Data Science technical test

Context

The prediction of a drug molecule properties plays an important role in the drug design process. The molecule properties are the cause of failure for 60% of all drugs in the clinical phases. A multi parameters optimization using machine learning methods can be used to choose an optimized molecule to be subjected to more extensive studies and to avoid any clinical phase failure.

Objective

The objective of this exercise is to develop a deep learning model to predict one or more basic properties of a molecule.

> *🙈 The context is around the chemical molecule, it is going to be complicated 🙈 ...*
> *Don't worry and keep going the exercise is simple 😉*

Dataset

There are 2 datasets for this exercise (the following datasets contains public data):

1. `dataset_single.csv` contains **4999** rows and **three** columns:
    - `smiles` : a string representation for a molecule
    - `mol_id` : Unique id for the molecule
    - `P1` : binary property to predict
2. `dataset_multi.csv` is an extension of the first dataset with multi properties. `P1, P2,...P9` represent the different properties to predict

> *You can split the dataset as you want to create the training/validation/test datasets*

Models

You have to deliver two deep learning models, Model1 and Model2 for the first dataset `dataset_single.csv`. The Model3 is optional but it can be used as bonus point for the candidate.

Model1

This model takes the **extracted features of a molecule** as input and predict the `P1` property. Use the function `fingerprint_features` of the `feature_extractor.py` module to extract the features from a molecule smile:

```
fingerprint_features('Cc1cccc(N2CCN(C(=O)C34CC5CC(CC(C5)C3)C4)CC2)c1C')
```

This will return a feature vector that you can use as input for your model

Model2

This model takes the smile string character as input and predict the `P1` property.

Model3

Extension of Model1 or Model2 to predict the `P1, P2,...P9` properties of the `dataset_multi.csv` dataset

Application

To deliver:

Main

A `main.py` module that allows the user to

- train the model
- evaluate the model
- predict the property `P1` for any given smile

Flask api

A simple flask api to serve you model with just one route `/predict` that you can use to send your molecule smile and get the prediction

Packaging

Your application must be installable using a `setup.py`. Use the entry point of your `setup.py` to make the following commands available after installing your package

```
servier train <your arguments>
```

or

```
servier evaluate  <your arguments>
```

or

```
servier predict  <your arguments>
```

Docker

Package your application in a docker to facilitate the deployment of your model

To build a docker image you can use the command (your dockerfile must be in the current directory):

```
 docker  build . -t servier
```

> *Don't include your data in your docker image you have to figure out how to use your dataset in your docker*

Readme file

This md file must have the documentation about your application, models, packaging, setup and dockerization

Environment

> *:warning: This environement works under Linux or Mac operating system*

The `feature_extractor.py` module use `rdkit` library. This library is available on `conda`. Run the following command to install the package and prepare your environment:

```
wget --quiet https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
Miniconda3-latest-Linux-x86_64.sh -b -p ~/miniconda
export PATH=~/miniconda/bin:$PATH
conda update -n base conda
conda create -y --name servier python=3.6
conda activate servier
conda install -c conda-forge rdkit
```

Once your environment `servier` is created you can use `pip` to install any package you want in this environment and you can use it in your IDE.

> *You can use the same code in your dockerfile to prepare the environment ;).*

Git

Use git to version your code and push it to any public repository and send us the link

> *If you want to share any big files like models, please store them in any public drive and send us the public link*

Evaluation

You we be evaluated on:

- Quality and structure of your code
- Models architecture
- Git commits quality
- Quality of the application (main and flask api)
- Data preparation and preprocessing
- Dockerization
- Documentation

> *The accuracy of the model is important but it will not be the main criteria to evaluate your models*

🙏 Good Luck 🙏

> *⚠️⚠️⚠️⚠️ PLEASE DON'T SHARE THIS TEST ... EVEN WITH YOUR RECRUTER ⚠️⚠️⚠️⚠️*