5450 Project Techniques Checklist

Datasets: Open Food Fact

Link: <https://www.kaggle.com/datasets/openfoodfacts/world-food-facts>

Data acquisition -> cleaning -> wrangling -> integration -> feature extraction -> models -> evaluation

(Note: This is a concept list for 5450. We don’t need all. Just enter “v” on selected methods, or “x” on unwanted methods, or add comments)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name&Date | Xiaoyun 20241109 |  |  |  |  |
|  |  |  |  |  |  |
| ***Data formats:*** |  |  |  |  |  |
| Numerical independent data | v |  |  |  |  |
| Categorical independent data | v |  |  |  |  |
| Time series data | ??? |  |  |  |  |
| Text data | v |  |  |  |  |
|  |  |  |  |  |  |
| ***Techniques:*** |  |  |  |  |  |
| *Data acquisition:* |  |  |  |  |  |
| Web crawling? |  |  |  |  |  |
| Info retrieval: gather dataset through API? | x |  |  |  |  |
| *Structured, clean, integrated data:* |  |  |  |  |  |
| Pandas | v |  |  |  |  |
| EDA: Correlation heatmap, corr | v |  |  |  |  |
|  |  |  |  |  |  |
| Store in RDMS or NoSQL? (tables? Arrays? Nested? ) | Rough RDMS |  |  |  |  |
| Storage/updates/queries database |  |  |  |  |  |
| Split hierarchical data when not 1:1 |  |  |  |  |  |
| PandaSQL? SQLite? | v |  |  |  |  |
| Transactions? Concurrency control? | ??? |  |  |  |  |
|  |  |  |  |  |  |
| Knowledge Representation, ER Diagrams | v |  |  |  |  |
| knowledge graph |  |  |  |  |  |
| Normal forms (1st, 2nd, 3rd, Boyce-Codd) at least should be 1st for tables | v? |  |  |  |  |
| Coreference/entity resolution | v |  |  |  |  |
| Record linking |  |  |  |  |  |
| Data validation (rules/master list) | v |  |  |  |  |
| NLP:  Remove Unwanted Characters -> Expand Contractions & Lowercase | short texts |  |  |  |  |
| Time series: resampling (downsampling/upsampling) |  |  |  |  |  |
|  |  |  |  |  |  |
| *Optimization for big data:* |  |  |  |  |  |
| Optimizing relational algebra: push down early, indexing, sharding | v |  |  |  |  |
| Optimizing join orders | v |  |  |  |  |
| Algorithmic techniques: index, buffers/blocks | v |  |  |  |  |
|  |  |  |  |  |  |
| *Extract features:* |  |  |  |  |  |
| Dask? |  |  |  |  |  |
| Spark? (minimize repartition: groupby&JOIN; persist()) | v |  |  |  |  |
| AWS cloud? |  |  |  |  |  |
|  |  |  |  |  |  |
| Visualizations | v |  |  |  |  |
| Vis Time series: grouping; binning; rolling statistics |  |  |  |  |  |
|  |  |  |  |  |  |
| One-hot encoding | v |  |  |  |  |
| Text: word embedding | v |  |  |  |  |
| Text: Parts of Speech |  |  |  |  |  |
| Text: identify named entities |  |  |  |  |  |
| Text: entity resolution, co-reference resolution, relation extraction |  |  |  |  |  |
|  |  |  |  |  |  |
| Scaling | v |  |  |  |  |
| Encoding for feeding to NN | v |  |  |  |  |
|  |  |  |  |  |  |
| Subset selection / forward selection / backward selection / PCA / t-SNE before supervised ML | v |  |  |  |  |
| Before classification: clustering | v |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| *Descriptive/Inferential model:* |  |  |  |  |  |
| Parametric models (linear regression, logistic regression, …) better for inference | v |  |  |  |  |
| Hypothesis testing (test stats? )  Bonferroni Correction |  |  |  |  |  |
|  |  |  |  |  |  |
| Time series: Checking for Stationarity: Augmented Dickey-Fuller (ADF); remove trends using moving averages (especially  EWMA); Deseasonalizing by Decomposition |  |  |  |  |  |
| Time series: Forecasting with ARIMA |  |  |  |  |  |
|  |  |  |  |  |  |
| Train/validation/test splitting | v |  |  |  |  |
| PyTorch or Keras? | PyTorch |  |  |  |  |
|  |  |  |  |  |  |
| Classification: SVM | v |  |  |  |  |
| Classification: Multinomial Naïve Bayes | v |  |  |  |  |
| Classification: Random Forest | v |  |  |  |  |
| Classification: K-Nearest Neighbors | v |  |  |  |  |
|  |  |  |  |  |  |
| Hold-out validation set (if large sample) | v |  |  |  |  |
| K-fold CV (if small sample) |  |  |  |  |  |
| Gradient descent | ??? |  |  |  |  |
|  |  |  |  |  |  |
| Fully-connected NN | v |  |  |  |  |
| CNN | v |  |  |  |  |
| RNN (will not add marks) |  |  |  |  |  |
| Tune hyperparameters: batch size, number of layers, activation functions, neuron numbers, optimizers, learning rate, regularization, kernel size (CNN), # of filters (CNN), padding (CNN), pooling layer parameters (CNN), network architecture | v |  |  |  |  |
| Grid-search to tune | v |  |  |  |  |
|  |  |  |  |  |  |
| Text: sentiment analysis | x |  |  |  |  |
| Text: Latent Dirichlet Allocation | v |  |  |  |  |
| ***Evaluation metrics:*** |  |  |  |  |  |
| Underfitting or overfitting | v |  |  |  |  |
| MSE | v |  |  |  |  |
| RSS | v |  |  |  |  |
| accuracy | v |  |  |  |  |
| precision | v |  |  |  |  |
| recall | v |  |  |  |  |
| F1 | v |  |  |  |  |
| Confusion matrix (in heatmap) | v |  |  |  |  |
| ROC/AUC | v |  |  |  |  |
| Visualization of all models’ perform |  |  |  |  |  |
| ***Future improvements:*** |  |  |  |  |  |
|  |  |  |  |  |  |
| Mod13 concepts |  |  |  |  |  |
| ***Ethical discussions*** |  |  |  |  |  |
|  |  |  |  |  |  |