Data Science - Take home test

This test is to understand your hands on data science skills in detail. The test consists of the following tasks -

- Develop a machine learning model for the given problem statement and the dataset
- Implement an algorithm to get feature contributions at record level from the given pseudo-code
- Create a presentation to walk us through the details of all the steps you took and also the choices you made.

Please note that the goal of this test is to understand your approach. The performance of the model itself is not important and is not something we will judge you on. So, if there are some things which could improve the model performance but may take a long time to do, you can just list them down in the presentation as 'things to try' along with the reasoning why you think they would help improve the model.

Problem Statement

On Poshmark, the price for an item is set by the user who is listing the item. Some users who are listing for the first time, struggle to set an appropriate price for their item. To help such users, we want to develop a model which suggests an appropriate listing price when the item is being listed. The inputs to such a model will be the attributes of the listing that the user enters while creating the listing.

To develop this model, we have provided a sample of sold listings from the year 2019. The dataset is a CSV file. A brief description of the fields is as follows -

Column	Type	Description
ID	Identifier	Unique identifier of the record. Also indicates the temporal order of when the item was sold i.e. records with higher value of ID were sold later in time compared to records with lower values of ID e.g. ID=100 was sold later than ID=50

Column	Type	Description
attr1	Nominal categorical	Attribute 1
attr2	Nominal categorical	Attribute 2
attr3	Nominal categorical	Attribute 3
attr4	Nominal categorical	Attribute 4
attr5	Nominal categorical	Attribute 5
attr6	Numeric	Attribute 6
title	Text	Title of the listing
sold_price	Numeric	The price at which the item was sold on Poshmark

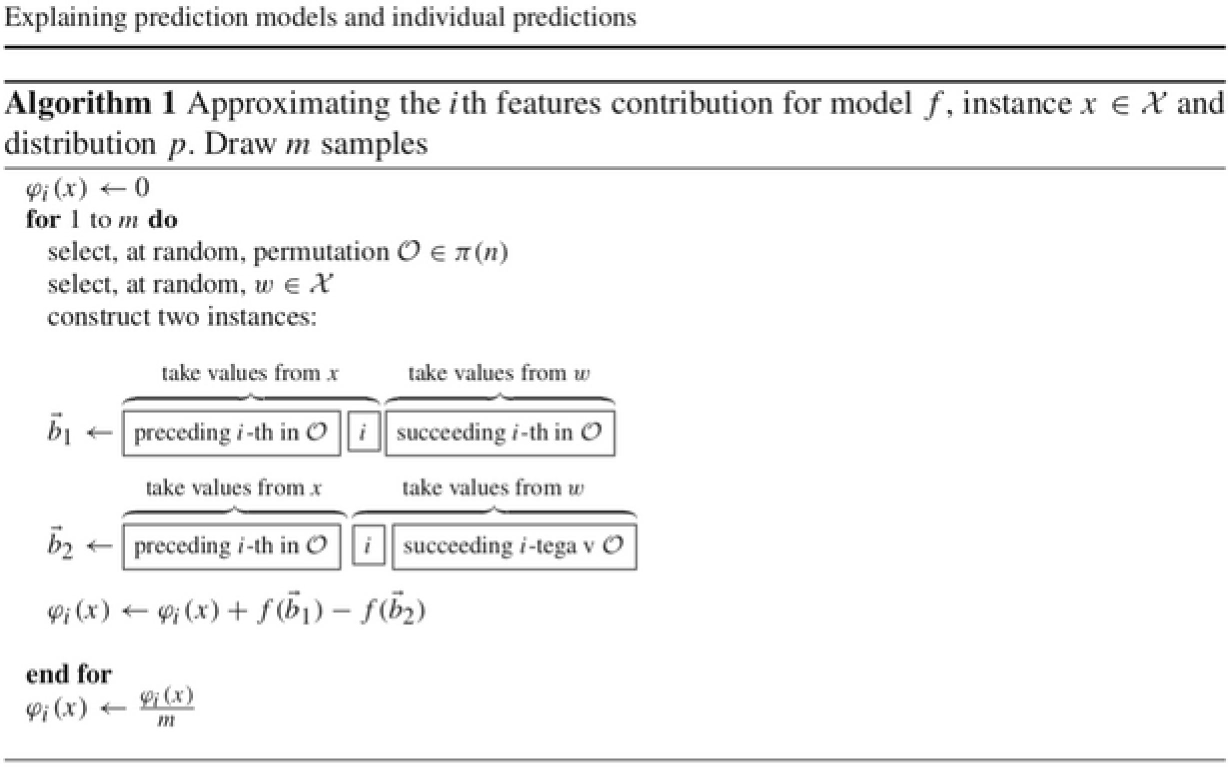
For some of the columns, the data has been encoded so as to not reveal the actual values but that should not affect your approach or the performance of the model you create.

Given this dataset and the business problem, we want to develop a model which is able to estimate the sold price as accurately as possible using the other attributes as inputs (excluding the identifier). In terms of judging how good the model is, one of the most important criterion is to have nearly equally good performance across five sold price ranges/buckets defined by - **0-50, 50-100, 100-500, 500-1000 and 1000+**. The choice of modeling technique, loss function and evaluation metric is yours. However, please keep in mind that a broader audience (business, product etc.) should be able to understand and relate to the evaluation metrics you choose.

Please create a model and share all your work in a Jupyter notebook which should be runnable by us. The preferred language to develop the model is **Python**. Please specify the details of the Python environment you used to develop the model.

Once you develop a model, you have to implement a method to get instance level feature contributions given a model and a data instance (record). The pseudocode of the method is provided in the image below. Create a function with appropriate inputs and outputs implementing the algorithm as in the pseudocode. Make sure that your implementation is scalable i.e. is reasonably fast, uses vectorised operations if required and can be used to compute the feature importance for multiple records at a time.

In the pseudocode, n is the number of features that the model uses as inputs, f denotes the trained model, x denotes a single data instance i.e. a record, $\mathcal{O}$ is an ordered permutation of the feature indices $\{1, 2, ..., n\}$ and $\varphi_i(x)$ is the contribution of ith feature for the predicted value of data instance x.



Explaining prediction models and individual predictions

Algorithm 1 Approximating the ith features contribution for model f, instance $x \in \mathcal{X}$ and distribution p. Draw m samples

```
$\varphi_i(x) \leftarrow 0$
**for** 1 to $m$ **do**
  select, at random, permutation $\mathcal{O} \in \pi(n)$
  select, at random, $w \in \mathcal{X}$
  construct two instances:
  $\vec{b}_1 \leftarrow \overbrace{\boxed{\text{preceding } i\text{-th in } \mathcal{O}}\ \boxed{i}}^{\text{take values from } x} \overbrace{\boxed{\text{succeeding } i\text{-th in } \mathcal{O}}}^{\text{take values from } w}$
  $\vec{b}_2 \leftarrow \overbrace{\boxed{\text{preceding } i\text{-th in } \mathcal{O}}}^{\text{take values from } x}\ \boxed{i}\ \overbrace{\boxed{\text{succeeding } i\text{-tega v } \mathcal{O}}}^{\text{take values from } w}$
  $\varphi_i(x) \leftarrow \varphi_i(x) + f(\vec{b}_1) - f(\vec{b}_2)$
**end for**
$\varphi_i(x) \leftarrow \frac{\varphi_i(x)}{m}$
```

Using the above function, compute the feature contributions for the model you created for records with ID in {1, 2, 3, ..., 100}.

Finally, create a presentation to walk us through the details of all the steps you took and also the choices you made while building the model. Your presentation should address the following questions -

- Data import - did you face any issues and if yes, how did you deal with them?
- Modeling dataset (before any train-validation split) - how did you create the modeling dataset? Share any exploratory analysis you did to understand this dataset.
- Feature engineering - Did you do any feature engineering? If yes, for each feature you created, mention the reasoning why it was created.
- Did you do any treatment e.g. missing value, outliers etc.?
- Explain your train-validation-test and/or cross-validation split method
- What techniques do you want to try and why? These are the choices you make based on your understanding of the data even before trying any of the techniques.

- If you tried multiple approaches, how did you choose the final model?
- How will you monitor such a model when deployed to production? What will you log, how frequently and what all alerts will you configure to make sure that the model keeps running well in production and any issues surface as quickly as possible?
- How often will you retrain the model? How would you decide that and how will the retraining work (e.g. data period used, model versioning, maintaining performance levels etc.)
- If the product team comes back saying the performance is not good enough, what will be your response (backed by data)?
- What is the biggest risk for this model when in production? Do you think the model will perform as expected?
- Would there be cases where the model may not be able to predict when in production? If yes, in what scenario/s and how do you estimate the number of cases where model won't be able to predict at all?
- What else would you try to improve the model if you had more time?