niologic | STRATEGIC DATA INSIGHTS

Machine Learning Assessment

Thank you for expressing an interest in Niologic and for your time earlier.
As mentioned, the next stage of the recruiting process is to complete the below coding assignment.

After handing this back, we will have two engineers review your code and get back to you with feedback (either way). If your code shows potential and meets our expectations, we will invite you into our offices to carry out a pair programming exercise (extension problem on your provided assignment), among some other interviews.

Enclosed is the Seattle Fire Department problem. We are seeking evidence of clean, simple, tested code with evidence of good design and ML, OOP (or functional) skills. We would like to see that your code has been developed using TDD (desirable) and expect a good level of knowledge of the chosen language and framework.

Note:

- For the solution, we request that you use **Python**.
- The problem below requires some kind of input. You are free to implement any mechanism for feeding input into your solution(e.g hard coded data within the unit test).
- The application must run
- You should provide sufficient evidence that your solution is complete by, as a minimum, indicating that it would work correctly against the supplied test data.
- Upload your code to github.com into a public repository and share the link with us. Everything for understanding your code should be included (comments, documentation as markdown etc.)

Rules:

1. Use a library for machine learning like Scikit.Learn or Tensorflow (or Keras or TFX) or CatBoost with a Scikit.Learn pipeline (e.g. you can use Google AI platform with a test account or https://colab.research.google.com/ to run your code,
2. Provide us with a ML pipeline of getting data in, processing data (scoring) and retrieving back data,
3. Use a Jupyter notebook to create your ML code. Please subdivide your code into training, preprocessing, testing, model and write the code from notebook to separate files (e.g. train.py via writefile),
4. We need to be able to run and build your code ourselves, so please submit your code as a collection of source code and supporting files, without any compiled code.
5. Please include a brief explanation of your design and assumptions, along with your code, as well as detailed instructions to run your application (in your README as markdown).
6. We assess a number of things including the design aspect of your solution and your object oriented (or functional) programming skills. While these are small problems, we expect you to submit what you believe is production-quality code; code that you'd be able to run, maintain, and evolve. You should not gold plate your solution, however we are looking for something more than a bare-bones algorithm (KISS & YARN).
7. We want our hiring process to be fair, and for everyone to start from the same place. To enable this, we request that you do not share or publish these problems. Do not copy ready-made solutions from the web.

As a general rule, we allow 5 days from the date that you receive these instructions to submit your code, but you may request more time if needed.

Problem : Prediction of Call Volume for Seattle Fire Department

Dispatching emergency calls in a city is challenging: There are large variations over a year, saison and especially time of day. Someone has to decide how many dispatchers are on duty for every day. Hence, we are looking for a model, which can predict the call volume for each day of the year and each hour of the day.

Please go to data.seattle.gov and download the dataset of 911 calls: https://data.seattle.gov/Public-Safety/Seattle-Real-Time-Fire-911-Calls/kzjm-xkqj - Then use the years before the current year to predict the call volume for the current year. This method is called backtesting: You may not train the model with overlapping time series. If your model does not perform well for the whole year, try to reduce the prediction to the last month and use previous month as training data.

Hints:
In order to ensure a small enough scope for this challenge, here are some hints;

- Good results can be achieved by Gradient Boosting or Boosted Decision Trees
- Features like weather data, time of day, time of year, saison are of importance, borrow
- You don't need to differentiate between type of calls (fire, hazmat etc.) but could try.
- Feature engineering using SQL, especially Google Bigquery can be a time-saver (don't use the example dataset from Bigquery since it is too small.)
- Training with 5 years timeframe can already be sufficient

Input:

- The input of your model should take a vector with at least the date/time

Output:

- The output should contain the call volume (count) of 911 calls predicted for this time period.
- Please provide a backtesting plot comparing the actual (measured) call volume vs. predicted call volume for a month and a week.
- Please think about measures to judge the quality and certainty of predictions and provide them as well.