**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

**Pós-graduação *Lato Sensu* em Ciência de Dados e Big Data**

**Cesar Augusto Siqueira Santos**

**TÍTULO DO PROJETO**

Belo Horizonte

2022

**Cesar Augusto Siqueira Santos**

**TÍTULO DO PROJETO**

Trabalho de Conclusão de Curso apresentado ao Curso de Especialização em Ciência de Dados e Big Data como requisito parcial à obtenção do título de especialista.

Belo Horizonte

2022 **SUMÁRIO**

[**1. Introdução**](#_heading=h.gjdgxs) **4**

[1.1. Contextualização](#_heading=h.p6mam5rtf8p) 4

[1.2. O problema proposto](#_heading=h.s2y2hfthdyz2) 6

[1.3. Objetivos](#_heading=h.p3tr8prre52z) 7

[**2. Coleta de Dados**](#_heading=h.jmv4tl6qsj3t) **7**

[**3. Processamento/Tratamento de Dados**](#_heading=h.2et92p0) **8**

[**4. Análise e Exploração dos Dados**](#_heading=h.tyjcwt) **9**

[**5. Criação de Modelos de Machine Learning**](#_heading=h.3dy6vkm) **10**

[**6. Interpretação dos Resultados**](#_heading=h.1t3h5sf) **12**

[**7. Apresentação dos Resultados**](#_heading=h.4d34og8) **13**

[**8. Links**](#_heading=h.2s8eyo1) **14**

[**9.REFERÊNCIAS**](#_heading=h.zb4kfb3uqv7y) **15**

# 1. Introdução

## 1.1. Contextualização

Nessa seção você deverá introduzir e contextualizar o seu problema de interesse. Lembre-se: seu problema deve ser **REAL** e **RELEVANTE** (útil para sua organização para a sociedade).

Câncer é o principal problema de saúde pública no mundo, em paralelo, as doenças de pele são um dos principais tópicos debatidos nos meios acadêmicos.

Numa estimativa mundial conduzida no ano de 2018, aponta que ocorreram no mundo 18 milhões de casos novos de câncer e cerca de 9,6 milhões de óbitos. A incidência em homens (9,5 milhões) representa 53% dos casos novos, sendo um pouco maior nas mulheres, com 8,6 milhões (47%) de casos novos. (BRAY *et al*., 2018).

- https://www.inca.gov.br/estimativa/introducao  
 Tratando agora sobre o cânceres de pele, um dos mais comuns, este projeto tem ênfase na identificação de melanoma, o câncer de pele com maiores índices de mortalidade, ainda que represente apenas 1% dos cânceres de pele identificados, é de longe o maior causador de mortes.

Tomando por exemplo os Estados Unidos, para o ano de 2022 é esperado o diagnóstico de 99.780 novos casos de melanomas, distribuídos conforme: 57,30% para homens (57.180) e 42,70% em mulheres (42.600). Sobre mortalidade, é esperado que 7.650 pessoas venham a óbito em virtude do melanoma.

É fato que o melanoma atinge em sua maioria pessoas de pele branca, sendo 20 vezes mais comum do que em pessoas de pele preta. Sumarizando, o risco de desenvolver melanoma ao longo da vida para pessoas brancas é de cerca de 2,6%, para pessoas de pele preta apenas 0,1%.

O risco de melanoma aumenta conforme a idade avançada, a idade média de diagnóstico (nos Estados Unidos) é de 65 anos, entretanto o melanoma não é tão incomum entre abaixo dos 30, é na verdade um dos cânceres mais comuns entre jovens adultos, especialmente mulheres.

- https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics.html

A detecção precoce, consiste em diagnosticar o câncer ainda em um estágio inicial, possibilitando assim melhores resultados no tratamento, pode ser buscado com a investigação de sinais e sintomas como o aparecimento de pintas escuras com bordas irregulares acompanhada de coceira e descamação, ou até mesmo alteração de pintas já existentes, como aumento de tamanho, mudança de cor e forma, passando também a apresentar bordas irregulares.

Uma regra adotada internacionalmente é a do “ABCDE” , utilizado na identificação precoce de melanoma. Não é necessariamente 100% assertiva, mas pode indicar uma certa atenção e justificar uma consulta médica com um dermatologista:

* **Assimetria**: uma metade da mancha é diferente da outra;
* **Bordas irregulares**: contorno mal definido;
* **Cor variável**: presença de várias cores em uma mesma lesão (preta, castanha, branca, avermelhada ou azul);
* **Diâmetro**: maior que 6 milímetros;
* **Evolução**: mudanças observadas em suas características (tamanho, forma ou cor).

Discorrendo sobre o tratamento, o mais indicado para maioria dos casos é remoção da mancha de pelo através de cirurgia, porém radioterapia e/ou quimioterapia, podem ser utilizadas dependendo do estágio da doença. Quando acontece metástase (câncer já se espalhou para outros órgãos), o melanoma tem sido tratado hoje com novos medicamentos, que vem apresentando boas taxas de sucesso no tratamento, que hoje consistente em atrasar o avanço da doença o máximo possível, oferecendo assim, chance de sobrevida mais longa a pacientes que antes tinham um prognóstico mais desanimador.

- <https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>

## 1.2. O problema proposto

Nessa seção você deverá descrever o problema que você está propondo resolver através dos dados. Faça isso de forma muito clara. Uma dica interessante é utilizar a técnica dos [5-Ws](https://its.unl.edu/bestpractices/remember-5-ws). Responder algumas dessas perguntas te dará uma melhor visão do seu problema e da solução.

Os diversos cânceres de pele estão entre os cânceres mais comuns no mundo, embora o melanoma represente apenas 1% da variedade de cânceres de pele diagnosticados pelo globo, é de longe o mais mortífero. Se identificado em estágio inicial, pode realmente significar a sobrevivência de milhares de pessoas espalhadas pelo globo anualmente.

Este projeto visa classificar as imagens de câncer entre *melanocytic nevi* ou *naevus* (macha de pele comum e não cancerígena ) e melanoma(mancha de pele cancerígena) de maneira prática, auxiliando em diagnósticos precoces da doença.

Dessa forma, foram coletadas imagens de bases públicas disponibilizadas por iniciativas médicas ou acadêmicas. Neste projeto foram usados dados e imagens provenientes do HAM1001, extraídos do Kaggle, dados estes que foram reunidos entre o ano de 2016 e 2018. Além disso, os dados foram agrupados com uma base menor de dados provenientes de um Instituto Portugues, a base conhecida como MED-NODE e foi coletada até meados de 2015. Totalizando cerca de 8 mil imagens distribuídas em 6805 caracterizadas como *naevus* e outras 1183 já identificadas como melanoma.

A análise matemática deste projeto consiste em converter as imagens em matrizes normalizadas de números inteiros representando assim as nuances das imagens, através de modelos de deep learning já pré-treinados para classificar as imagens. Foi necessário normalizar, separar em diretórios e redimensionar as imagens antes de efetivamente processar os dados.

## 1.3. Objetivos

Este projeto possui apenas ênfase acadêmica, e visa colocar em prática diversos temas aprendidos durante a especialização *stricto sensu*, passando por coleta de dados, tratamentos e transformações, classificação e enriquecimento de dados, modelos de machine learning e deep learning*.* Todas estas etapas foram construídas utilizando Python como linguagem de programação, aliados a Jupyter Notebook. Logo, o objetivo não foi obter um aplicativo em estado de deploy em produção ou software para identificar melanoma ou mesmo substituir exames clínicos performados por médicos, nem mesmo altos índices de acurácia ou mesmo explorar em demasia atributos e funções de diferentes modelos de machine learning. O foco foi realmente a exploração e aprimoramento de conhecimentos através de uma atividade prática atrelada a realidade.

Assim sendo, o objetivo deste projeto foi testar pelo menos três modelos de classificação de imagem e definir também o modelo mais assertivo de machine learning para identificar melanoma, através do diagnóstico de imagens de manchas de pele.

# 2. Coleta de Dados

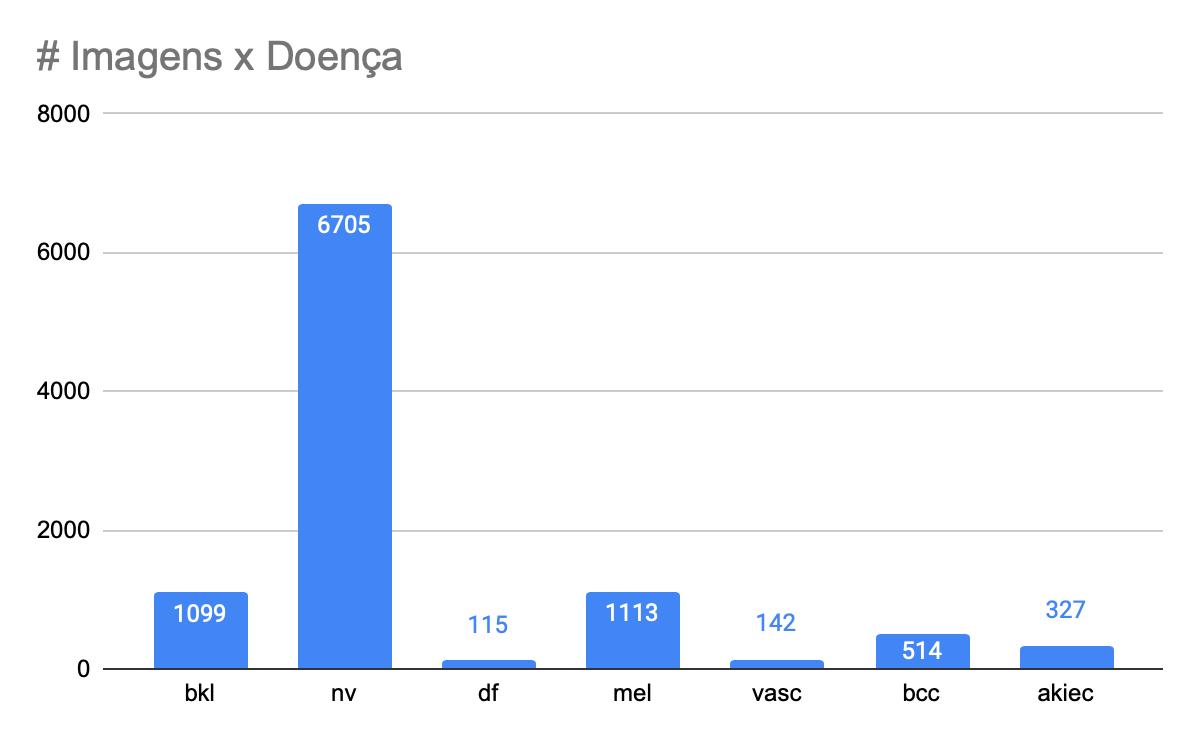
O tema deste projeto tem sido muito trabalhado por vários projetos ao longo dos últimos anos, porém existem apenas algumas bases muito populares, praticamente todas providas pela mesma fonte a ISIC que anualmente (desde 2016) tem lançado um desafio de machine learning, disponível no Kaggle, e providenciado datasets com milhares de imagens com este tema. Entretanto, como exigência do projeto tive de buscar uma base de dados distinta e após pesquisa encontrei um dataset pequeno, produzido pela MED-NODE mas que me fornece imagens de *nevuas* normais e melanoma. A fim de unir as bases, precisei reduzir o escopo do dataset provido pela HAM10000 (imagens colhidas pela ISIC em 2018), em apenas *Nevuas* normais ou melanomas.

Segue o detalhamento das bases de imagens e como elas foram unidas:

## 2.1 HAM10000:

*Human Against Machine with 10000 training images ou* humanos contra máquinas usando 10000 imagens para treinamento, em tradução literal. Tem este nome pois foi usada em desafios de machine learning, os dados foram coletados de diferentes populações, o dataset final é composto por 10015 imagens dermatoscópicas, que podem ser usadas para treinar diferentes tipos de modelos ou com propósitos distintos. Esta base de imagens contém um acervo representativo de imagens e dados da mais importantes lesões de pigmentação de pele, dentre elas:

| **Abreviação** | **Nome da Doença** | **Número de registros** | **Mantido?** |
| --- | --- | --- | --- |
| akiec | Doença de Bowen | 327 | Não |
| bcc | Carcinoma basocelular | 514 | Não |
| bkl | Verruga seborreica | 1099 | Não |
| df | Dermatofibroma | 115 | Não |
| mel | Melanoma | 1113 | Sim |
| vasc | Lesões Vasculares | 142 | Não |
| nv | Nevo Melanocítico ou *Naevus* | 6705 | Sim |



Mais de 50% das lesões presentes na base de dados foram confirmadas através de histopatologia, estudo de uma doença a partir de um conjunto de células, também conhecido como biópsia. Os outros resultados foram inferidos através de exames, consenso de especialistas ou confirmação através de microscopia confocal sem extração. Dentro da base de dados, existem algumas lesões com mais de uma imagem.

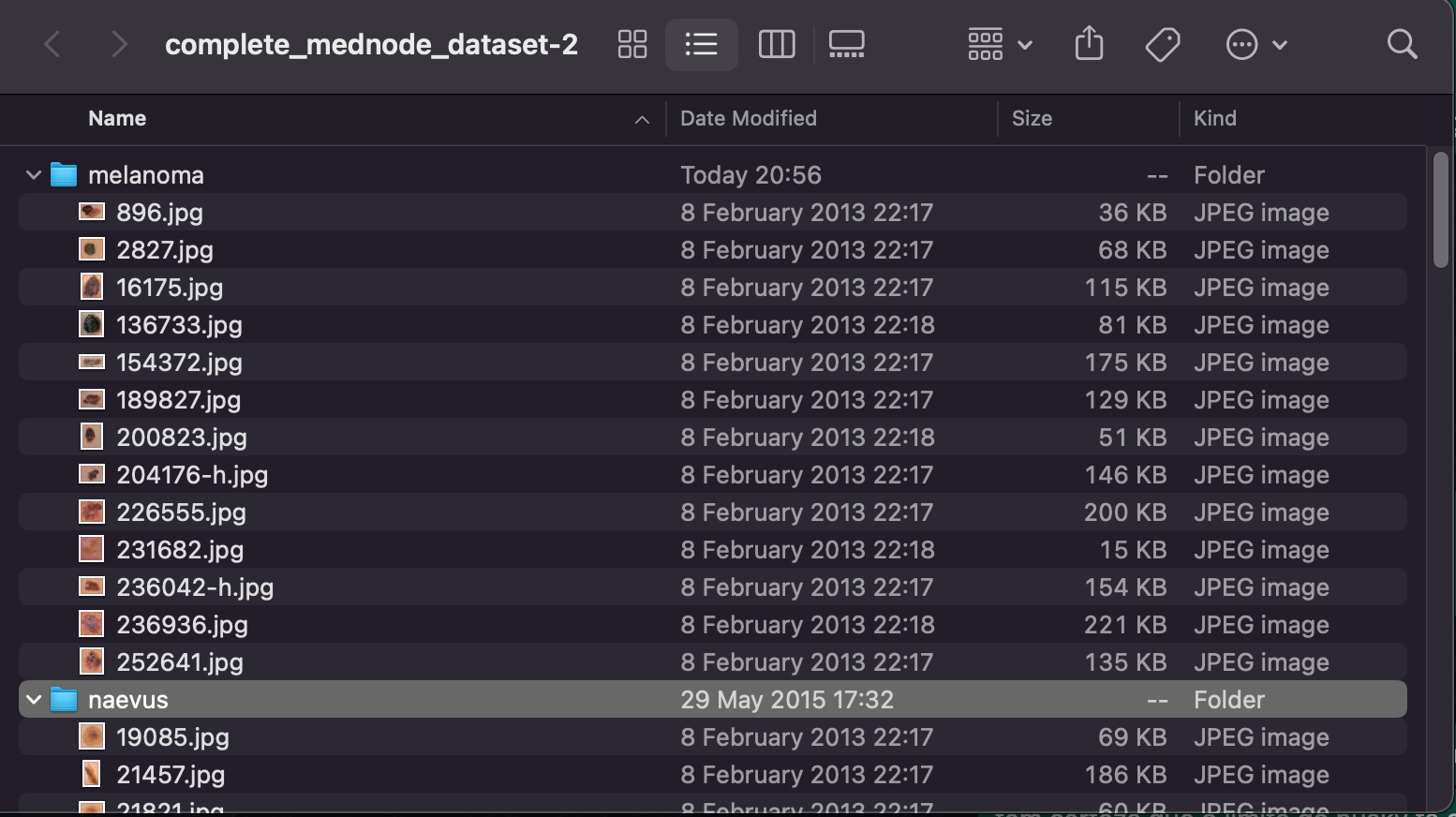
Junto a base de dados também foi providenciado um arquivo CSV contendo as seguintes informações:

| Nome do campo | Descrição | Tipo |
| --- | --- | --- |
| lession\_id | Identificador da lesão | texto |
| image\_id | Identificador da imagem | texto |
| dx | Abreviatura da lesão constatada | texto |
| dx\_type | Método de confirmação da lesão | texto |
| age | Idade do paciente | inteiro |
| sex | Sexto do paciente | texto |
| localization | Parte do corpo que contém a lesão | texto |
| dataset | Dataset de origem | texto |

Entretanto, dado os requisitos deste projeto, foi necessário abrir mão de várias imagens e também registros de metadados, em virtude da natureza do próximo conjunto de dados a ser explorado. Esta base está disponível para download pelo site de Harvard: <https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T>

## 2.2 MED-NODE:

Conjunto de imagens reduzido, composto por 70 imagens de melanoma e 100 *naevus*, estas imagens foram coletadas e arquivadas pelo Departamento de Dermatologia da *University Medical Center Groningen (UMCG)*, localizada nos Países Baixos. Este dataset foi coletado para desenvolvimento e testes para de um sistema para identificação de câncer de pele através de imagens macroscópicas.

Este dataset não fornece nenhum arquivo de CSV contendo metadados ou algo do gênero, ele apenas consiste num arquivo zip contendo as imagens em formato JPEG, em diretórios separados, conforme imagem abaixo:  


A base de imagens encontra-se disponível para download pelo site:  
[complete\_mednode\_dataset.zip](http://www.cs.rug.nl/~imaging/databases/melanoma_naevi/complete_mednode_dataset.zip)

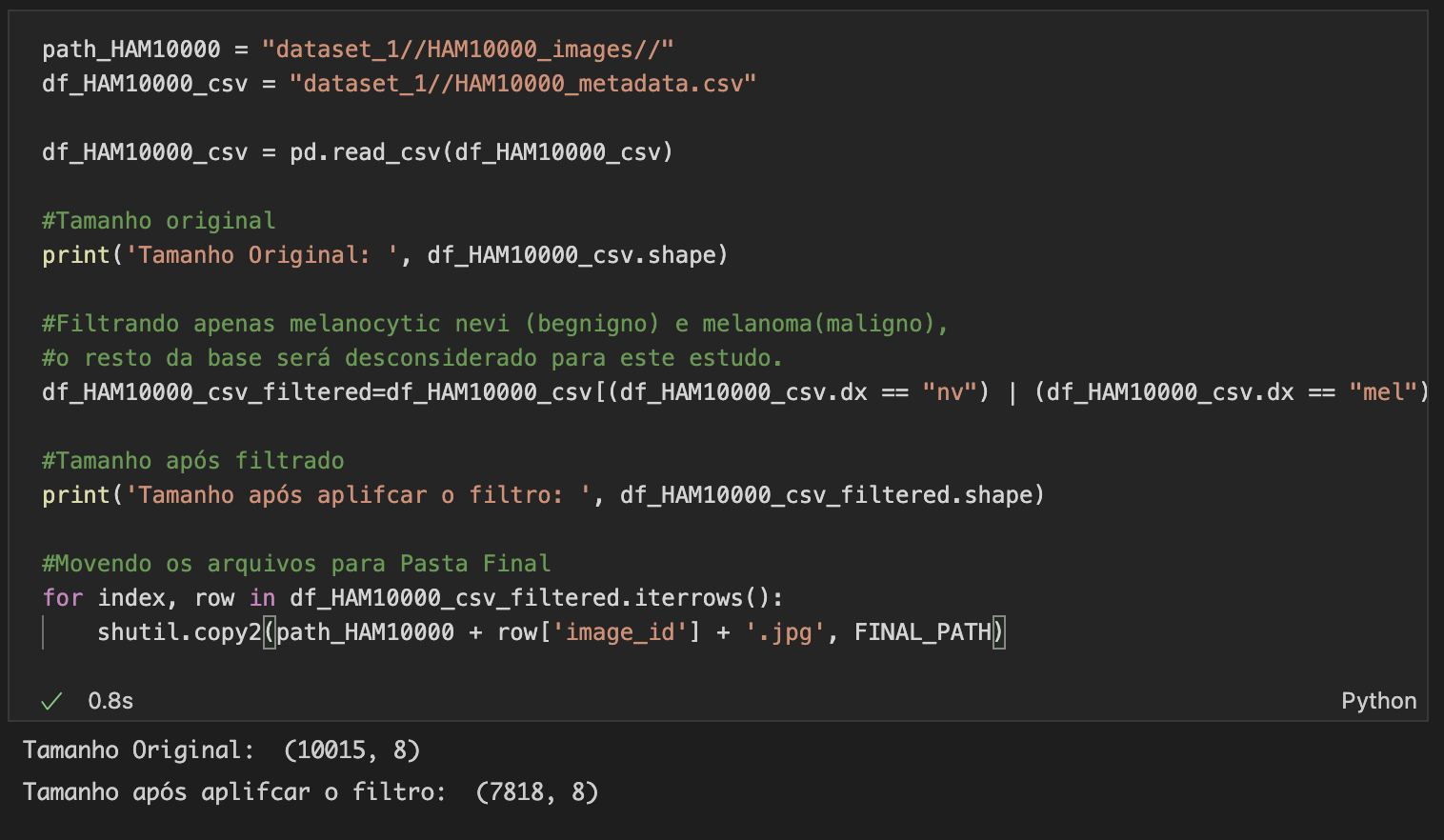
# 3. Processamento/Tratamento de Dados

Ambos os datasets possuíam suas peculiaridades, em um ( HAM10000 ) temos mais doenças de pele, inclui ainda um arquivo de metadados que poderia enriquecer o modelo, melhorando acurácia, considerando que informações como região do corpo que apresenta a lesão e também idade do paciente são informações bastante relevantes até mesmo para avaliação de um profissional de saúde. A base da MED-NODE por outro lado possuía poucos registros e apenas duas classificações feitas através de diretórios.

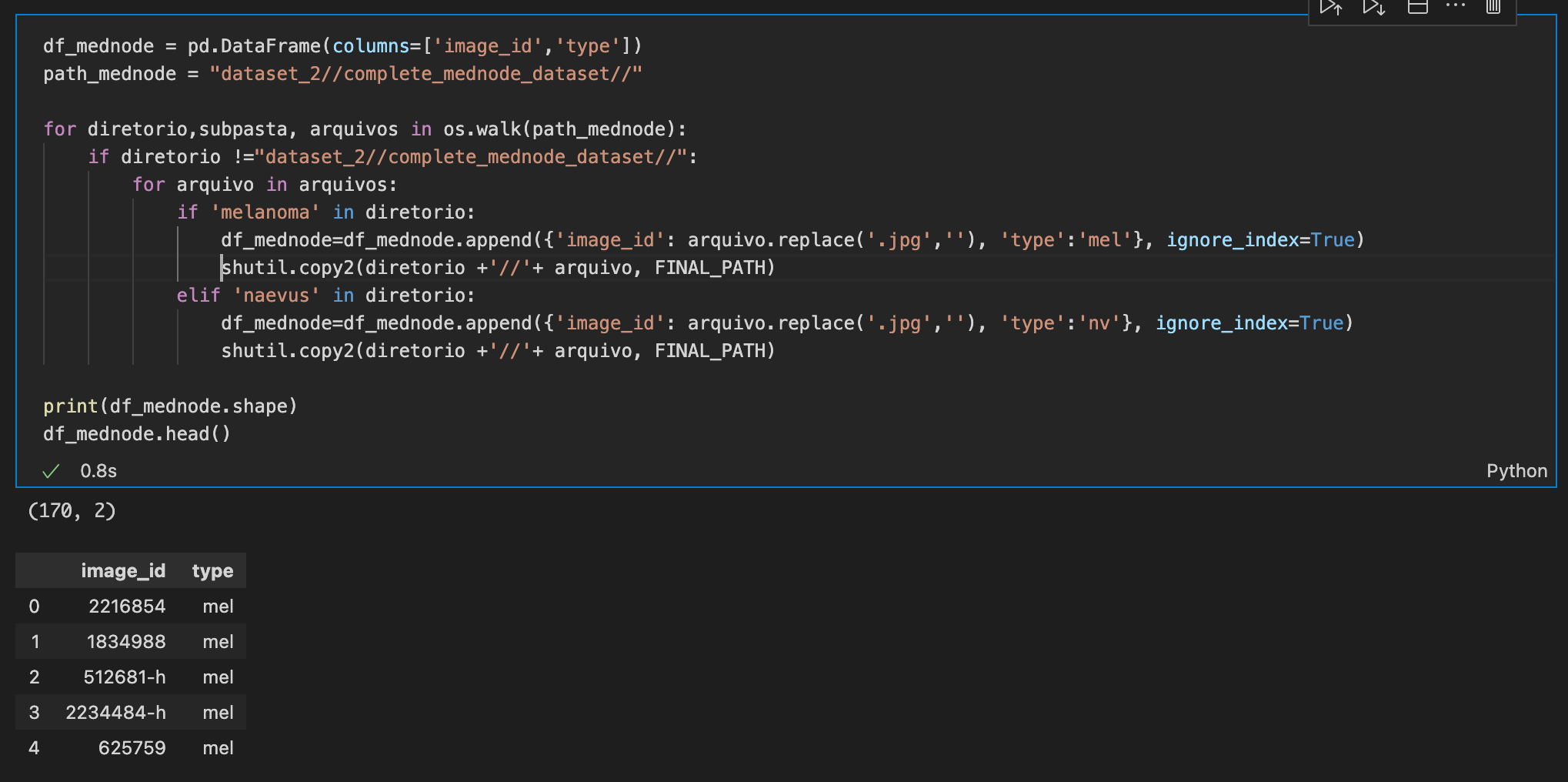
Dessa forma foi necessário fazer uma limpeza nos dados vindos da HAM10000, eliminando portanto cerca de 2200 imagens referentes a doença de Bowen, carcinoma basocelular, verruga seborréica, dermatofibroma e lesões vasculares, além do arquivo de metadados, que foi usado apenas no momento de tratamento dos dados para classificar as imagens e carregá-las para um diretório correto. Na sequência o arquivo também foi descartado e não foi usado durante o treinamento dos modelos.

O tratamento em si de ambas as bases se deu através de bibliotecas python de manipulação de arquivos e também pandas para ler e compor um arquivo destino por parte de ambas as bases de dados.

Para tratar a base da HAM10000, foi necessário ler o arquivo CSV a fim de identificar a classificação das lesões e também excluir aquelas que não fariam sentido deste projeto, para isso exclui do dataframe (criado pela leitura do arquivo de metadados providenciado) filtrando apenas as lesões referentes a *naevus* ou melanoma.



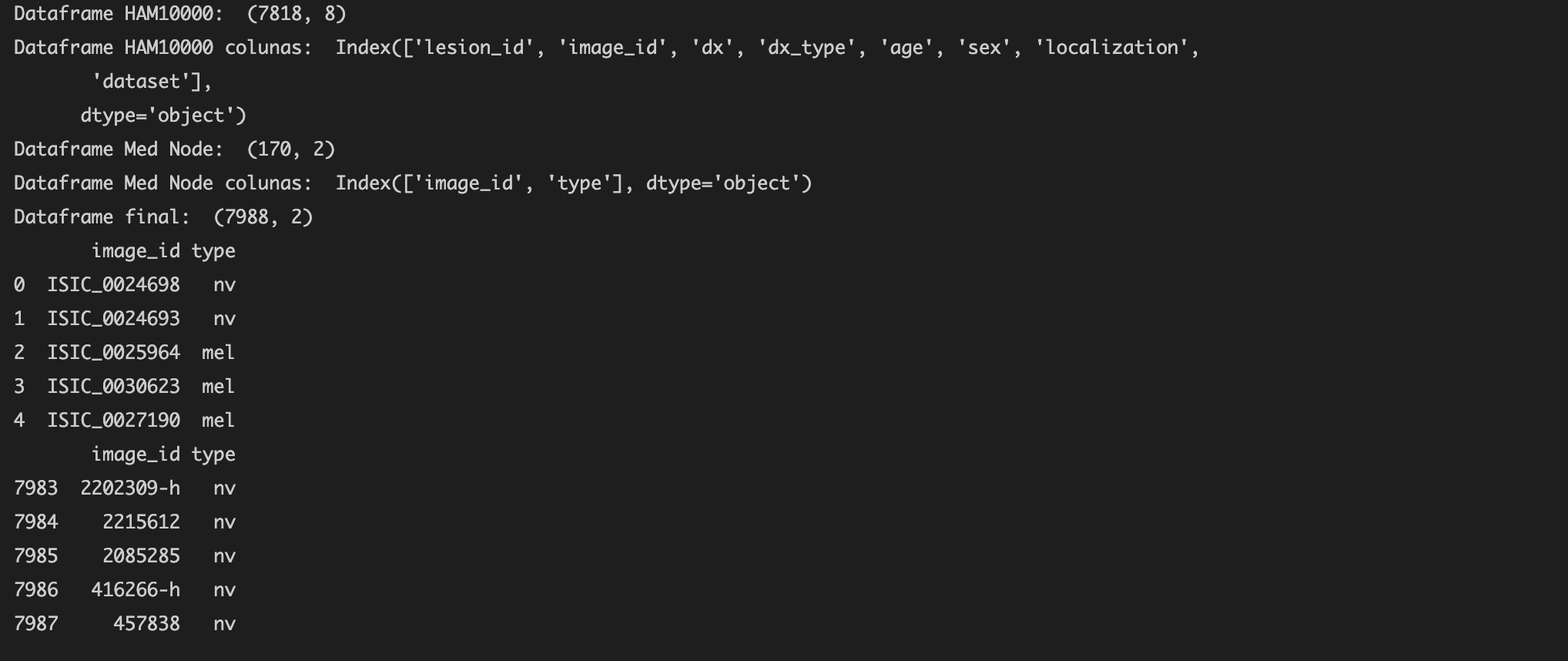
A imagem acima retrata a leitura do CSV, o filtro do dataframe e também o processo de cópia de todos os arquivos referentes a melanoma e *naevus* para o FINAL\_PATH. Também percebe-se a redução do dataframe que antes continha 10015 imagens e após o filtro caiu para 7818 imagens.

Por parte da base MED-NODE, nenhum tratamento foi necessário no sentido de limpeza ou tratamento dos dados, foi necessário apenas fazer a leitura das imagens nos diretórios correspondentes e juntar todas as imagens, após classificá-las. Conforme descrito pelo código presente na imagem abaixo:  


O trecho de código acima destacado, faz a leitura dos diretórios e imagens provenientes da base MED-NODE, foi necessário criar um Dataframe pandas contendo o ID da imagem, nesse caso preenchido pelo nome do arquivo e também o tipo da lesão , extraído pelo nome da pasta mãe, se a imagem está na pasta melanoma receberá o label de mel, se estiver na pasta naevus, receberá o label de nv. o dataframe criado será unido posteriormente ao dataframe tratado do HAM10000. Além de construir o dataframe, o código destacado acima também copia os arquivos, utilizando da biblioteca python shuttil, para um diretório chamado de FINAL\_PATH, onde todas as imagens do projeto ficarão, sejam melanoma ou *naevus*.

Após o tratamento individual de ambas as bases e também o mover das imagens para o FINAL\_PATH, o código abaixo retrata o tratamento e append dos dataframes, tornando-os em uma único flat file (CSV). Dessa forma na hora de criar os modelos, basta ter acesso ao arquivo de metadados (CSV) com a classificação de imagens e as imagens em si.



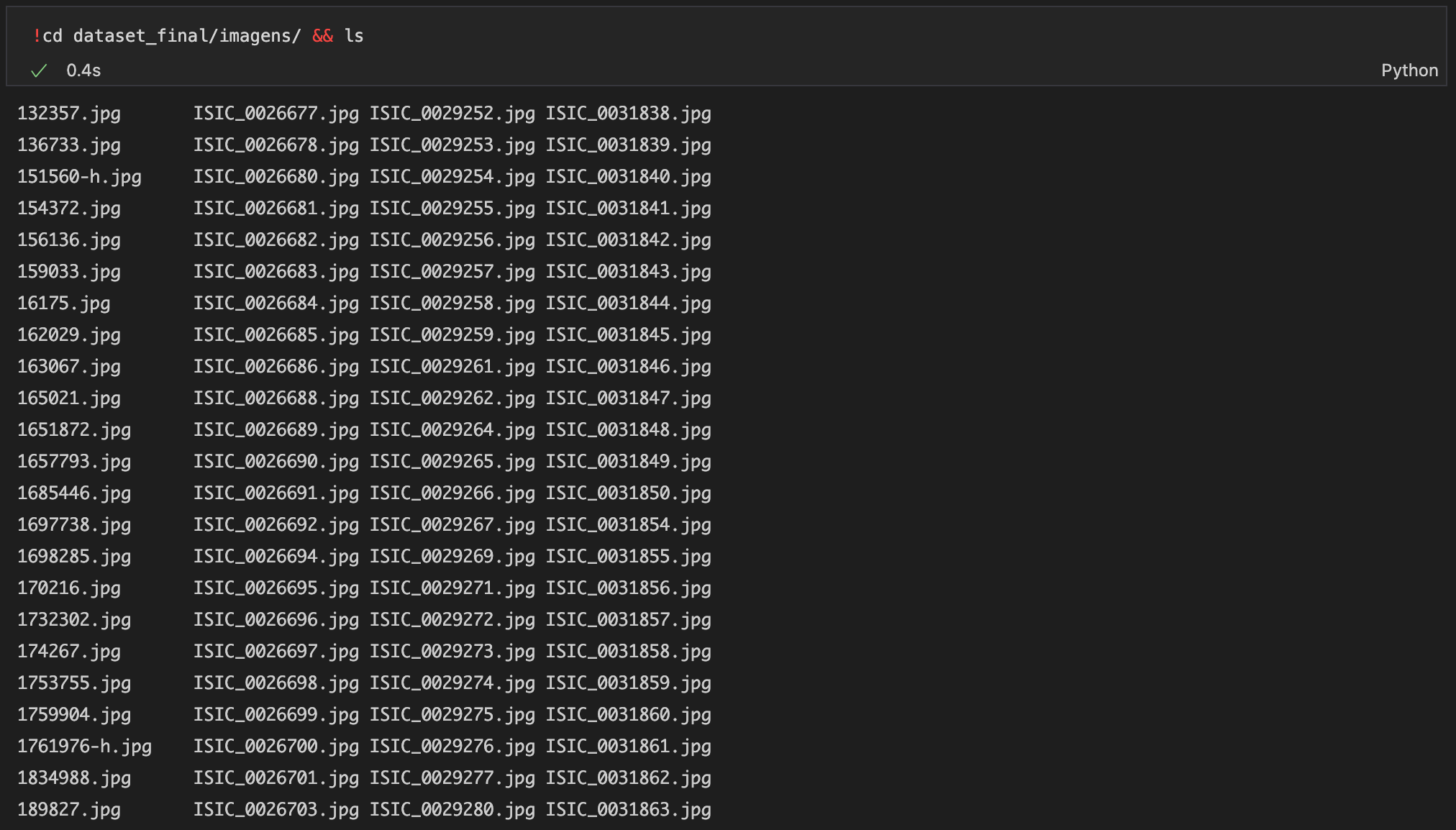


Sobre os resultados e outputs do código acima, podemos ver que por parte do HAM10000 foram ingeridas 7818 imagens, enquanto do MED-NODE apenas 170 imagens (todas que foram fornecidas), o dataframe final ficou com 7988 imagens e duas colunas: identificador único da imagem e tipo de lesão.

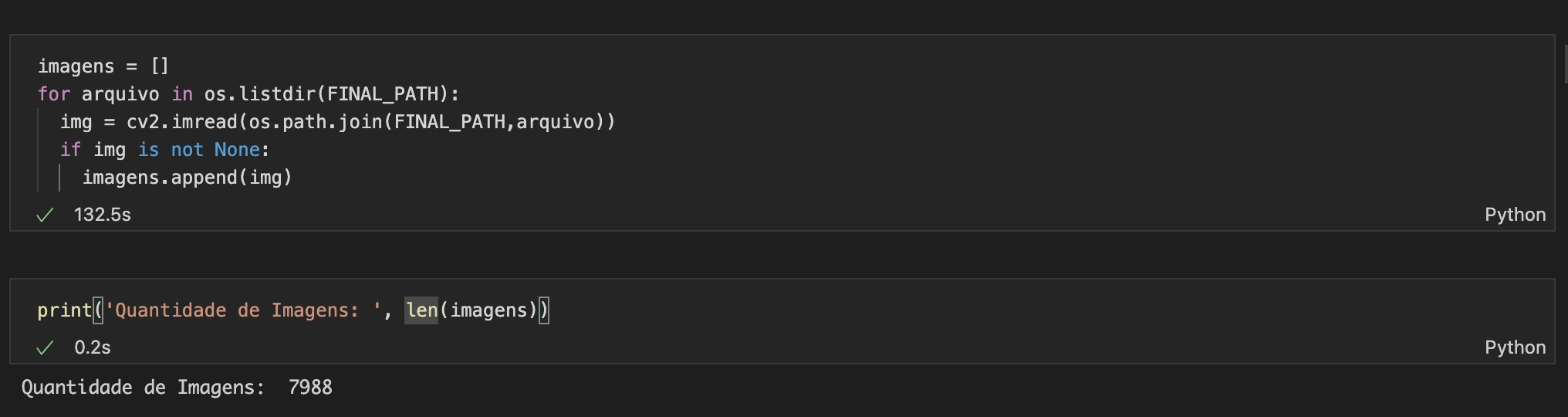
Embora o requisito do projeto fosse uma base de apenas 1000 registros, tomei a liberdade de manter mais registros a fim de obter uma melhor acurácia do modelo, considerando que o input do modelo será apenas as imagens em si.

Nessa seção você deve deixar registrado todo o processamento e tratamento feitos sobre os dados obtidos. É importante que você informe a quantidade de registros obtidos, a quantidade de registros duplicados ou com informações ausentes, que tratamento você deu para cada problema encontrado em seus datasets, etc. Você deve descrever cada passo de forma minuciosa, de forma que outra pessoa consiga reproduzir o seu processamento/tratamento de forma precisa. Justifique as decisões tomadas no tratamento dos dados. Por exemplo: para os valores ausentes para o campo X eu decidi preenchê-los utilizando o cálculo da média aritmética pelo motivo ..., e então justifique sua decisão.

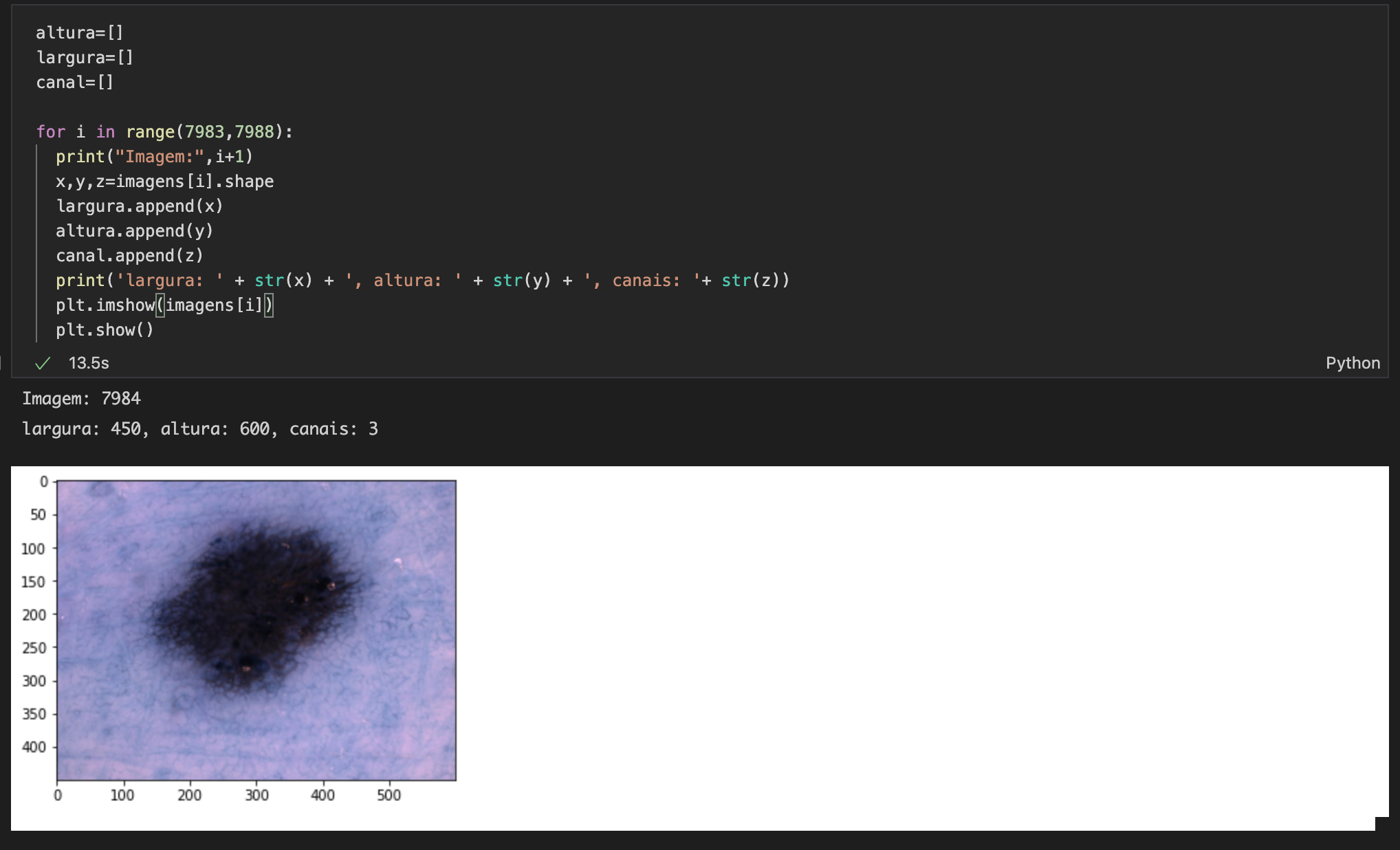
# 4. Análise e Exploração dos Dados

Para este projeto, não foi necessário realizar grandes análises ou mesmo exploração dos dados, ainda assim precisei tratar alguns outros pontos, num primeiro momento acessei a pasta definida como FINAL\_PATH e listei os arquivos a fim de confirmar que todos existiam. Conforme evidencia providenciada abaixo:  


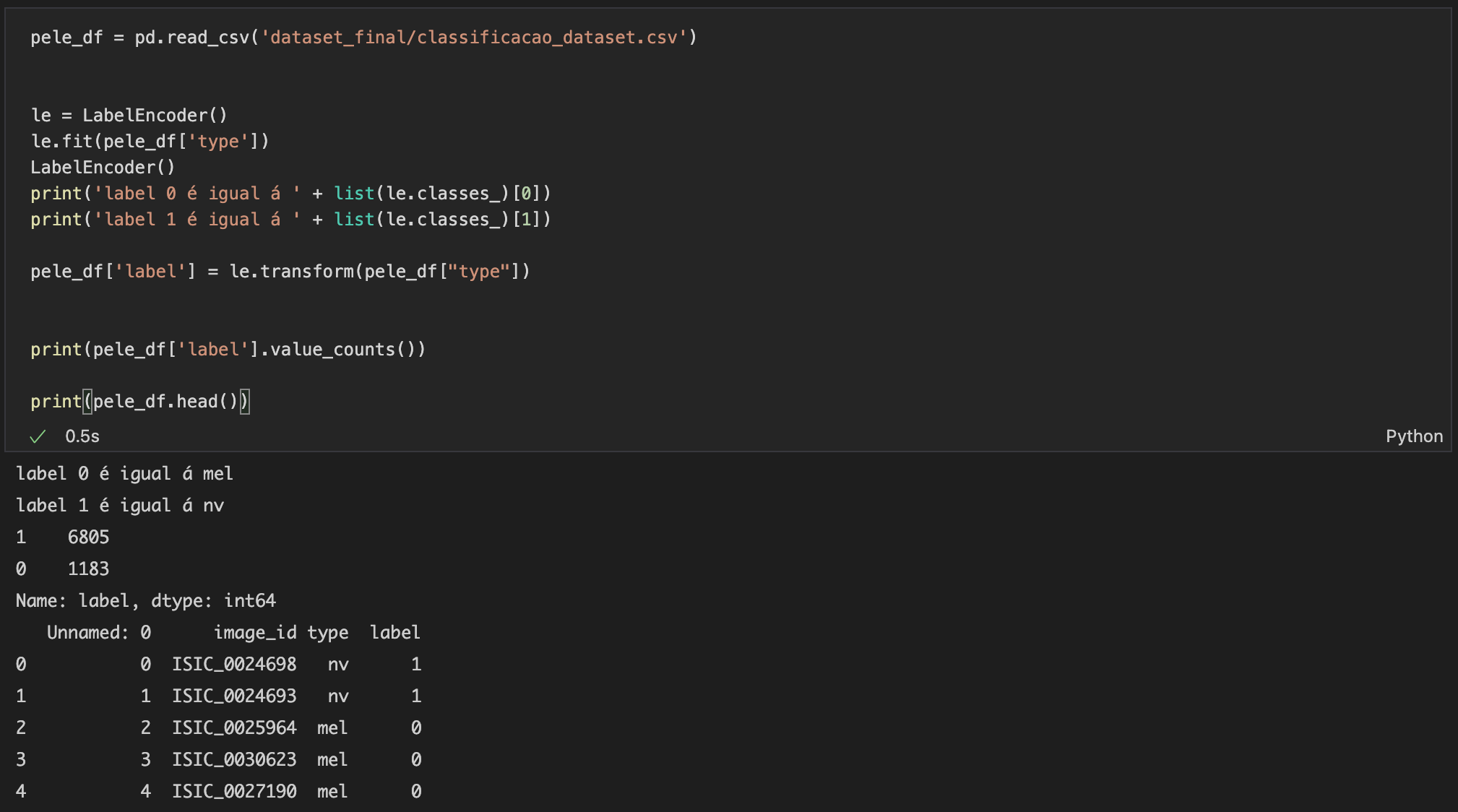
Em seguida listei a quantidade de imagens dentro da pasta, confirmando a existência de todas as 7988 imagens unidas pelo processo de tratamento dos dados.



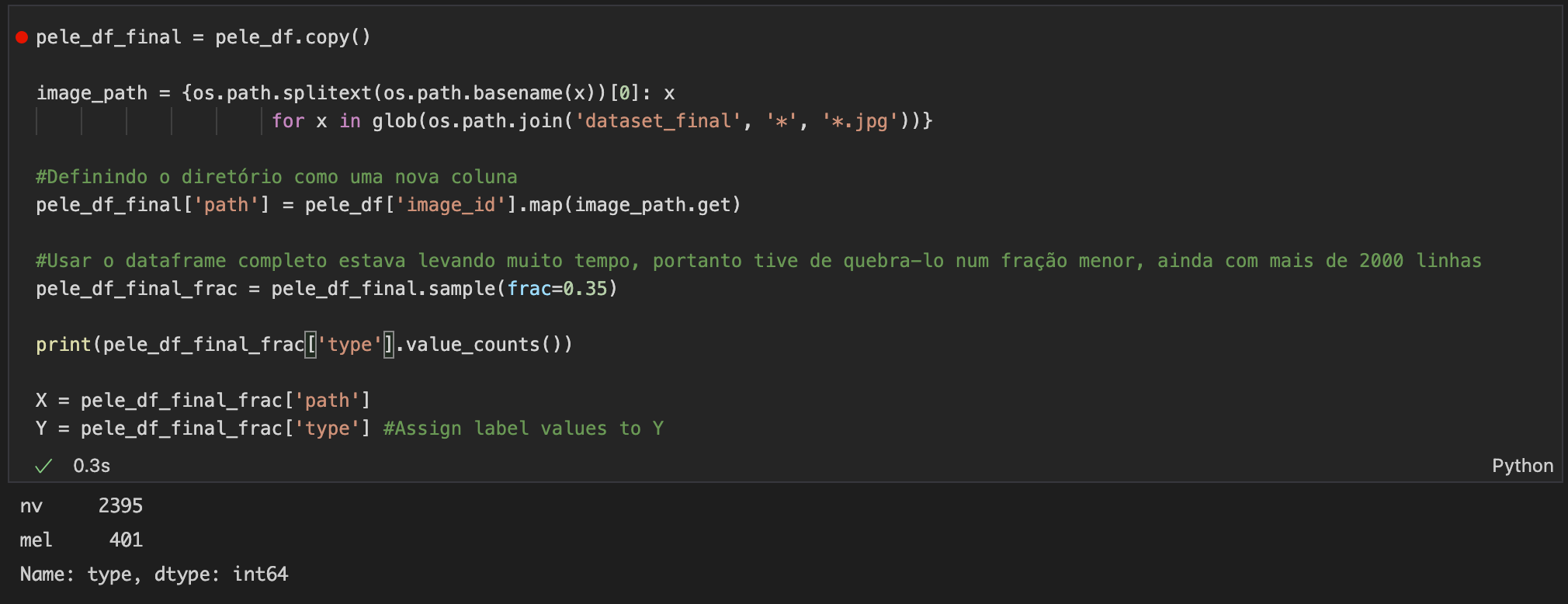
A fim de mostrar algumas das imagens e também confirmar informações como altura, largura e número de canais, representa o número de componentes usados para representar cada pixel, 3 canais por exemplo representam RGB (Red, Green, Blue), obtidos através da biblioteca Python OpenCV.



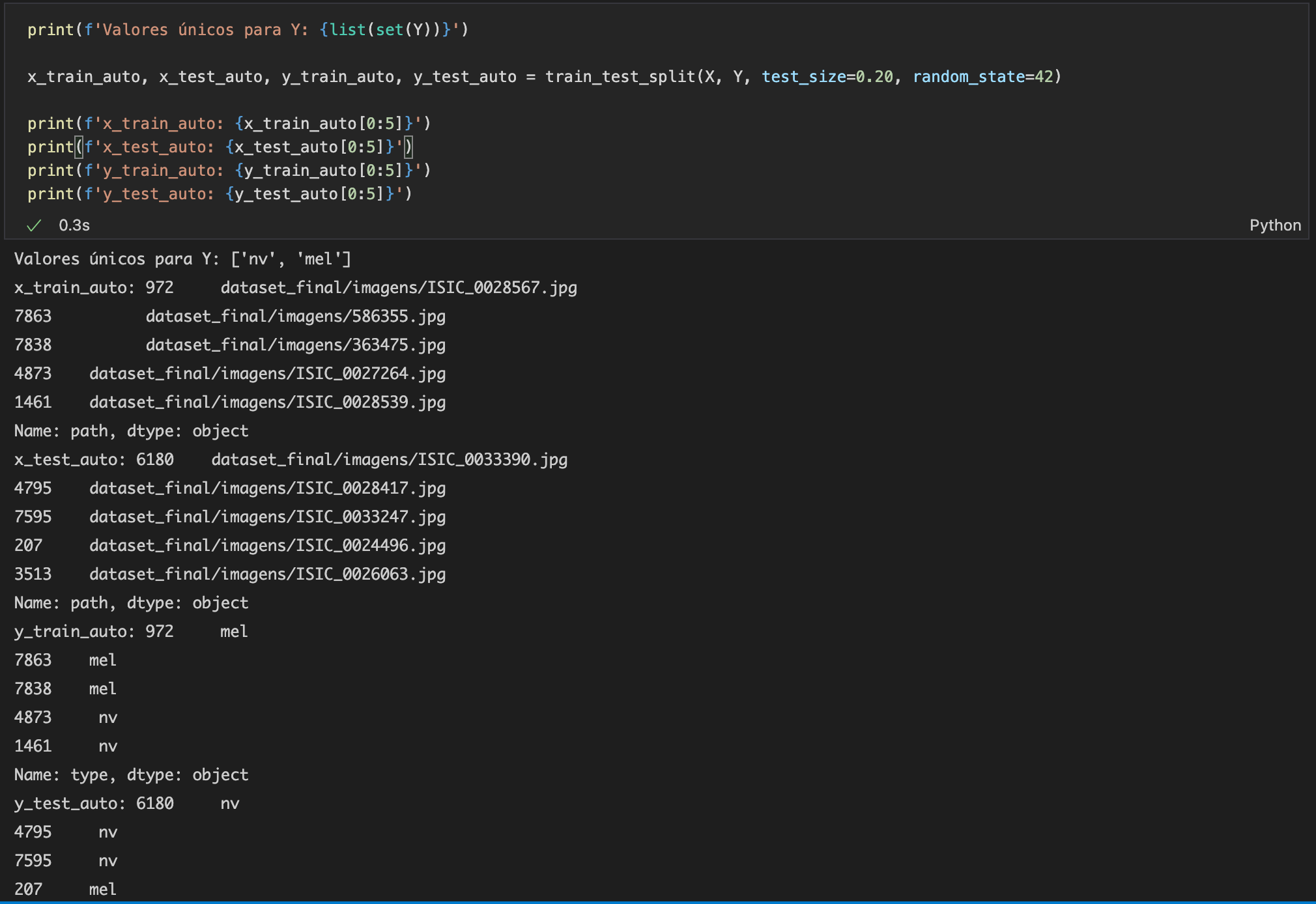
Por fim, antes de efetivamente fazer a criação de modelos, foi necessário ainda separar e classificar os arquivos com label antes de disponibilizá-los aos modelos. Para isso fiz a leitura do arquivo CSV gerado durante a etapa de Processamento/Tratamento de dados, além disso retornei também o as quantidades, mostrando que 6805 imagens são de *naevus* e 1183 imagens são de melanoma. O dataframe em si é composto por identificador de imagens, tipo e também label de maneira binária.



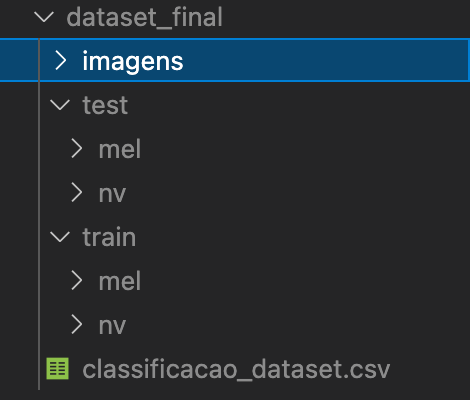
Seguindo com a análise e exploração dos dados, a base totalizava 7988 imagens, um eventual processamento de uma base desse tamanho seria muito custoso e tomaria demasiado tempo, decidi portanto reduzir a base de imagens para aproximadamente 3000 imagens, utilizando 35% da base através de um fracionamento. Sendo 2367 imagens classificadas como *naeuvs* e 429 imagens classificadas como melanoma. Por fim, atribui ao eixo X a coluna de path contendo o arquivo da imagem que será analisada, ao eixo Y foi atribuído o label de classificação entre nv e mel.



Na sequência separei base de treino e validação, usando a função *train\_test\_split* fornecida pela biblioteca *sklearn.model\_selection*. A base foi separada usando o paradigma de 80/20, onde 80% da base será usada para treinamento e 20% da base validará o modelo. Finalmente, usei um parâmetro de *random\_state* a fim de permitir a replicar a mesma base entre todos modelos testados, evitando assim algum viés individual na validação do modelo.



Por fim, para continuar com a construção dos modelos. Copiei, usando novamente a biblioteca shuttil, as imagens da base de validação (test) e também treino (train), cada uma destas pastas contém ainda outras duas pastas uma para melanoma (mel) e outra para *naeuvs* (nv), conforme a seguinte estrutura:



Dentro desta mesma pasta *dataset\_final*, a subpasta de imagens, em evidência na imagem acima, contém todas as 7988 imagens provenientes do tratamento das bases HAM10000 e MED-NwODE, além do arquivo CSV também gerado durante o tratamento das bases de dados.

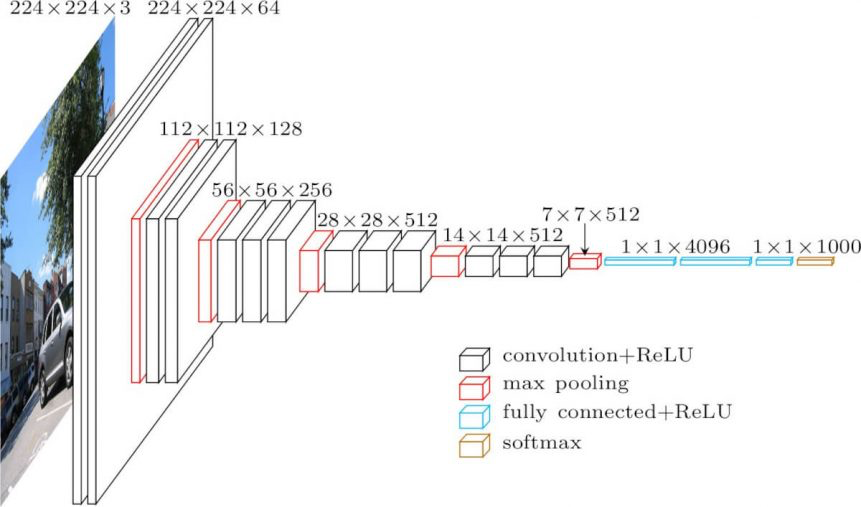
# 5. Criação de Modelos de Machine Learning

Antes de seguir com as explicações e também código utilizados para explicitamente trabalhar com os modelos, vale ressaltar alguns conhecimentos bastante utilizados para os textos seguintes. Dentre eles:  
 **Convolução** é um operador linear que, a partir de duas funções dadas, resulta numa terceira que mede a soma do produto dessas funções ao longo da região subentendida pela superposição delas em função do deslocamento existente entre elas.

Modelos Pré-treinados, eliminam a necessidade de grandes volumes de dados de treinamento, recursos de computação massivos e também amplo conhecimento de Inteligência Artificial (IA), através de modelos pré-treinados é possível acelerar os fluxos de trabalho e desenvolvimento.

Agora que esses conceitos já foram brevemente detalhados, é possível entender um pouco mais dos seguintes modelos de machine learning:

## 5.1 VGG-16

O modelo pré treinado *Very Deep Convolutional Networks for Large-Scale Image Recognition* (em tradução literal: Redes Convolucional Muito Profundas para Reconhecimento de Imagem em Larga Escala), mais conhecido como VGG-16 é um modelos pré-treinados mais populares para classificação de imagem. Introduzido na famosa conferência ILSVRC de 2014, ainda é um dos modelos mais assertivos e modernos no tempos de hoje, cerca de 8 após seu anúncio. Desenvolvido pelo Grupo de Visuais Gráficos da Universidade de Oxford. O VGG-16 bateu o até então padrão *AlexNet* e rapidamente foi adotado por pesquisadores e também por indústrias para tarefas de classificação de imagem. Segue a arquitetura por trás do VGG-16:  


Abaixo um modelo mais intuitivo do layout usado pelo modelo VGG-16:



O modelo é composto pelas seguintes camadas:

* + Camadas de Convolução: 13
  + Camadas de Conjugação (também chamadas de Pool): 5
  + Camadas de Compactação: 3

Explorando em detalhes:

1. Entrada: Imagem com dimensões: altura de 224, largura de 224, 3 camadas
2. Camada de Convulação Conv1:
   1. Conv1-1: 64 filtros;
   2. Conv1-2: 64 filtros e também *Max Pooling;*
   3. Imagem com as seguintes dimensões: 224 x 224.
3. Camada de Convolução Conv2: Filtros aumentados para 128.
   1. Imagem ingerida com 112 x 112 de dimensões
   2. Conv2-1: 128 filtros;
   3. Conv2-2: 128 filtros;
4. Camada de Convolução Conv3: Novamente, filtros dobrados para 256, mais uma camada de convolução
   1. Imagem ingerida com dimensões reduzidas: 56 x 56.
   2. Conv3-1: 256 filtros;
   3. Conv3-2: 256 filtros;
   4. Conv3-3: 256 filtros e também *Max Pooling.*
5. Camada de Convolução Conv4: Semelhante ao Conv3, mas agora com 512 filtros:
   1. Imagem ingerida com dimensões reduzidas: 28 x 28.
   2. Conv4-1: 512 filtros;
   3. Conv4-2: 512 filtros;
   4. Conv4-3: 512 filtros e *Max Pooling.*
6. Camada de Convolução Conv5: Semelhante a Conv4.
   1. Imagem ingerida com dimensões reduzidas: 14 x 14.
   2. Conv5-1: 512 filtros;
   3. Conv5-2: 512 filtros;
   4. Conv5-3: 512 filtros;
   5. Na saída as dimensões da imagem são de 7 x 7. Nesse momento a saída desta camada gera um vetor.
7. Totalmente Conectada/Compactação FC1: 4096 nós, gerando um vetor de features de tamanho 1 x 4096
8. Totalmente Conectada/Compactação FC2:4096 nós, gerando um vetor de features de tamanho 1 x 4096.
9. Totalmente Conectada/Compactação FC3: 4096 nós, gerando 1000 canais para 1000 classes. Então é passado para função de ativação do *Softmax*
10. Camada de Saída

Como pode ser visto, o modelo é sequencial por natureza e faz uso de muitos filtros. Nesse estágio, pequenos filtros de 3 \* 3 são usados para reduzir o número de parâmetros, todas as camadas ocultas usam da função de ativação ReLU. Mesmo assim, o número de parâmetros é da ordem de 138 bilhões, o que torna o processamento todo muito mais devagar em modelos consideravelmente maiores em relação a outros. Durante os testes deste projeto, este modelo levou cerca de 28 minutos para treinar o modelo.

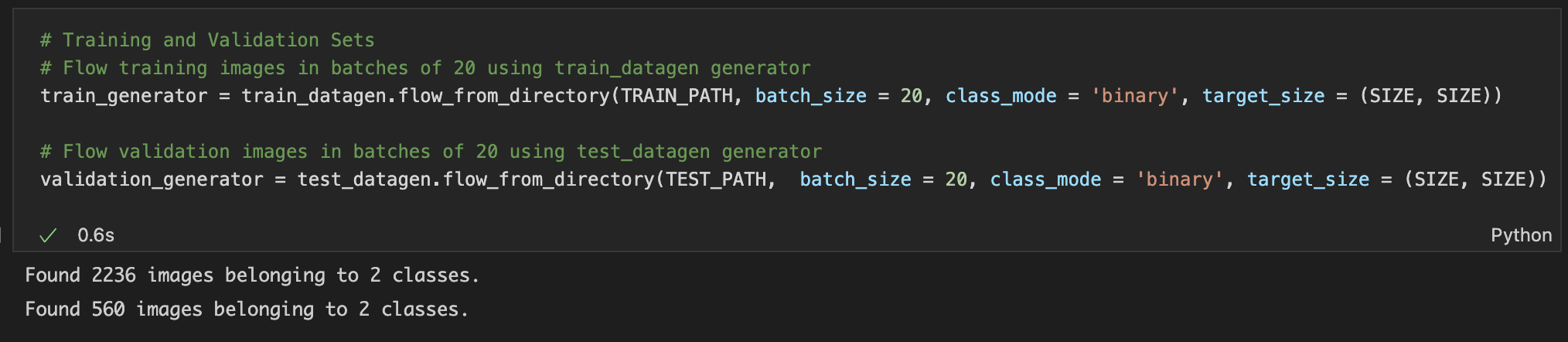
Existem ainda variações ao modelo VGG16, que no geral são apenas aprimoramentos em relação ao mesmo, como por exemplo o VGG19, composto por 19 camadas. Abaixo segue o detalhamento, junto aos códigos aplicando o modelo VGG16:

### 5.1.1 Aumentação de dados:



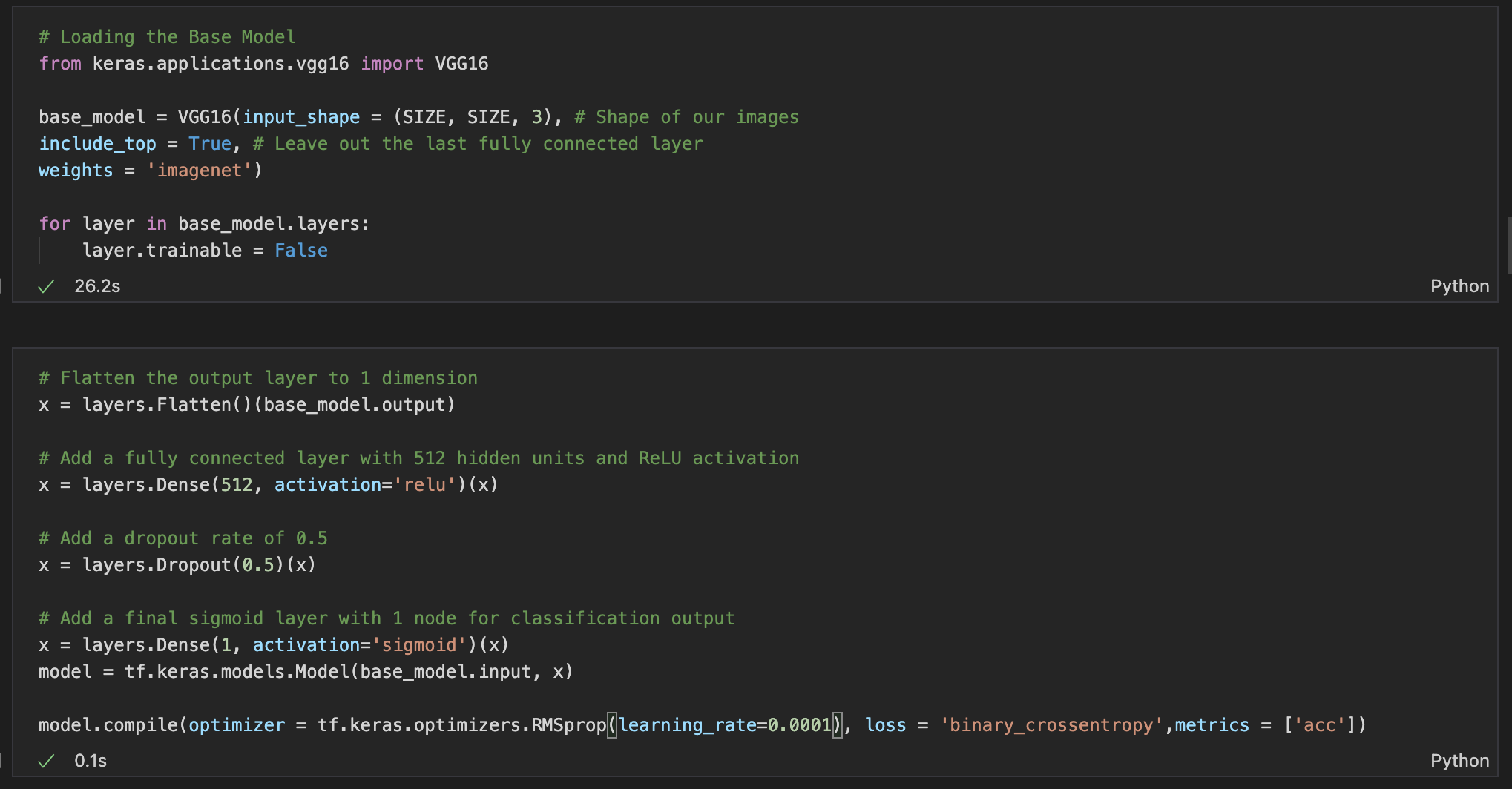
Para aumentar a base de dados foi utilizado o *ImageDataGenerator*, construído pela biblioteca Tensorflow e Keras. É utilizado para gerar lotes de imagens em formato de dados de imagem *tensor* (próprio para modelos de machine learning baseados em Tensorflow), contendo dados aumentados em tempo real. Este é um passo bastante importante pois uma etapa com esta pode aumentar consideravelmente a acurácia do modelo e ao mesmo tempo reduzir possíveis vieses dos dados.

### 5.1.2 Separação de bases de treinamento e validação:

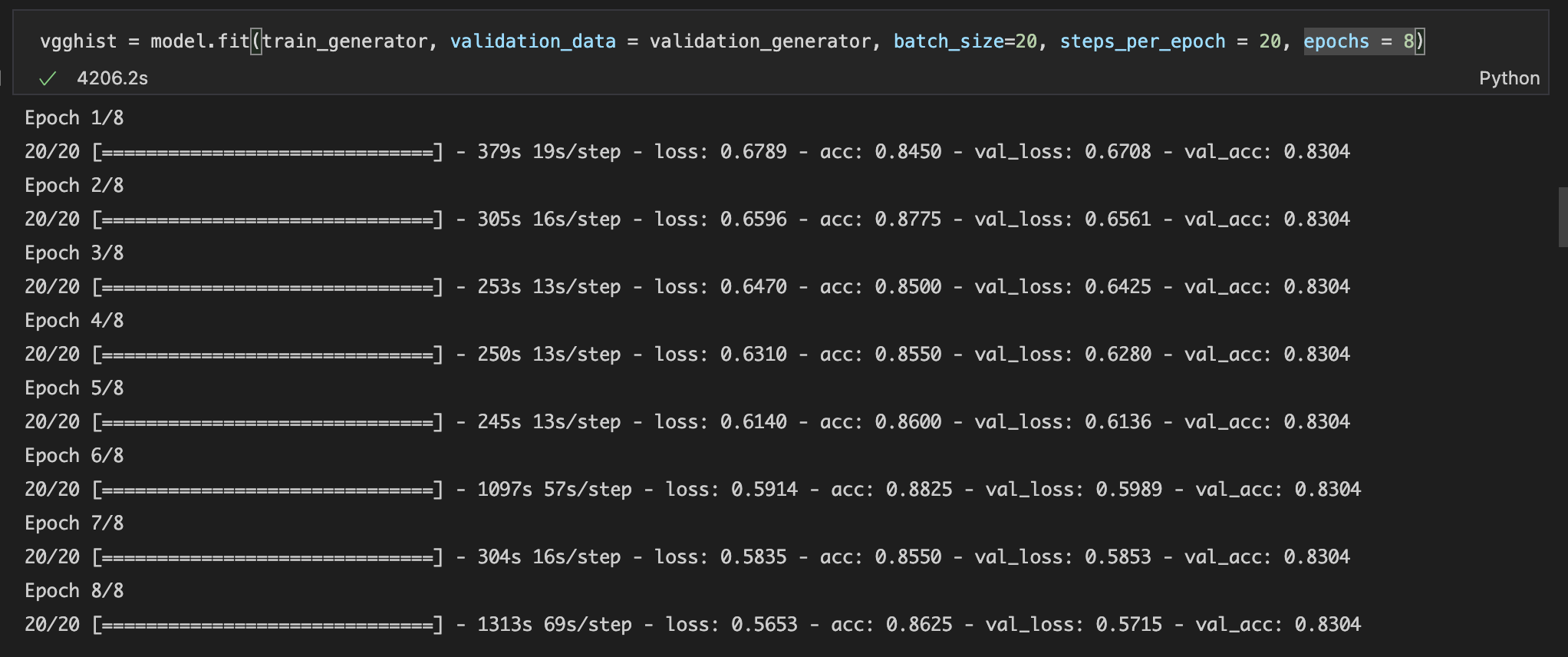


Etapa simples onde separamos as bases de treino e validação, através dos diretórios definidos na camada de Análise e Exploração dos Dados. Foi utilizado a função de *flow\_from\_directory*, classificando as imagens de maneira binária. A base de treino contém 2236 imagens, classificadas entre melanoma ou *naeuvs*. A base de validação, por outro lado, contém 560 imagens, também classificadas em duas classes: melanoma ou *naeuvs*.

### 5.1.3 Carregamento do modelo e compilação:



### 5.1.4 Treinamento:



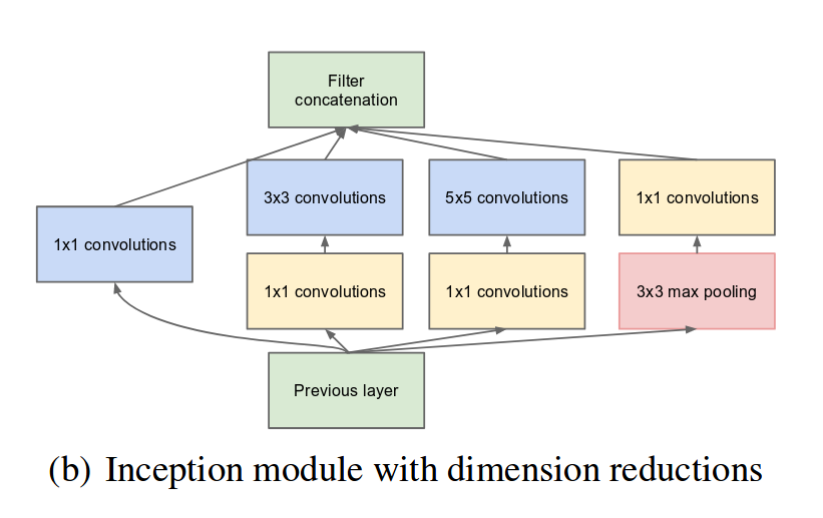
O treinamento deste modelo faz uso da função fit, inteirando por lotes de 20. Após alguns testes locais, ficou claro que 20 passos para cada *epoch* e 8 *epochs.* Foram mais do que suficientes para atingir uma acurácia de 83,04%.

## 5.2 Inception:

*Inception* é um modelo pré-treinado para classificação de imagens, modelos pré-treinados são alternativas menos custosas para treinamento em máquinas locais de pequena capacidade computacional. O modelo pré-treinado Inception é segurado por ninguém menos que o Google, era inicialmente conhecido como *GoogLeNet*, e agora mais recentemente como Inception.

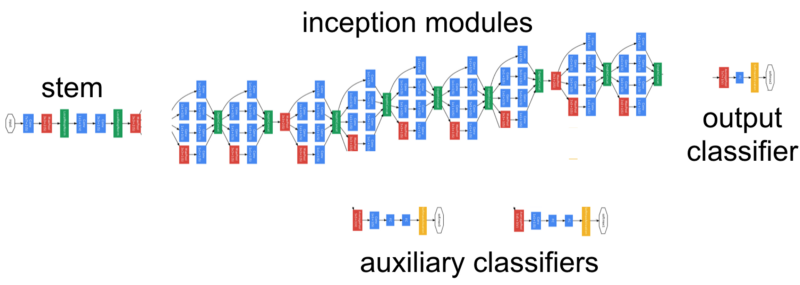
O documento original propôs o modelo *Inceptionv1*, contendo 7 milhões de parâmetros, muito menor que outros modelos predecessores como VGG e *AlexNet*. Adicionando a isso um índice menor de falha, possível ver abaixo no detalhamento do modelo. Além disso, a maior inovação presente no documento original que propôs esse modelo, foi na verdade outro detalhamento do módulo Inception.

Em outras palavras, o módulo Inception apenas executa convolações com diferentes filtros aplicados na entrada dos dados, performando portanto *Max Pooling*, e concatenando o resultado com a próximo módulo Inception. A introdução de 1 para 1 nas operações de convolução reduziu o número de parâmetros drasticamente.



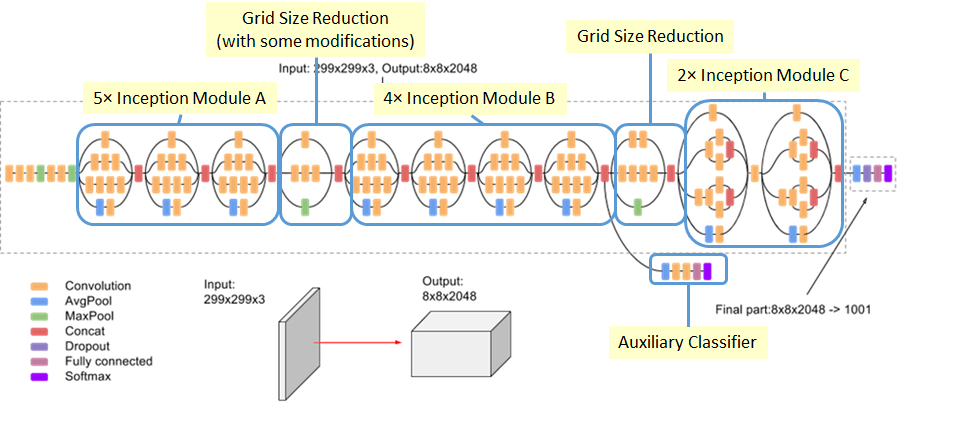
Em tradução literal acima: (b) Módulo Inception com reduções de dimensões.

O modelo pré-treinado *Inceptionv1* é composto de 22 camadas, uma redução também massiva de parâmetros tornou esse modelo dificilmente batível por outros.



Já o modelo *Inceptionv2*, teve ainda um grande aprimoramento em relação ao *Inceptionv1*, o que basicamente aumentou a acurácia e tornou também com o passar do tempo o modelo menos complexo. No mesmo documento que o modelo *Inceptionv2* foi proposto, o autor introduziu ainda o modelo *Inceptionv3* com mais algumas melhorias em relação à versão anterior também recém introduzida. Dentre as maiores melhorias, essas são as de maior destaque:

* Introdução da Normalização por Lote;
* Mais fatorização;
* Otimizador *RMSprop.*



Não é possível explicar em detalhes todas as camadas e detalhes do modelo Inception dentro deste pequeno artigo. Entretanto é possível ver que o número de camadas no modelo *Inceptionv2* é de 42 camadas, o modelo *Inceptionv3* reduziu ainda o índice de erro para apenas 4.2%.

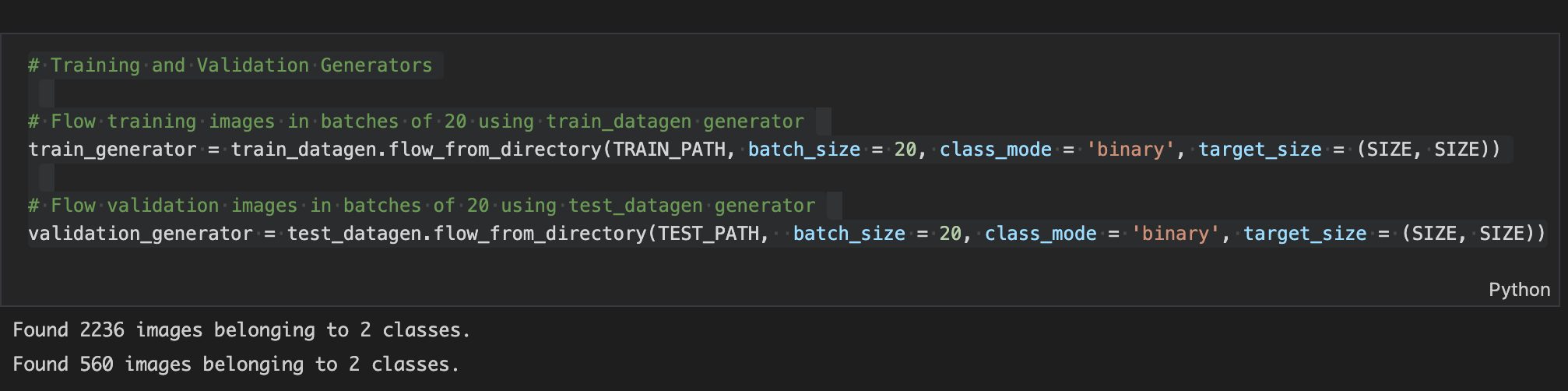
Entrando agora no desenvolvimento do modelo em si, aplicando a nossa base de imagens e contexto de classificação de imagens de lesões de pele. Dentre as etapas que se seguem:

### 5.2.1 Aumentação de dados:



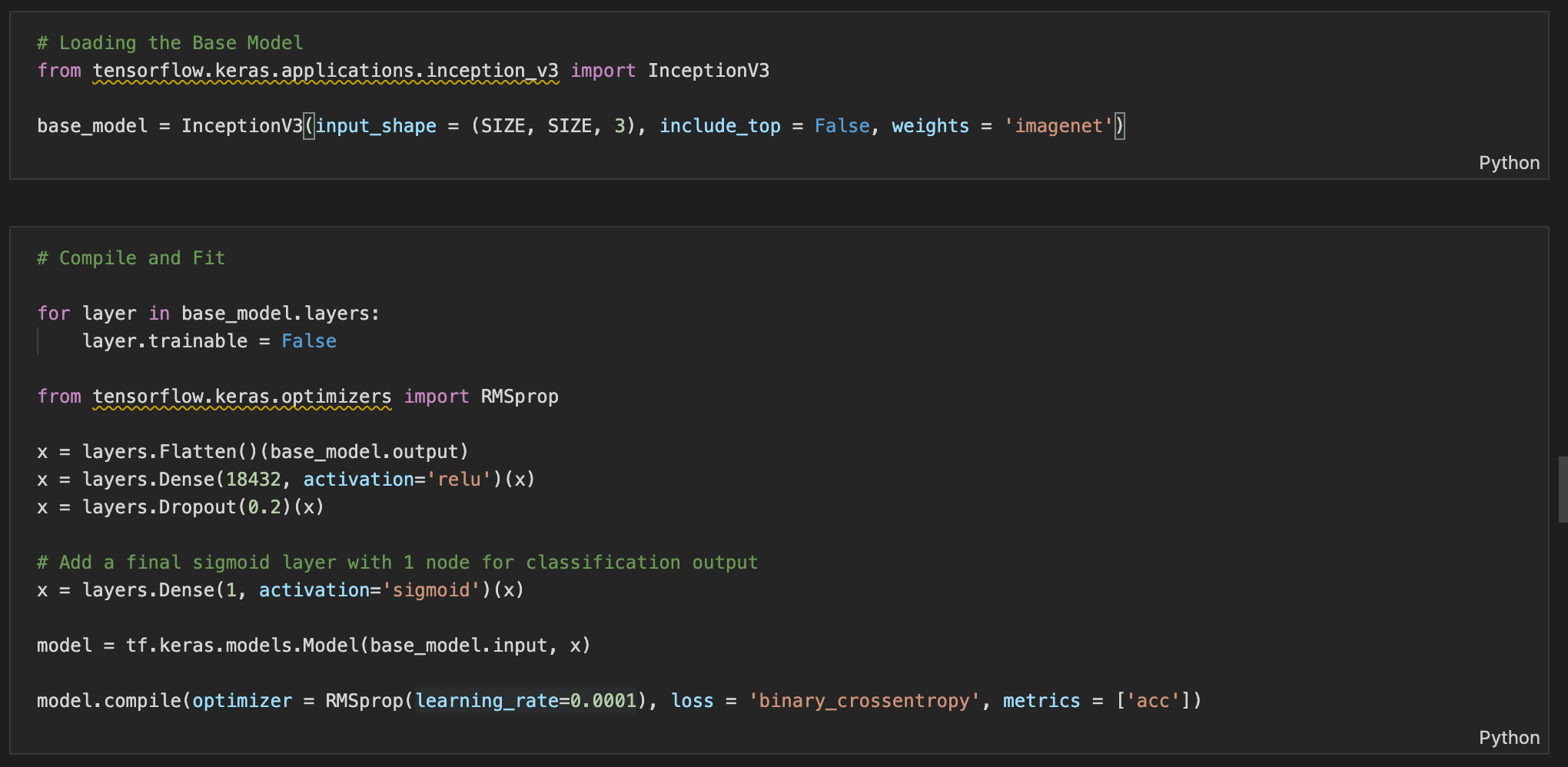
Para aumentar a base de dados foi utilizado o *ImageDataGenerator*, construído pela biblioteca Tensorflow e Keras. É utilizado para gerar lotes de imagens em formato de dados de imagem *tensor* (próprio para modelos de machine learning baseados em Tensorflow), contendo dados aumentados em tempo real. Este é um passo bastante importante pois uma etapa com esta pode aumentar consideravelmente a acurácia do modelo e ao mesmo tempo reduzir possíveis vieses dos dados.

### 5.2.2 Separação de bases de treinamento e validação:



Etapa simples onde separamos as bases de treino e validação, através dos diretórios definidos na camada de Análise e Exploração dos Dados. Foi utilizado a função de *flow\_from\_directory*, classificando as imagens de maneira binária. A base de treino contém 2236 imagens, classificadas entre melanoma ou *naeuvs*. A base de validação, por outro lado, contém 560 imagens, também classificadas em duas classes: melanoma ou *naeuvs*.

### 5.2.3 Carregamento do modelo e compilação:

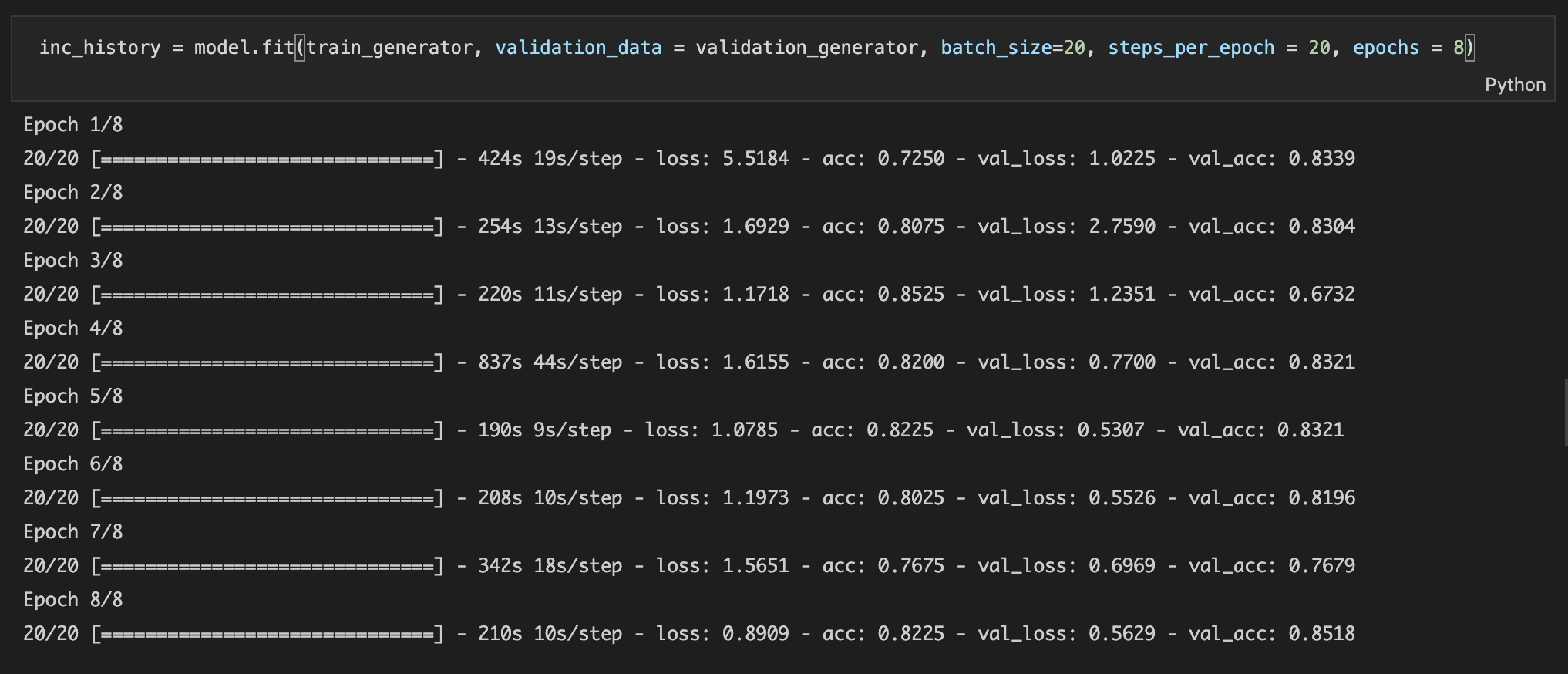


Nesta etapa, é feito o carregamento do modelo *InceptionV3*, mais moderno modelo de *Inception*. Para isso seguimos com o tamanho de imagem de 150 pixels por 150 pixels. Para Otimização deste modelo foi utilizado o mais novo *RMSprop* com índice de aprendizado de 10ˆ-4. Permitindo um leve crescimento de aprendizado na variação de parâmetros, tornando os ajustes ao longo das interações mais refinados. O otimizador RMSprop implementa o algoritmo de *RMSprop*, que tem como base os seguintes conceitos matemáticos:

* Manter a média móvel (descontada) dos gradientes quadráticos.
* Divide o gradiente pela raiz da média.

*RMSprop*, faz uso de momentum plano, ao invés do momentum *Nesterov*. A versão centralizada deste otimizador mantém ainda as médias móveis dos gradientes, e usa essa média para estimar a variação.

### 5.2.4 Treinamento:



O treinamento deste modelo faz uso da função fit, inteirando por lotes de 20. Após alguns testes locais, ficou claro que 20 passos para cada *epoch* e 8 *epochs.* Foram mais do que suficientes para atingir uma acurácia de 85,18%.

## 5.3 Último modelo:

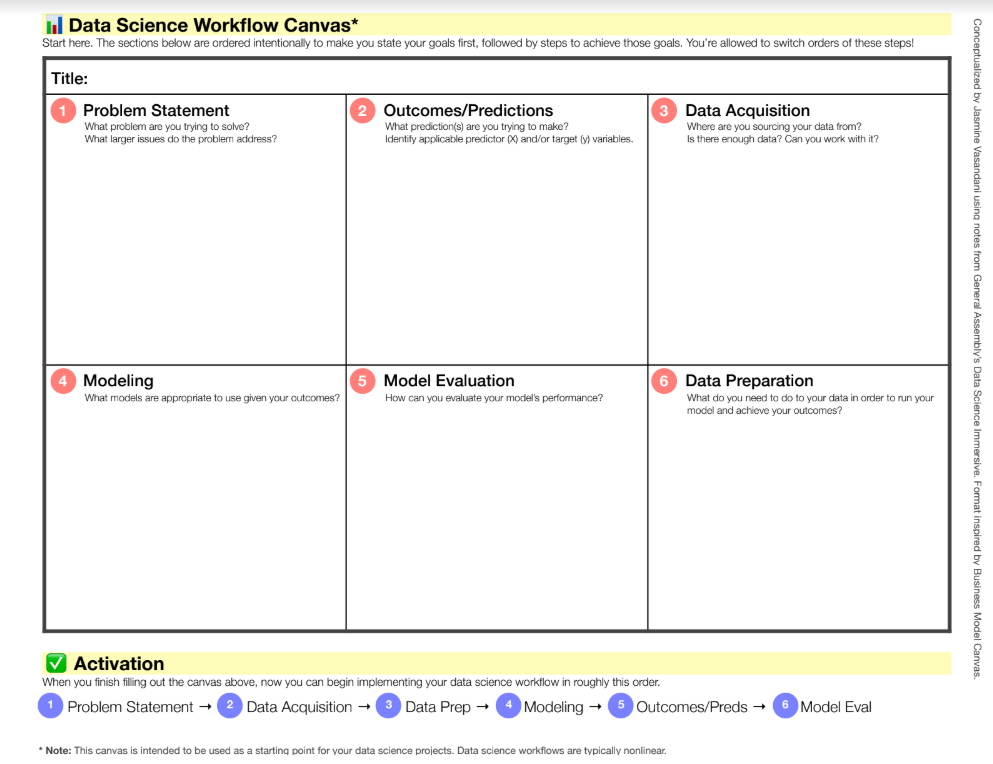
Conforme o documento de instruções para o TCC, essa etapa é obrigatória. Nessa seção você irá descrever as ferramentas e algoritmos utilizados. Se você utilizou o Knime, coloque aqui um print dos seus modelos e a descrição detalhada do workflow de cada modelo. Caso você tenha escrito scripts em Python ou R, por exemplo, coloque aqui apenas os trechos do código que você considera extremamente importantes para entendimento do seu trabalho. Explique as *features* utilizadas, faça a comparação entre diferentes algoritmos/modelos, justifique a escolha por determinado modelo, os parâmetros utilizados, etc. Por fim, salienta-se que embora você possa utilizar o KNIME para testar protótipos do seu modelo de ML, encorajamos você a fazer seus modelos em Python ou R.

# 6. Interpretação dos Resultados

Nessa seção você deve interpretar os resultados obtidos na análise e exploração de dados e também interpretar os resultados da aplicação dos algoritmos de Machine Learning, descobrindo insights importantes para responder o problema proposto.

# 7. Apresentação dos Resultados

Nessa seção você deve apresentar os resultados obtidos. Apresente gráficos, *dashboards*, conte a sua história de forma bastante criativa. Aqui você pode utilizar os modelos de Canvas propostos por Dourard (clique [aqui](https://www.louisdorard.com/machine-learning-canvas)) ou por Vasandani (clique [aqui](https://towardsdatascience.com/a-data-science-workflow-canvas-to-kickstart-your-projects-db62556be4d0)).



# 8. Links

Aqui você deve disponibilizar os links para o vídeo com sua apresentação de 5 minutos e para o repositório contendo os dados utilizados no projeto, scripts criados, etc.

Link para o vídeo: youtube.com/...

Link para o repositório: github.com/...

# 

# 9.REFERÊNCIAS

Um projeto/relatório técnico de Ciência de Dados não requer revisão bibliográfica. Portanto, a inclusão das referências não é obrigatória. No entanto, caso você deseje incluir referências relacionadas às tecnologias ou às metodologias usadas em seu trabalho, relacione-as de acordo com o modelo a seguir.

| [I. Giotis](http://www.cs.rug.nl/~jannis/), N. Molders, S. Land, M. Biehl, M.F. Jonkman and N. Petkov: "MED-NODE: A computer-assisted melanoma diagnosis system using non-dermoscopic images", *Expert Systems with Applications*, 42 (2015), 6578-6585 |
| --- |

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

**APÊNDICE**

**Programação/Scripts**

Cole aqui seus scripts em Python e/ou R.

**Gráficos**

Cole aqui workflows (KNIME), gráficos e figuras que você tenha gerado e não colocou no texto principal.

**Tabelas**

Cole aqui tabelas de dados que você tenha gerado e não colocou no texto principal.