Machine Learning Engineer Nanodegree

Capstone Proposal

Yingying Hu
September 29th, 2020

Proposal

Domain Background

"How well is a product likely to sell?" is a crucial question for every retailer. Before the coming of eCommerce, brick-and-mortar businesses can only rely on intuition and experience when forecasting customer's demand on a product. Nowadays, more and more people choose to shop online because of the convenience and product varieties. At the same time, business owners can gain more customer data by tracking their activity on the shopping website. Analyzing and understanding what data suggests product sales performance will help businesses to optimize their stocks. Furthermore, a machine learning algorithm can bring it to the next level by predicting product sales and make decisions on product stock for business owners.

Problem Statement

The goal of this project is to create a ML model to predict the product sales. Tasks including:

1. Explore and preprocess the “Sales of summer clothes in E-commerce Wish” dataset on Kaggle. Wish is a popular American online e-commerce platform that facilitates transactions between sellers and buyers.
2. Select important features for predicting sales for products
3. Choose an ML model and train the model
4. Check the model accuracy by using Mean Absolute Error (MAE) and Root Mean Square Error (RMSE)

Datasets and Inputs

The "Sales of summer clothes in E-commerce Wish" dataset is public on Kaggle. It contains three tables: "summer-products-with-rating-and-performance_2020-08", "unique-categories.csv" and "unique-categories.sorted-by-count".The tables are from crawling the result page after searching "summer" on Wish, and they are collected in August 2020.

The "rating and performance" table has 43 columns. It contains information about summer clothes products and their sellers, such as the product id, retail price, ratings, seller's reputation, etc. These usually drive customer's decision on purchasing a product, so they can be relevant features on predicting a product sale.

The two "unique-categories" tables extract values found in the "rating and performance" table under the "tags" column.

All tables are in csv files. Data cleaning will be needed since there are some duplicated entries and some missing values. Some columns, like product_id and product_url will not be able to use by the model, so they will be removed.

Solution Statement

In the project, a final model will be chosen between several experimental ML models, such as RandomForest, XGBoost, and Neural Network model. Models are built and trained on selected features from the Wish dataset after data preprocessing and feature engineering. The model will be tuned on a validation set. Finally, it will be tested on a test set and its accuracy will be calculated using the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

Benchmark Model

The benchmark model is a linear regression with combined L1 and L2 priors as regularizer. This model is shown in a public notebook shared by a Kaggle user. The model's MAE is 2045.58

Evaluation Metrics

Since the goal is to predict product sales, the model's label is numerical. The model's evaluation metrics will be Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE and RMSE are two common metrics used to measure accuracy for continuous variables. Since the benchmark model uses the MAE, it will be used for comparison purpose.

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

Project Design

This project will be worked on a Jupyter notebook environment. The workflow will be:

1. Set up a git repository and a notebook instance on the AWS SageMaker platform
2. Download the data files from Kaggle and upload to a notebook instance
3. Explore, visualize, and clean the table, including handling missing values and duplicated entries
4. Feature engineering, such as check the correlation on numerical variables and perform one-hot encoding on the categorical features
5. Split the data into train, validation, and test set
6. Choose 2-3 base models, such as Random Forest, XGBoost, and Neural Networks.
7. Training the selected base models on the train set, and compare them and the benchmark model on the validation set using RMSE
8. Select a model and do hyperparameter tunning
9. Check the final model accuracy on the test set and get the MAE and RMSE