**Predicting Sex using machine learning**

Among the deliverables are the following files: a jupyter notebook (end-to-end-technical-test-data-scientist.ipynb), the trained model (sex\_predictor.pk), the Python3 script (sex\_predictor.py) and the requirements to run the project (requirements.txt).

The notebook contains the whole project from loading data, EDA, and the code to generate the expected output (newsample\_PREDICTIONS\_Igor\_Barbosa\_Lima.csv).

The trained model is in format .pk, so we need import pickle to save and reload it.

The requirements include: Pandas, Numpy, Matplotlib, Seaborn, Jupyter and Scikit-learn.

I started the project launching a Jupyter notebook, imported the needed libraries, and load the ‘test\_data\_CANDIDATE.cvs’ file. After that, I check the shape of the dataset and saw that it is made of 288 rows and 18 columns.

**Exploratory Data Analysis**

As Pandas create a new column named “index”, I used the .drop() function to exclude it.

Then, I checked how many male and female samples we have in the data frame, and I found the first problem in the dataset. The target (sex) column must contain only two options: female (F) or male (M). However, there are entries written in lowercase and others in uppercase. In this way, the computer recognizes 4 classes instead of 2: F, M, f and m. To fix this I converted all letters in column “sex” to uppercase. Thus, the target column started to have only two categories (F = female, M = male). However, the data is imbalanced, we have much more female records then male`s.

The next step was check if there were any missing values. The dataset has 16 missing values in column “chol” and 143 missing values in column “slope”. To fix this I decided to fulfill “chol” missing values with the median of data and the “slope” with the value 1, since it is the most frequent value and when we interpret the meaning, it is found that 1 represents no alterations in the slope of ecg results after physical activities (a kind of neutral option).

With the dataset complete, I checked the data types. As expected, the target column is categorical, so, I converted it to integers (F = 0, M = 1).

Now that the data is all numeric, I plotted some graphs to get some insights about the data:

apparently, the younger someone is, the higher their max heart rate (dots are higher on the left of the graph) and the older someone is, there are more green dots. But this may be because there are more dots all together on the center of the graph (more cases between 50 and 60 years old). Also, looks like that female individuals have more stained vessels during the examination than male individuals. This can be an important feature for model learning.

To find significative correlation between variables I plotted a correlation matrix, from which, is possible observe that 'cp' does not significantly correlate with the other variables.

After have some information I started modeling:

1. Remove the target column (sex) from the data frame,
2. Set random seed as 42 to achieve reproducibility,
3. Split the data set in train and test data sets (if there was more data it would be possible to create a validation set too) and set the test size to 0.2 (80% of data to train on and 20% to test on),
4. Model choice: by following the Scikit-Learn map I decide to test the Logistic Regression, K-Nearest Neighbors, and Random Forest,
5. Score models and compare: KNN performs worse (0.53),
6. Tune KMN by hand: the KNN model achieved accuracy after tuning similar to the other two without tuning (0.68), so it was the first to be inconsiderate,
7. Tunning Random Forest (0.67) and Logistic regression by RandomizedSearchCV also not improved the models enough (0.67), so, I tuned the models hyperparameters again with GridSearchCV. This method improved slightly the performance of Logistic Regression (0.68). As the three models have similar accuracy, I decided to use Logistic Regression because it is a classic algorithm for binary classification problems. As the accuracy of the models was not rising, I used other metrics more complexes to evaluate the model:

* ROC curve and AUC score - plot\_roc\_curve()
* Confusion matrix - confusion\_matrix()
* Classification report - classification\_report()
* Precision - precision\_score()
* Recall - recall\_score()
* F1-score - f1\_score()

With the classification report generated it was possible to find out why the model was not improving. The data imbalance mentioned above (more female then male) makes the model really good to correctly label individuals as female while dos not perform good to identify male individuals. In a real-world situation, I would decide to collect more data. About the feature importance, some are negative and some are positive. The larger the value (bigger bar), more the feature contributes to model’s decision. If the value is negative, it means there's a negative correlation. And vice versa for positive values. It means the model has found a pattern which reflects the data. Looking at these figures and this specific dataset, it seems that the number of major vessels colored by flourosopy (ca) and the number of arms (nar) contribute more intensely to the model's decision-making in classifying a patient as female or male.

**#Readme**

Steps to setup and run the test

Choose the directory where the test will be run (I recommend desktop). I use the package management system conda to run Python projects. Check documentation in < <https://docs.conda.io/en/latest/>>.

1. Open the terminal.app;
2. Choose the directory with the command **cd Desktop;**
3. Create a folder to host files and the environment with the command **mkdir test**. This will create a folder named “test” on your desktop;
4. Save the files ‘end-to-end-technical-test-data-scientist.ipynb’, ‘sex\_predictor.pk’, ‘sex\_predictor.py’, ‘requirements.txt’, and the input data named as ‘newsample.csv’ inside the folder ‘test’;
5. Use the command **cd test** to change the directory to the folder “test”;
6. Create a virtual environment for your project. Use the venv module that comes with Python to create a virtual environment. For example, to create a virtual environment called env in the root directory of your project, run the following command: **python3 -m venv env;**
7. Activate the virtual environment running the following command: for macOS or linux run: **source env/bin/activate.** If you’re using Winddowns run **env\Scripts\activate.bat**;
8. Use pip to install the packages from the requirements.txt file. Run the following command: **pip install -r requirements.txt;**
9. This will install all the required packages and their dependencies in the virtual environment.

Note: Make sure that you are using the correct version of pip and Python for your project. It's a good practice to create and use a virtual environment for each Python project to isolate its dependencies and avoid conflicts with other packages;

1. Use the command **python3 sex\_predictor.py** to run the model. It will generate a file named ‘newsample\_PREDICTIONS\_Igor\_Barbosa\_Lima.csv’ containing a column named ‘Sex’ with the predictions made by the Logistic Regression model.

\*Obs: Make sure that your input file has no missing values.