

Assignment: Fullstack/ML Engineer (Python/DriverlessAI)

Target group	Difficulty
FullStack/ML Eng	Medium

Create web based application, where you can upload dataset, train a linear regression model, and visualize the model performance.

1. Create web server using e.g. Tornado (<https://www.tornadoweb.org>) with minimum of following handlers:
 - a. Async upload handler
 - b. Async training handler
 - c. (Async) Model metrics visualize handler
2. Implement own Linear Regression (https://en.wikipedia.org/wiki/Linear_regression) estimator from scratch using arbitrary estimation method:
 - a. **Avoid** using 3rd party libraries like **Scikit-learn** and similar, which already contain this type of Model
 - b. **Avoid** using 3rd party **libraries which implement e.g. gradient descent**, for optimizing the model
 - c. **You can use Numpy** and similar, to operate on matrices.
3. Implement ROC curve metric (https://en.wikipedia.org/wiki/Receiver_operating_characteristic):
 - a. Again, **avoid** using 3rd party libraries which already implement it, but **you can use Numpy** and similar, to operate on matrices
4. Create Web based UI handling:
 - a. Uploading of datasets
 - b. Model training
 - c. Model listing
 - d. Visualize interactive model performance chart (ROC curve) e.g. using D3 library, Vega, etc.
 - i. It's sufficient that the ROC has hover effect, displaying tooltip containing the threshold value and corresponding Confusion Matrix

For a sake of simplicity, you can assume that all datasets will have only numerical (continuous) columns. The target (predicted column) will be binary, which means only 0/1 values.

The assignment will be evaluated on brevity, effectiveness, elegance and quality of both code and structure.

There is no strict requirement on technologies used, however it's expected, that the frontend part will have at least basic styling and structure.

Usage of Typescript and/or some framework is appreciated.

Example data:

Titanic survival data [titanic.csv](#):

ID - Row number

PassengerID - ID of passenger

Survived - 0/1 whether passenger survived

Pclass - Ticket class (1 = 1st, 2 = 2nd, 3 = 3rd)

Sex - Gender (0 - female, 1 - male)

Age - Age

SibSp - # of siblings / spouses aboard the Titanic

Parch - # of parents / children aboard the Titanic

Fare - Passenger fare