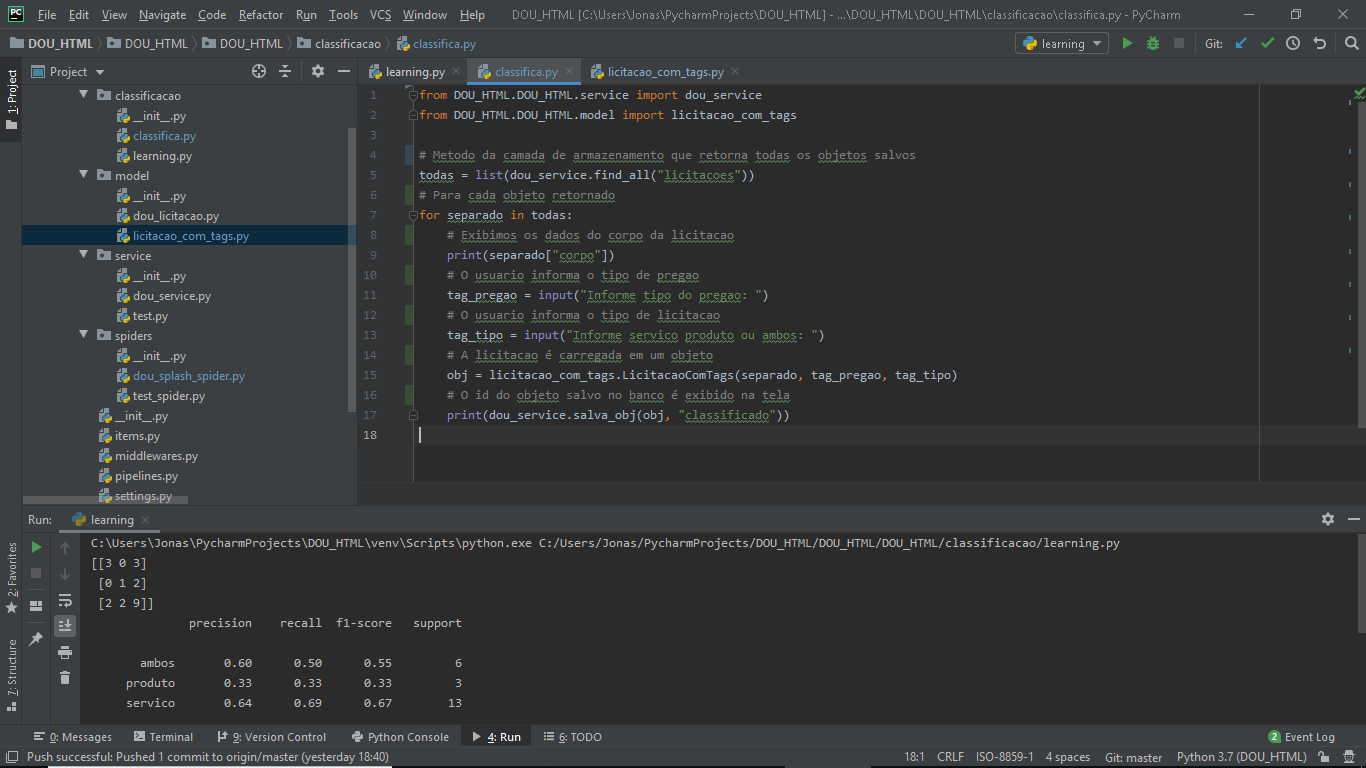
- Docker

- Dataclass

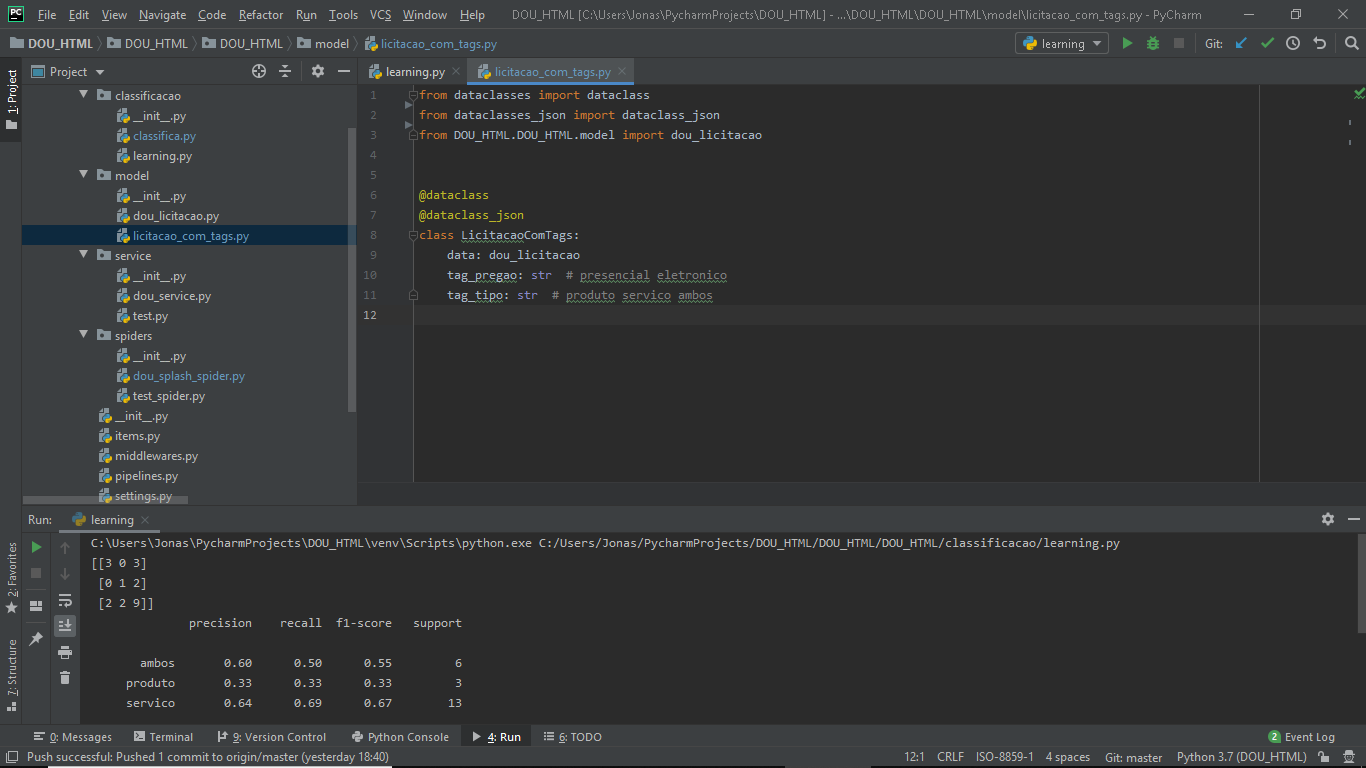
- Dataclass\_json

1. Fase de processamento:
   1. Apresentação das ferramentas;
   2. Apresentação das técnicas;
   3. Apresentação das integrações;
   4. Apresentação das fontes de dados.

Primeiramente, foi feita a classificação manual de cada uma das licitações coletadas, através do método apresentado abaixo:

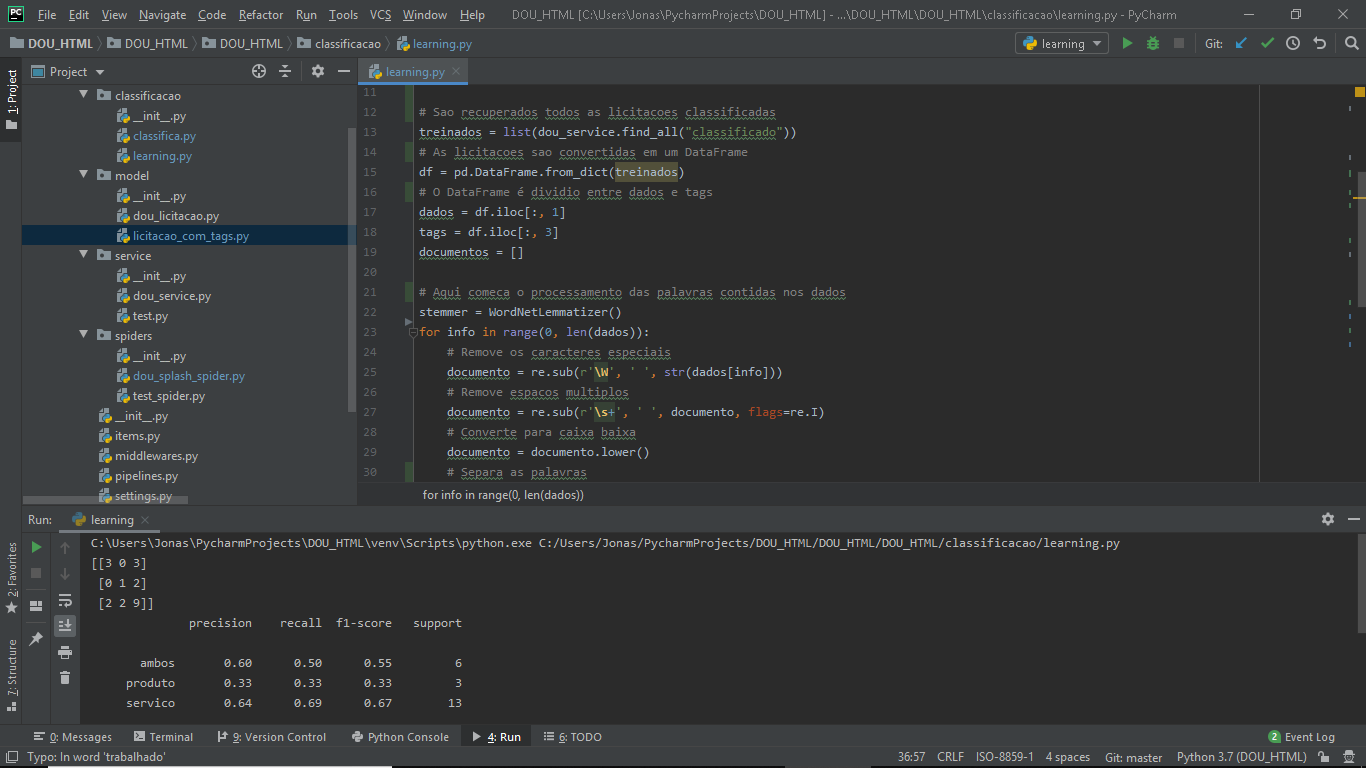


A função acima recupera cada uma das licitações coletadas, que estão salvas no banco NoSQL, em seguida, para cada uma das licitações, são exibidas as informações salvas do corpo da licitação e é feita a classificação de duas tags pregão e tipo, definidos valores padrão para cada uma delas, sendo que pregão pode ser presencial ou eletrônico e o tipo de licitação pode ser produto, serviço ou ambos. Ao final do método, é realizada a persistência desse novo objeto em um contexto diferente do banco de dados.



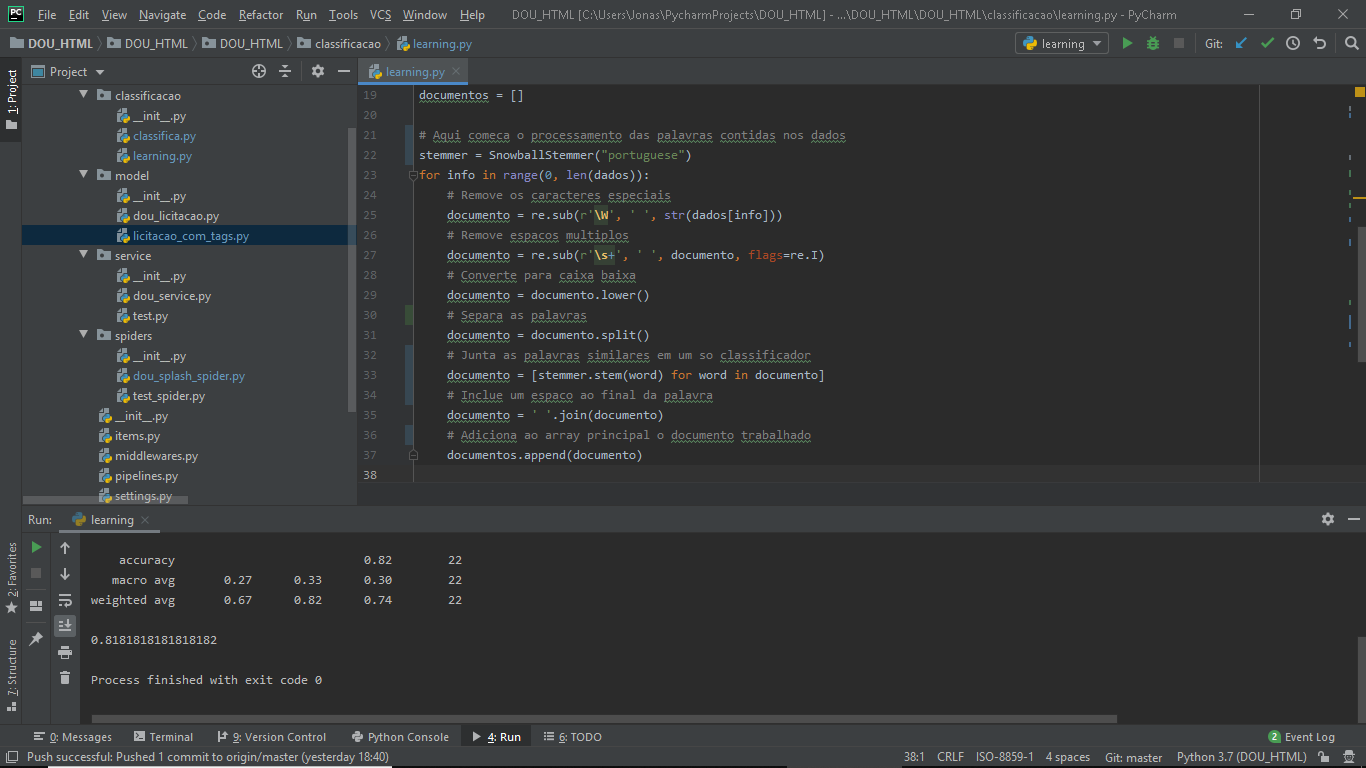
A imagem acima representa o objeto de licitação já classificado, seguindo os mesmos conceitos para o objeto dou\_licitacao, apresentado no tópico anterior desse relatório.

Nesse momento é executada uma nova camada da aplicação que realizada a classificação das licitações:



Primeiramente é realizada a carga dos dados em um objeto que é convertido para DataFrame, através da biblioteca Pandas, através de um método que recebe um dict, objeto retornado pelo banco de dados, e converte em um objeto DataFrame. Por fim, os dados do DataFrame são separados em dois novos arrays, um contendo as informações das licitações e outro contendo as tags classificadas, a separação é realizada através do método iloc, que recebe a linha e a coluna procurada. No caso acima, o operador “ : ” informa o método para buscar todas as linhas, em seguida é informada qual coluna do DataFrame deve ser retornada.

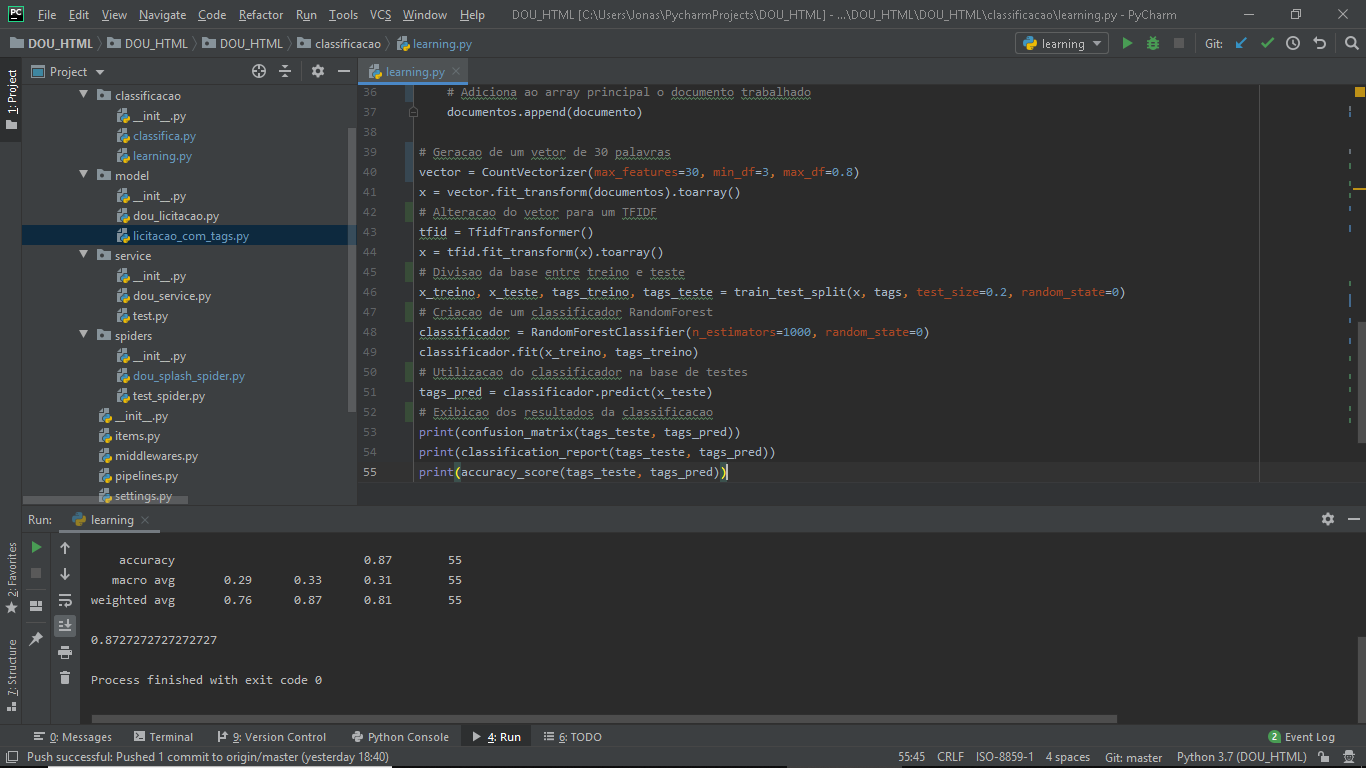
Em seguida, começa a fase de limpeza dos dados:



Nesse momento, o objeto SnowballStremmer é inicializado, informando a língua portuguesa como base, esse objeto, em seguida, é utilizado para reunir em um só token palavras similares como “gato” e “gatos”.

São removidos caracteres especiais, espaços múltiplos, o texto é convertido para caixa baixa, as palavras são separadas, em seguida são encaminhadas para o stemmer realizar a junção por similaridade, é adicionado um espaço ao início do documento, para evitar a junção com palavras do processamento anterior e por fim, são adicionados ao array de palavras já limpas.

Em seguida, começa a fase de Machine Learning do processo:



No primeiro momento é feita a criação de um objeto CountVectorizere que separa as 30 palavras que mais aparecem, em pelo menos 3 documentos e que não apareçam em mais de 80% dos textos. Esse objeto é utilizado para realizar a vetorização das palavras trabalhadas no passo anterior.

Então, o vetor é transformado em um TFIDF, que busca evitar que textos grandes tenham maior peso na classificação, trabalhando com a referência % da aparição das palavras em cada texto e não a quantidade absoluta de vezes que a palavra aparece no texto.

Em seguida a base é dividida entre teste e treino, a base de treino é utilizada no classificador RandomForest, para criar um algoritmo de classificação. Por fim, ele é utilizado na base de testes para prever as tags, os resultados são apresentados em seguida.

Nesse ponto do projeto foram utilizados:

- NLTK para o agrupamento das palavras

- RE para a utilização de expressões regulares

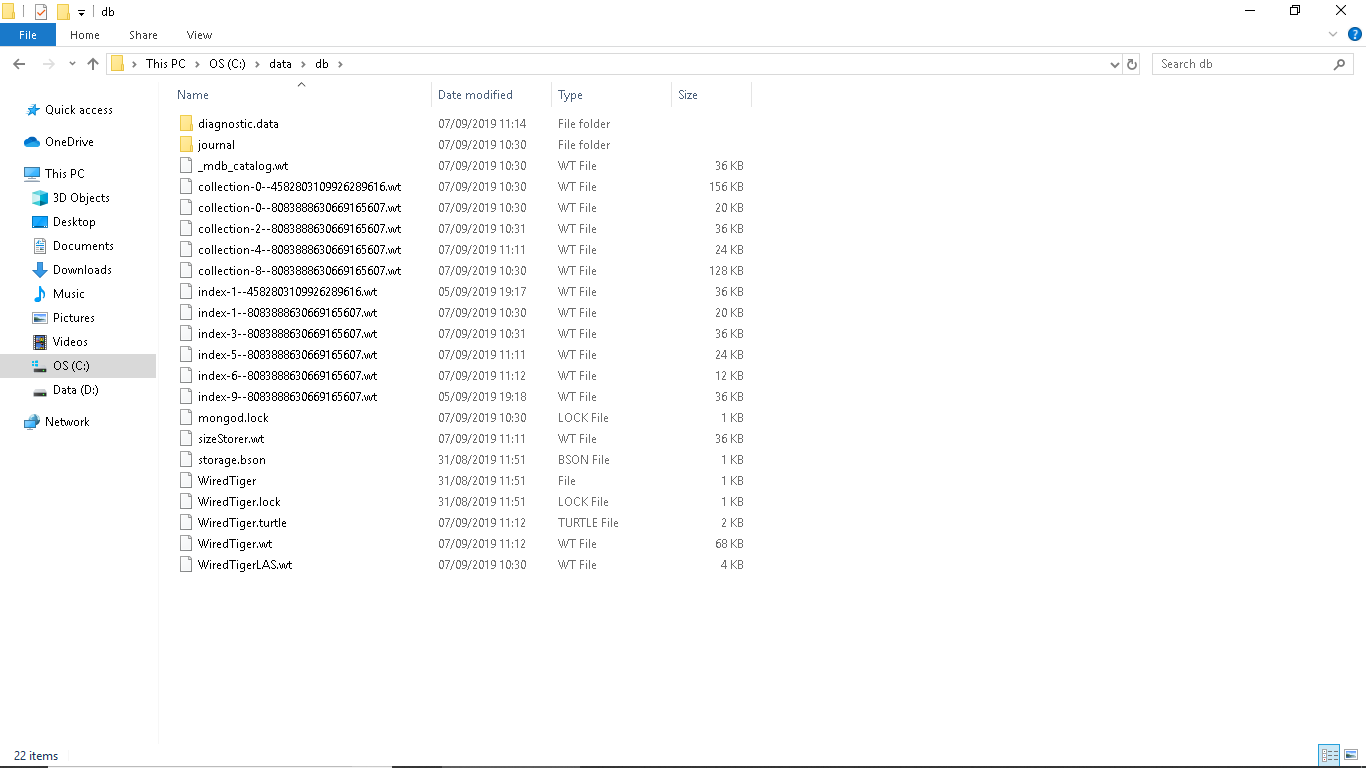
- SkLearn que fornece bibliotecas para aprendizado de máquina, separação das bases de teste e treino, vetorização das palavras, cálculo dos resultados

- Pandas para criação e utilização de DataFrames

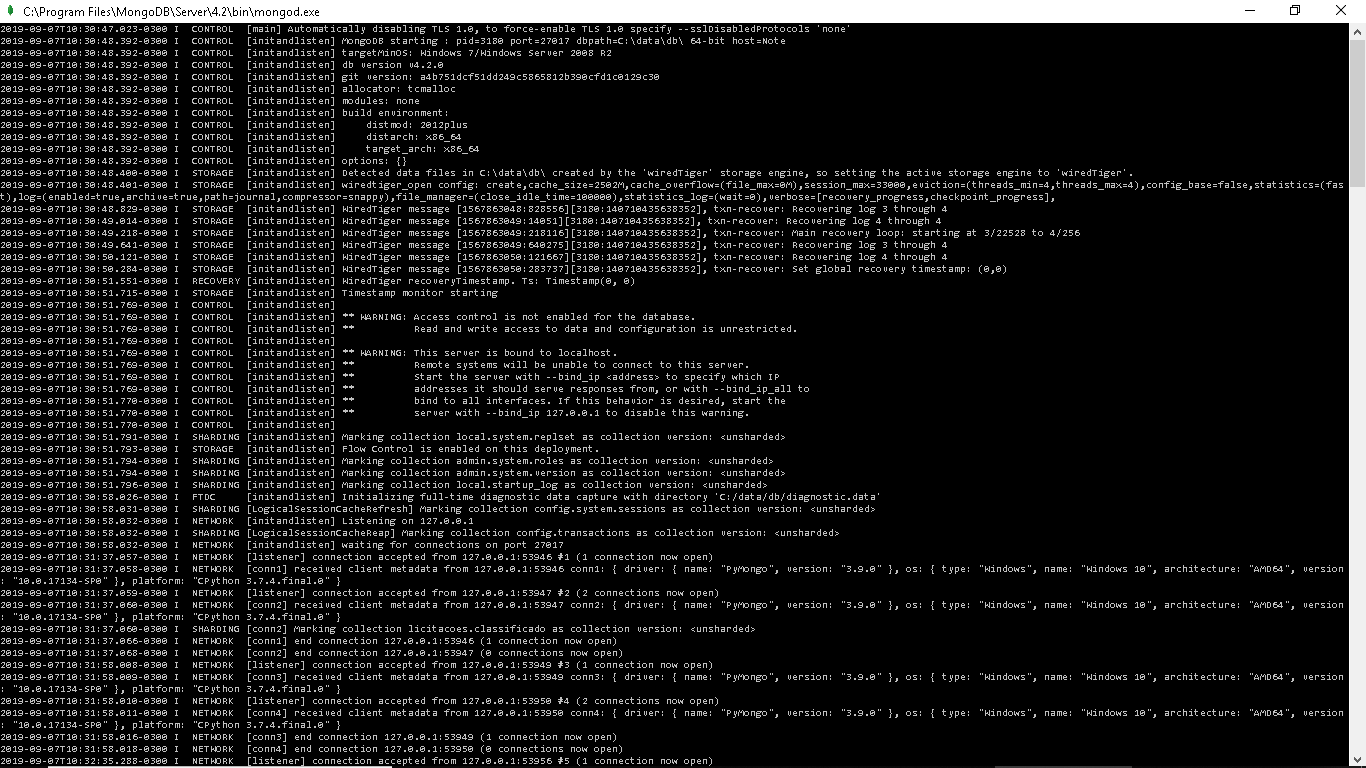
1. Fase de armazenamento.
   1. Apresentação das ferramentas;
   2. Apresentação das técnicas;
   3. Apresentação das integrações;
   4. Apresentação das fontes de dados.

Na fase de armazenamento do projeto, foi utilizado o MongoDB. Essa escolha foi devido a facilidade com o trabalho em objetos JSON, que são bem similares a estrutura de árvore do HTML que é coletado durante no início do projeto.

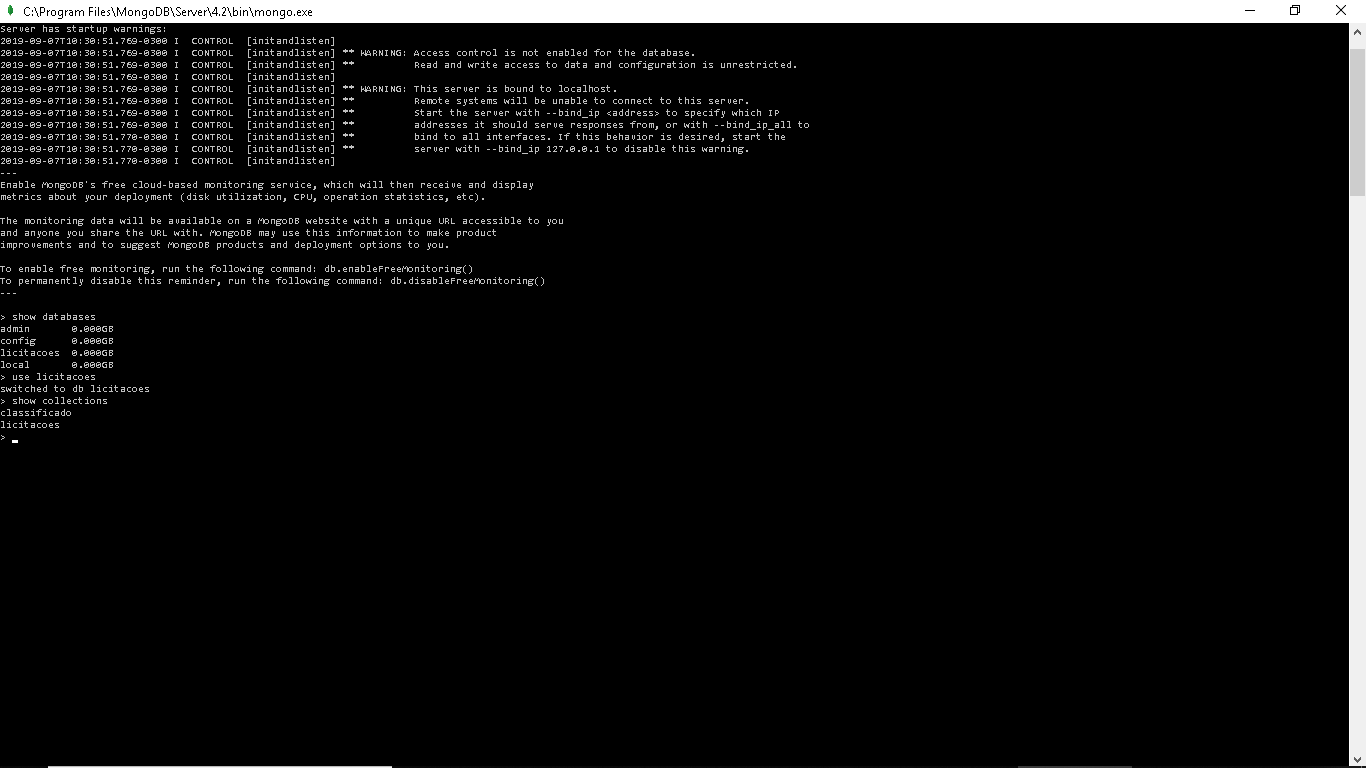
Foi realizada a instalação do MongoDB no Windows, sistema hospedeiro do computador. Para as configurações iniciais, foi necessária a criação do caminho de pastas C:\data\db\. Este é o diretório padrão que a instalação do MongoDB no Windows espera para salva os dados da estrutura de armazenamento:



Com o MongoDB instalado e o caminho de pasta já estruturado, é feita a inicialização do através da execução do arquivo mongod.exe, que instancia o Banco de Dados e permite acesso através do local host, na porta padrão 27017, conforme imagem na inicialização abaixo:

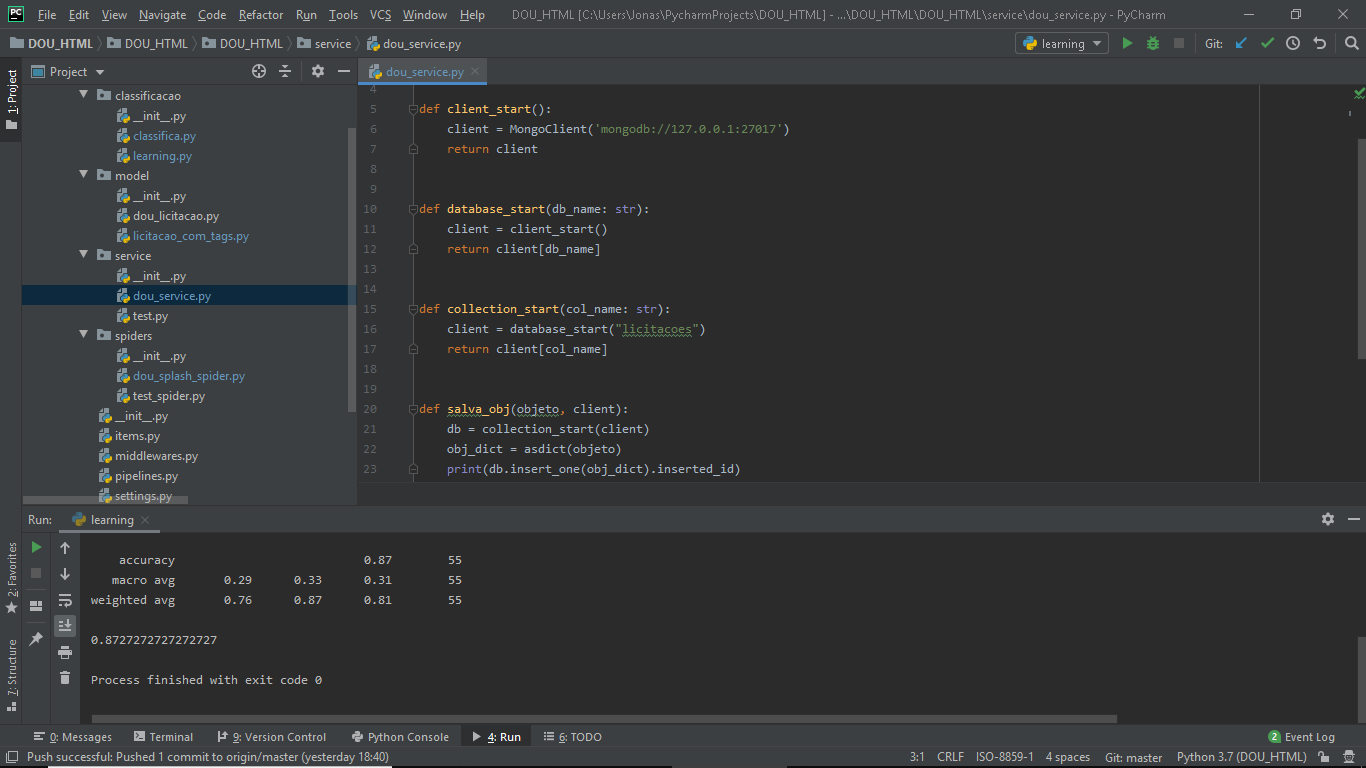


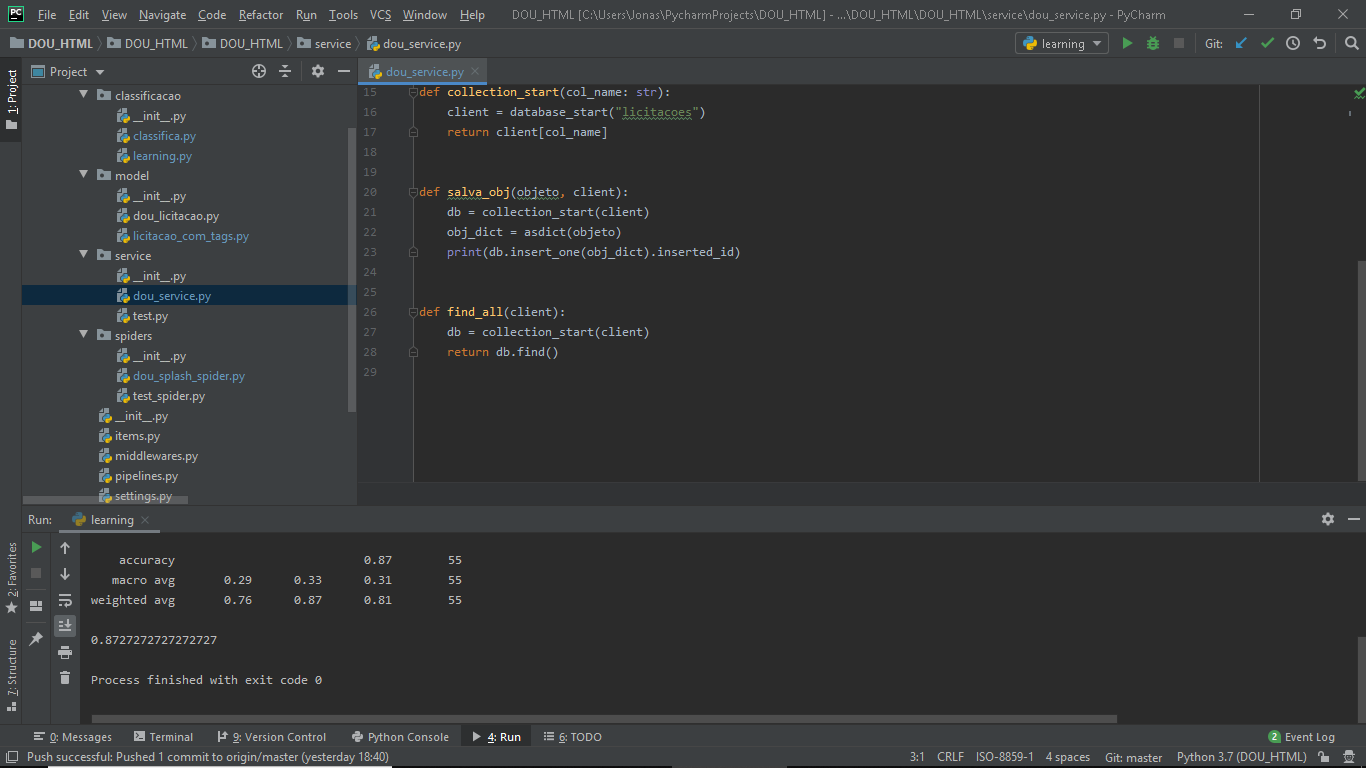
O Mongo Shell pode ser acessado através do executável mongo na pasta padrão de instalação, ele dá acesso ao um ambiente similar ao das demais ferramentas de banco de dados, onde pode-se executar comandos de busca, inserção, exclusão, etc:



No cenário proposto foram criadas duas coleções, uma com as licitações sem classificação e uma com as licitações classificadas.

Dentro da aplicação, foi criada uma cama de persistência conforme imagem abaixo:





Acima podemos ver as funções:

- cliente\_start(): aponta para o caminho do banco de dados e faz a conexão ao mesmo

- database\_start(db\_name): instancia um banco de dados de acordo com o nome recebido

- collection\_start(col\_name): chama a função database\_start e retorna o collection informado na chamada da função

- salva\_obj(objeto, cliente): chama a função collection\_start, de acordo com a collection informada e salva o objeto na collection instanciada.

- find\_all(client): recupera todos os documentos salvos em uma collection específica.

Nessa etapa do projeto foram utilizados:

- MongoDB para a instalação do MongoDB no Windows

- PyMongo para integração Python e MongoDB

1. Fase de visualização dos dados:
   1. Apresentação das ferramentas;
   2. Apresentação das técnicas;
   3. Apresentação das integrações.

**RESULTADOS**

1. Descrição e análise dos resultados alcançados:
   1. Resultados positivos encontrados (caso existam). Explique;
   2. Resultados negativos encontrados (caso existam). Explique.

Dentre os resultados positivos encontrados, podemos ressaltar:

- A fácil integração entre as ferramentas

**CONCLUSÃO**

1. Apresentação da conclusão:
   1. Principais contribuições que seu projeto gera aos envolvidos;
   2. Inovações, particularidades ou vantagens que o projeto/resultado possui em relação a similares;
   3. Limitações do projeto;
   4. Próximos passos necessários para que o projeto evolua/se desenvolva.