بسمه تعالی

دانشکده مهندسی کامپیوتر

طراحی سیستم‌های یادگیری ماشین

بهار ۱۴۰۲

علی زارع‌زاده

فاز اول پروژه | زمان تحویل: ۶ اردیبهشت

1 Overview

Our goal in this document is to evaluate the project grades and raise the important values by reviewing the important metrics that matter to MLOps projects. Make your project reports and documentation to the point and Logically integrated, there is no additional bonus on long and fancy reports.

2 ML Metrics

2.1 Data preprocessing

2.1.1 Data cleansing

- removing or correcting records that have corrupted or invalid values from raw data, and removing records that are missing a large number of columns.
- Remove no longer useful and unnecessary features from the model
- Use features that generalize the prediction well

2.1.2 Instances selection and partitioning

- selecting data points from the input dataset to create training, evaluation (validation), and test sets. This process includes techniques for repeatable random sampling, minority class oversampling, and stratified partitioning.
- Sampling methods (if necessary)
  - Simple Random Sampling
  - Stratified Sampling
  - Weighted Sampling
  - Importance Sampling
  - Reservoir Sampling
- Scale and normalize data after splitting
- Handling Class Imbalance (if necessary)
  - Evaluation Metrics
  - Resampling

2.1.3 Feature tuning

- improving the quality of a feature for ML, which includes scaling and normalizing numeric values, imputing missing values, clipping outliers, and adjusting values that have skewed distributions.

2.1.4 Feature transformation

- converting a numeric feature to a categorical feature (through bucketization), and converting categorical features to a numeric representation (through one-hot encoding, learning with counts, sparse feature embeddings, etc.). Some models work only with numeric or categorical features, while others can handle mixed-type features. Even when models handle both types, they can benefit from different representations (numeric and categorical) of the same feature.

2.1.5 Feature extraction

- reducing the number of features by creating lower-dimension, more powerful data representations using techniques such as PCA, embedding extraction, and hashing.

2.1.6 Feature selection

- selecting a subset of the input features for training the model, and ignoring the irrelevant or redundant ones, using filter or wrapper methods. Feature selection can also involve simply dropping features if the features are missing a large number of values.

2.1.7 Feature construction

- creating new features by using typical techniques, such as polynomial expansion (by using univariate mathematical functions) or feature crossing (to capture feature interactions). Features can also be constructed by using business logic from the domain of the ML use case.

2.1.8 For text documents

- stemming and lemmatization, TF-IDF calculation, and n-gram extraction, embedding lookup.

2.1.9 For images

- clipping, resizing, cropping, Gaussian blur, and canary filters.

2.1.10 For all types of data (including text and images)

- transfer learning, which treats all-but-last layers of the fully trained model as a feature engineering step.

3 Ops Metrics

- Containerization
- CI/CD and automation workflow
- Using container orchestration (Optional)
  - Swarm
  - Kubernetes

- Model Store
- Feature Store (Optional)
- Monitoring

4 The deployment of ML Project Flow

As a first step of the project, it is necessary to store your structured data in a data warehouse and then start preprocessing it using ETL (Extract, Transform and Load). For this purpose you can use Spark as an analytical operating system and Airflow as your pipeline orchestrator. In the data preprocessing step within the points mentioned in the ML metrics part; The following points should be analyzed and -if possible- represented with interpretable diagrams.

- Data Cleansing
  - Analyze features to find no longer useful features: features with a high rate of nulls, with noise
- Feature Tuning
  - Statistical feature analysis such as
    - Correlation between features with heatmaps
    - Analyze features scales
    - Analyze features importance
      - Gini importance for trees
      - Permutation feature importance
    - Analyze features generalization
      - Analyze with bias-variance tradeoff
    - Analyze features outliers
      - Z-score
      - dbscan
  - Label Analysis
    - Analyze data miss-balance

After Analyzing steps, suitable actions should be applied if needed due to dataset and model properties.