AI Engineer position

Technical assessment of Arthur Vieira

Document designated to guide the assessment of the technical skills for a candidate from different perspectives for the role of AI Engineer within SDG Group’s Data Technologies practice.

*Documento designado para orientar a avaliação das habilidades técnicas de um candidato de diferentes perspectivas para a função de Engenheiro de IA na prática de Tecnologias de Dados do Grupo SDG.*

# AI Engineer role technical assessment for SDG Group’s practice

Aim of this technical assessment is for the candidate **to demonstrate needed skills regarding Python development, data science lifecycle basic concept understanding and surrounding engineering** key aspects of production ready advanced analytics projects.

*O objetivo desta avaliação técnica é que o candidato demonstre as habilidades necessárias em relação ao desenvolvimento em Python, compreensão do conceito básico do ciclo de vida da ciência de dados e aspectos-chave da engenharia ao redor da produção pronta projetos de análise avançada.*

You might not know all components listed below for the deployment of an end-to-end solution, but it is also a key aspect of the proof to test your ability to have a basic understanding of some of those and been able to use them within the context of this challenge in a coherent manner. These are the key aspects to showcase during the oral demonstration:

*Você pode não conhecer todos os componentes listados abaixo para a implantação de uma solução de ponta a ponta, mas também é um aspecto fundamental da prova testar sua capacidade de ter um entendimento básico de alguns deles e poder usá-los dentro o contexto deste desafio de forma coerente. Estes são os principais aspectos a serem apresentados durante a demonstração oral:*

1. If you are not familiar to Docker, is a free accessible container managing technology that will help you deploy most of the rest of the technologies required for this challenge. Being a Windows user, you need to be aware of the version required for [Docker Desktop](https://www.docker.com/products/docker-desktop) to work, which in general is Windows 10 on its version where, from our experience WSL2 backend is mostly recommended. Docker Desktop is also available for Mac users as for Linux users it is more usual to have it directly install by means of their [package managing software](https://docs.docker.com/engine/install/ubuntu/).

*Se você não estiver familiarizado com o Docker, é uma tecnologia de gerenciamento de contêiner de acesso gratuito que o ajudará a implantar a maioria das outras tecnologias necessárias para esse desafio. Sendo um usuário do Windows, você precisa estar ciente da versão necessária para o Docker Desktop funcionar, que em geral é o Windows 10 em sua versão onde, pela nossa experiência, o backend WSL2 é o mais recomendado. O Docker Desktop também está disponível para usuários de Mac, já que para usuários de Linux é mais comum instalá-lo diretamente por meio de seu software de gerenciamento de pacotes.*

1. Use Docker to deploy an **Apache Spark** local instance. You can use this [DockerHub image](https://hub.docker.com/r/bitnami/spark/), following the instruction to enable *two local workers*:

*Use o Docker para implantar uma instância local do Apache Spark. Você pode usar esta imagem do DockerHub, seguindo as instruções para habilitar dois workers locais:*

* 1. Download deployment specification as a Docker Compose template (for example using cURL): curl -LO <https://raw.githubusercontent.com/bitnami/bitnami-docker-spark/master/docker-compose.yml>

*Baixe a especificação de implantação como um modelo do Docker Compose (por exemplo, usando cURL):*

* 1. From the path where the template was downloaded you can just call it to be deployed using the command prompt and following command: *docker-compose up*

*A partir do caminho em que o modelo foi baixado, basta chamá-lo para ser implantado usando o prompt de comando e o seguinte comando: docker-compose up*

* 1. From there, by means of your browser you could just access Apache Spark’s UI interface in the following direction: <http://localhost:8080/>.

*A partir daí, por meio do seu navegador, basta acessar a interface do Apache Spark na seguinte direção:* [*http://localhost:8080/*](http://localhost:8080/)*.*

1. Setup a **HDFS file system** as well by means of docker compose- You can find an example [here](https://dev-listener.medium.com/docker-compose-for-apache-hadoop-hdfs-d949150f745a).

*Configure um sistema de arquivos HDFS também por meio do docker compose-. Você pode encontrar um exemplo* [*aqui*](https://dev-listener.medium.com/docker-compose-for-apache-hadoop-hdfs-d949150f745a)*.*

1. From [this](https://databricks.com/blog/2020/01/27/time-series-forecasting-prophet-spark.html) **Databricks’ blog article**, download the notebook with the example code [here](https://pages.databricks.com/rs/094-YMS-629/images/Fine-Grained-Time-Series-Forecasting.html) with the objective of transforming the original code in a production grade PySpark application (usually invoked using spark-submit command). This is a key aspect given that one of key conditions for an AI Engineer role is not only being able to understand such monolithic code but also to refactor its different parts in separated modules so that each part can be used separately in an automated model creation pipeline where model gets trained and scored in different points of the model lifecycle and the model created and used in those phases gets usually stored as a separated structure to the code generating it (a Pickle file often). As you will see the notebook starts by creating a data table containing relevant features so that it can afterwards be used in both phases, training and scoring of the model, taking into account each model corresponds to a different group. This distributed action is done by means of Spark, so the data gets spread on different worker nodes applying an user Defined Function (UDF) to each subset. This last part is the one we require you to re-structure and industrialize eliminating all experimentation and visualization parts more relevant during exploratory phases. While building the PySpark application you will face different problems related to the self-service nature of the Databricks environment, related to environment specifics or dependency management that you will need to solve.

*A partir* [*deste*](https://databricks.com/blog/2020/01/27/time-series-forecasting-prophet-spark.html)***artigo do blog do Databricks****, baixe o notebook com o código de exemplo* [*aqui*](https://pages.databricks.com/rs/094-YMS-629/images/Fine-Grained-Time-Series-Forecasting.html) *com o objetivo de transformar o código original em um aplicativo PySpark de nível de produção (geralmente invocado usando o comando spark-submit). Este é um aspecto fundamental, pois uma das principais condições para uma função de engenheiro de IA não é apenas entender esse código monolítico, mas também refatorar suas diferentes partes em módulos separados para que cada parte possa ser usada separadamente em um pipeline automatizado de criação de modelos onde o modelo é treinado e pontuado em diferentes pontos do ciclo de vida do modelo e o modelo criado e usado nessas fases geralmente é armazenado como uma estrutura separada do código que o gera (geralmente um arquivo Pickle). Como você verá, o notebook começa com a criação de uma tabela de dados contendo recursos relevantes para que depois possa ser usada nas duas fases, treinamento e pontuação do modelo, levando em consideração que cada modelo corresponde a um grupo diferente. Essa ação distribuída é feita por meio do Spark, para que os dados sejam distribuídos em diferentes nós do trabalhador aplicando uma Função Definida pelo Usuário (UDF) a cada subconjunto. Esta última parte é a que exigimos que você reestruture e industrialize eliminando todas as partes de experimentação e visualização mais relevantes durante as fases exploratórias. Ao construir o aplicativo PySpark, você enfrentará diferentes problemas relacionados à natureza de autoatendimento do ambiente Databricks, relacionados a especificidades do ambiente ou gerenciamento de dependências que você precisará resolver.*

1. You will need to also **deploy an Apache Airflow** instance (you might use Docker for that as well). Apache Airflow is a process orchestrator that differs from other solutions as the orchestration processes are defined using Python scripts. You could use [following guidelines](https://airflow.apache.org/docs/apache-airflow/stable/start/docker.html) to deploy Airflow using docker compose as before.

*Você também precisará* ***implantar uma instância do Apache Airflow*** *(você também pode usar o Docker para isso). O Apache Airflow é um orquestrador de processos que se diferencia de outras soluções, pois os processos de orquestração são definidos usando scripts Python. Você pode usar as* [*seguintes diretrizes*](https://airflow.apache.org/docs/apache-airflow/stable/start/docker.html) *para implantar o Airflow usando o docker compose como antes.*

1. Once you have Airflow running and you have familiarized with it, the way dags are created and executed we **would like you to create your own DAG** where the typical time series pipeline Will be implemented. By using your Apache Spark deployment, your HDFS file system and your Apache Airflow instance using appropriate operator for the work ([SparkSubmitOperator](http://apache-airflow-docs.s3-website.eu-central-1.amazonaws.com/docs/apache-airflow/latest/_api/airflow/providers/apache/spark/operators/spark_submit/index.html), for example). Your DAG will execute following process:

*Depois de ter o Airflow em execução e você se familiarizar com ele, a forma como os dags são criados e executados, gostaríamos que você criasse seu próprio DAG onde o pipeline de série temporal típico será implementado. Usando sua implantação do Apache Spark, seu sistema de arquivos HDFS e sua instância do Apache Airflow usando o operador apropriado para o trabalho (*[*SparkSubmitOperator*](https://airflow.readthedocs.io/en/latest/_api/airflow/providers/apache/spark/operators/spark_submit/index.html#airflow.providers.apache.spark.operators.spark_submit.SparkSubmitOperator)*, por exemplo). Seu DAG executará o seguinte processo:*

* 1. Upload required CSV files to the HDFS

*Carregue os arquivos CSV necessários para o HDFS.*

* 1. Execute the ML steps (train, test, score…) that you have created on step 4

*Execute as etapas de ML (treinar, testar, pontuar…) que você criou na etapa 4.*

* 1. Download the models stored inside HDFS (some at least) to your local environment.

*Baixe os modelos armazenados dentro do HDFS (pelo menos alguns) para o seu ambiente local.*

The aim of this exercise is no other than showing you understand the usual lifecycle when comes to machine learning model deployment in productive environments. Batch predictions were also periodic or upon request model training phases are performed are part of the routine in many organizations today. Obviously, an enterprise solution would require this to happen automatically, with a set of tools that allows both monitoring and managing the whole lifecycle of these processes.

*O objetivo deste exercício é mostrar que você entende o ciclo de vida normal quando se trata de implantação de modelos de aprendizado de máquina em ambientes produtivos. As previsões de lotes também eram periódicas ou mediante solicitação, as fases de treinamento do modelo são realizadas fazem parte da rotina de muitas organizações atualmente. Obviamente, uma solução corporativa exigiria que isso acontecesse automaticamente, com um conjunto de ferramentas que permitissem tanto monitorar quanto gerenciar todo o ciclo de vida desses processos.*

Once you have achieved this, we invite you to add some extra features to your solution (you could also use Docker to ease their deployment):

*Depois de conseguir isso, convidamos você a adicionar alguns recursos extras à sua solução (você também pode usar o Docker para facilitar a implantação):*

* MLflow local instance to track experiments performed, store trained models or serve them to scoring steps as part of an MLOps pipeline.

*Instância local do MLflow para rastrear experimentos realizados, armazenar modelos treinados ou encaminhá-los para etapas de pontuação como parte de um pipeline de MLOps.*

* You could also use Flask or FastAPI libraries in order to expose one of those trained models as an online request service. You can encapsulate all this within a Docker image to ease its deployment and scalability afterwards.

*Você também pode usar as bibliotecas Flask ou FastAPI para expor um desses modelos treinados como um serviço de solicitação online. Você pode encapsular tudo isso em uma imagem do Docker para facilitar sua implantação e escalabilidade posteriormente.*

* Also, it would be valued any addition to the solution regarding monitoring (tools such as Prometheus or Grafana, for example).

*Além disso, seria valorizado qualquer acréscimo à solução referente ao monitoramento (ferramentas como Prometheus ou Grafana, por exemplo).*

Of course, all these extra points are additional to the initial requirements and can be tackled according to your expertise or availability. Apart from the technical implementation, it also will be valued if you have conducted your own research on tools, solutions or architectures that could be presented on a presentation to support your vision on the solution delivered apart from the points above stated. If you feel comfortable on adding visualizations or automating infrastructure deployment, feel free to do so, it will be very much valued.

*Claro, todos esses pontos extras são adicionais aos requisitos iniciais e podem ser resolvidos de acordo com sua experiência ou disponibilidade. Além da implementação técnica, também será valorizado se você tiver realizado sua própria pesquisa sobre ferramentas, soluções ou arquiteturas que possam ser apresentadas em uma apresentação para apoiar sua visão sobre a solução entregue, além dos pontos mencionados acima. Se você se sentir confortável em adicionar visualizações ou automatizar a implantação de infraestrutura, sinta-se à vontade para fazê-lo, será muito valorizado.*

If you have doubts at any time during the development of the assessment, feel free to contact and ask for clarifications. It also applies to any issue that may come up requiring to move the date. Feel free to reach out to Joan (joan.marchan@sdggroup.com - 669285434) or Jesús (jesusvicente.garcia@sdggroup.com - 669727656).

*Caso tenha dúvidas em qualquer momento durante o desenvolvimento da avaliação, fique à vontade para entrar em contato e solicitar esclarecimentos. Também se aplica a qualquer problema que possa surgir exigindo a mudança da data. Sinta-se à vontade para entrar em contato com Joan (joan.marchan@sdggroup.com - 669285434) ou Jesús (jesusvicente.garcia@sdggroup.com - 669727656).*

Thanks so much in advance for the effort and hope to talk soon. Please do inform us that you have received the technical assessment and that you accept the challenge.

*Muito obrigado antecipadamente pelo esforço e espero falar em breve. Por favor, informe-nos que recebeu a avaliação técnica e que aceita o desafio.*