CM2606 Data Engineering

Machine Learning with Big Data

Week 10 | Piumi Nanayakkara

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY

Learning Outcomes

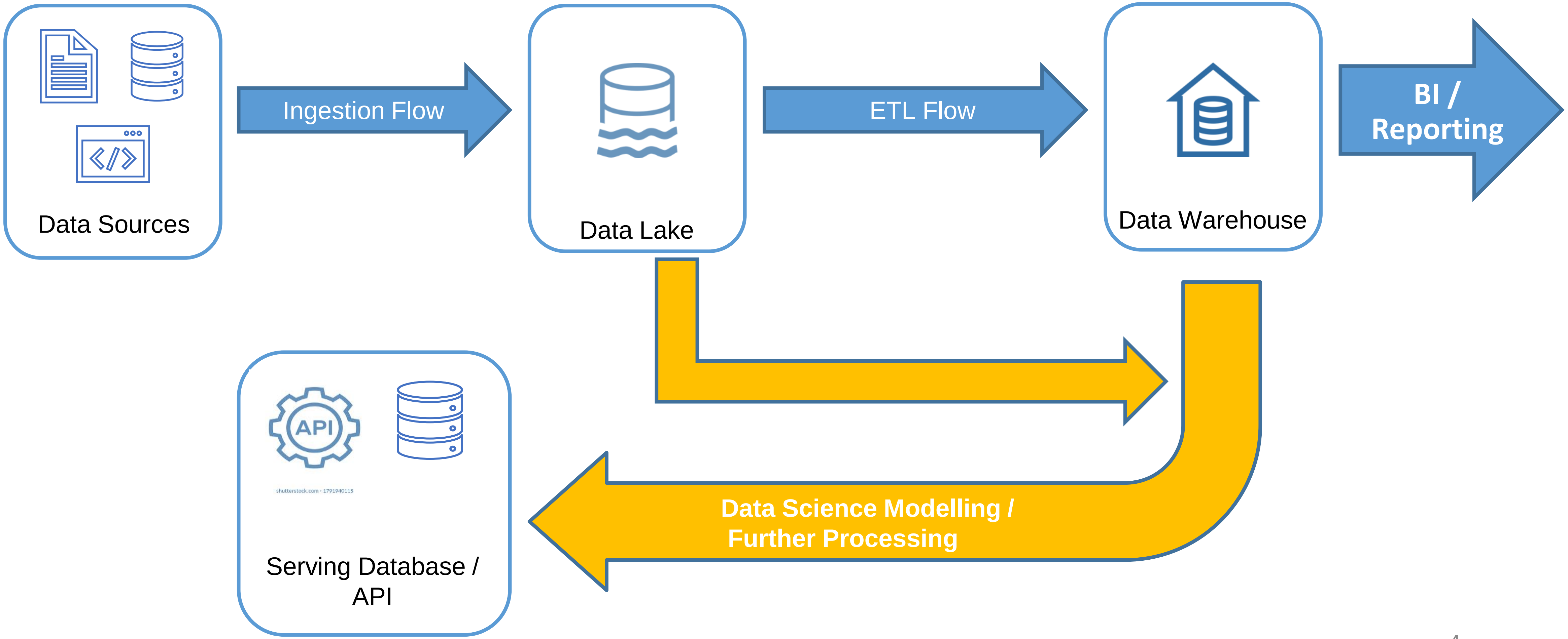
- Covers LO2 and LO4 for Module

- On completion of this lecture, students are expected to be able to:
  - Understand and explain how a machine learning project will be executed within a production setup.
  - Adapt Industry Practices and knowledge in designing production scale machine learning pipelines.

Content

- Challenges for Machine Learning with Big Data
- Machine Learning Project Life Cycle
- Training Stage
    - Model Exporting
    - Orchestration
    - Spark ML / ML Lib
- Predicting Stage
    - Model Serving
    - Model Monitoring

Data Pipeline: Common Usage



Machine Learning in Production

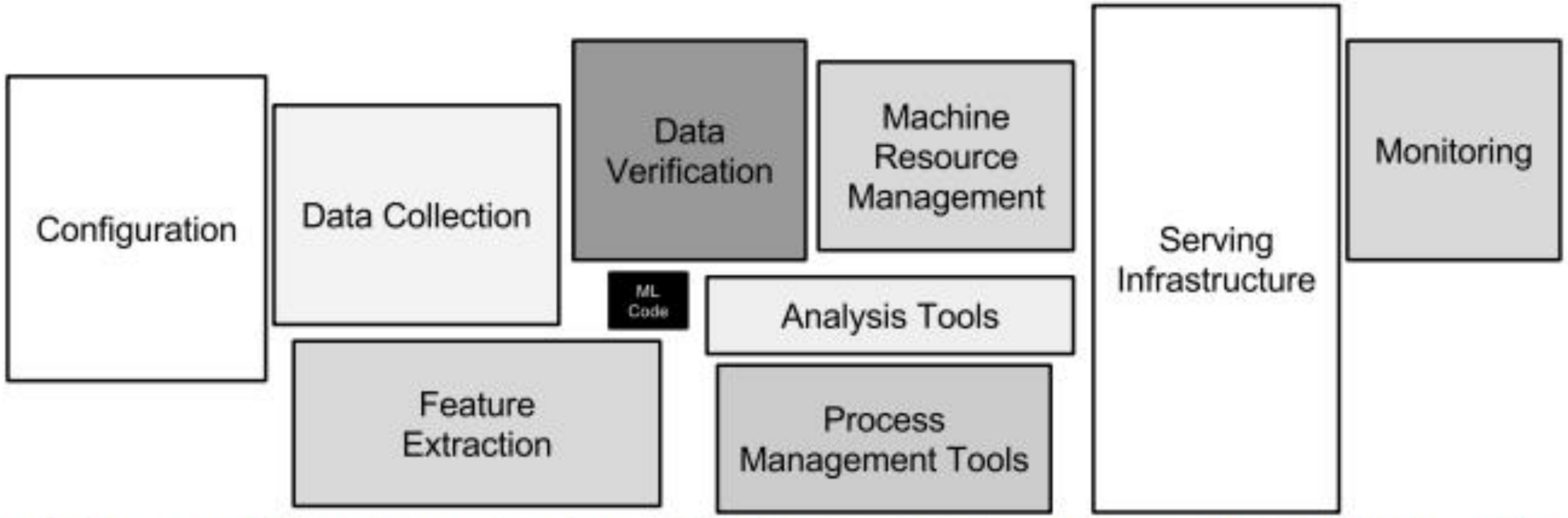


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

[Source: Hidden Technical Debt in Machine Learning Systems]

Importance of ML in Production

- Pipeline needs to be automated since models can be outdated quickly.
- Data sources and types change rapidly
- Need to make use of all available data for a better model performance
- Need for Realtime / Near-Realtime Processing
- Rollback or Failover Mechanisms
- From Notebooks to modularized, versioned coding

Challenges for ML with Big Data

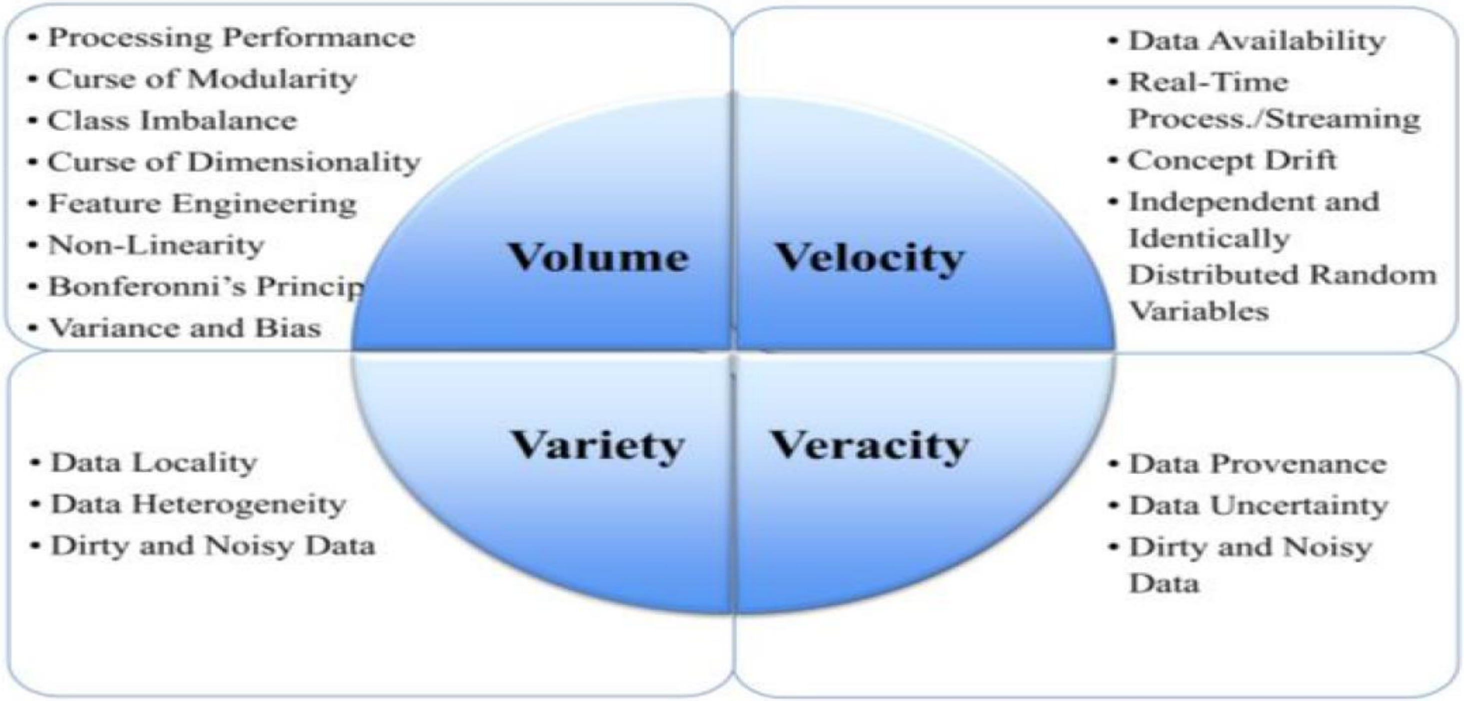


FIGURE 1. Big Data characteristics with associated challenges.

[Source: Machine Learning With Big Data: Challenges and Approaches]

Challenges to ML due to Volume

- W.r.t. Machine learning volume can be defined by either,
  - Number of data points / records
  - Number of features / attributes

- Performance Issues in Processing
  - SVM algorithm has training time complexity of $O(m^3)$ and a space complexity of $O(m^2)$
  - Time complexity of logistic regression is $O(mn^2 + n^3)$
    - where m is no. of records in dataset and n is the number of features

Challenges to ML due to Volume

- Curse of Dimensionality
  - As the number of features increases, the performance and accuracy of machine learning algorithms degrades.
- Feature Engineering
  - This is the most time-consuming preprocessing tasks in machine learning
  - As the dataset grows, time and effort to be invested at this step further increases

Challenges to ML due to Variety

- With data coming in from heterogeneous sources it will contain various types of measurement errors, outliers, and missing values
  - Which would require additional effort in data pre-processing and cleaning
- In addition, there could also be semantic heterogeneity which refers to differences in meanings and interpretation of the same data across different sources
  - Definition of a “Year” in Financial data and HR data

Challenges to ML due to Velocity

- Velocity here refers not only to the speed at which data are generated, but also the rate at which they must be analyzed.

- Models will be outdated quickly, Need to be re-trained

- Need for real-time predictions

Challenges to ML due to Veracity

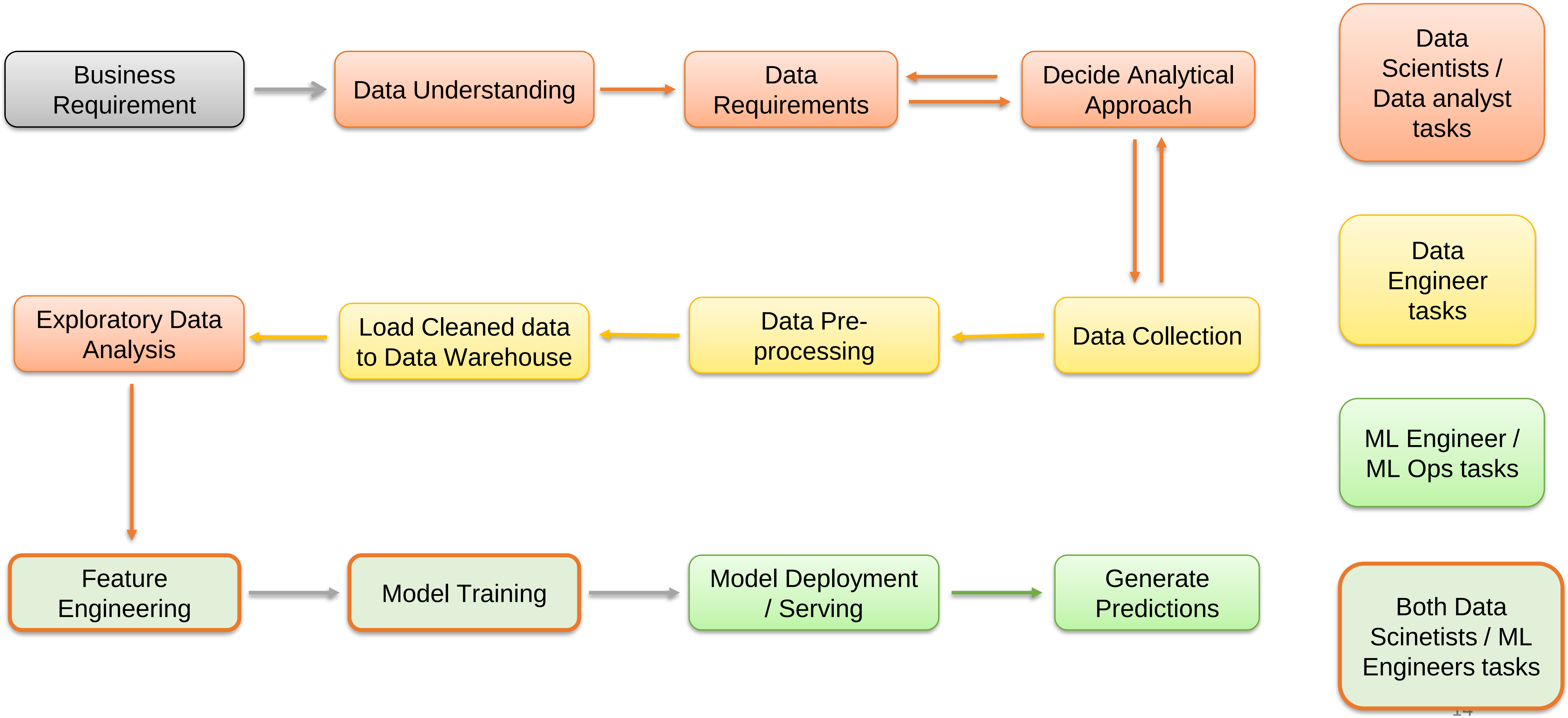
- Refers to accuracy or truthfulness of a data sources

- Raise the need for Data provenance
  - This is the process of tracing and recording the origin of data and their movements between locations
  - This recorded information, can be used to identify the source of processing error since it identifies all steps, transactions, and processes undergone by invalid data

Solution: Distributed ML

- A Distributed Machine Learning Framework provides the ability to:
  - Train over large data
    - Data split over multiple machines
    - Model replicas train over different parts of data and communicate model information periodically
  - Train over large models
    - Models split over multiple machines
    - A single training iteration spans multiple machines

Machine Learning Project Life Cycle



Training Stage and Prediction Stage

- Training Stage
  - Train a model using historical data and save it
  - The training phase ends when we dump the model to a file.

```
model = GradientBoostingRegressor(**params)
model.fit(X_train, y_train)
```

```
model.feature_names = list(X_train.columns.values)
```

```
joblib.dump(model, filename)
loaded_model = joblib.load(filename)
```

- Prediction Stage / Model Serving
  - The prediction phase starts when we load the saved model.
  - Model is applied to current data to predict outcomes in future

```
f_names = loaded_model.feature_names
loaded_model.predict(X_pred[f_names])
```

Model Training

- When the model is trained using historical data and used continuously for predictions, model might get outdated

- Re-Training the model at a defined frequency can help
  - Consider a moving window
  - E.g., Train the model monthly where data of the previous month is included when we train the model for this month

- Once trained model needs to be exported / saved and versions should be maintained in a model repo for rollback purposes

Model Export

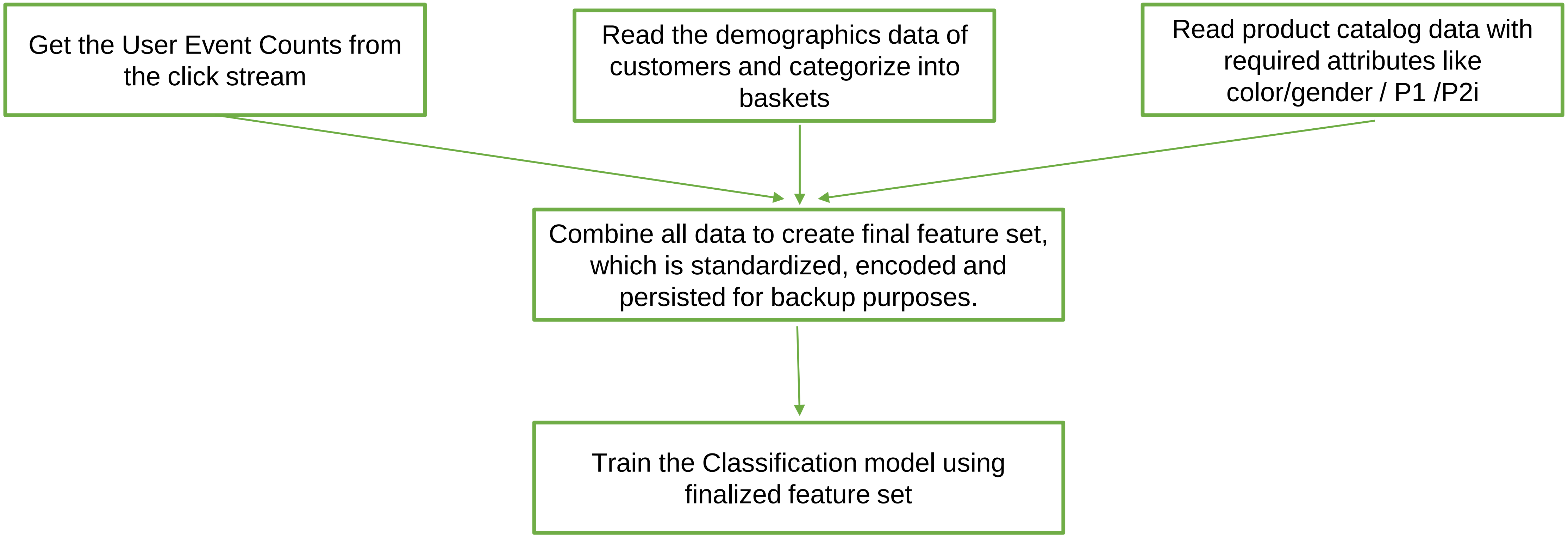
- Generally used formats
  - For python-based developments: .pkl file
  - For Java/Scala based developments: Java object serialization

- When productionizing, a platform-independent model export mechanism is needed because:
  - A single model can be consumed by many business applications
  - A single business app can be consuming more than one model
  - A model developed in python might need to be served in a Java based platform for availability and scalability requirements

- Generally used formats in model saving:
  - PMML: Predictive Markup Model Language – XML based
  - ONNX: Open Neural Network Exchange – ideal for deep learning

Orchestrating Model Training

Example DAG for training a purchase propensity model



Spark ML Lib / Spark ML

- ML Lib Shipped with Spark since Spark 0.8 with a RDD based API (now in maintenance)
- The primary Machine Learning API for Spark is now the DataFrame-based API in the spark.ml package
- Advantage is in scalability and in memory processing
- Supports different types of machine learning tasks with a collection of algorithms
  - Obtaining basic statistics
  - Classification & Regression
  - Clustering
  - Collaborative filtering & Frequent Pattern Mining
  - Feature extraction and transformation

Model Serving / Prediction Stage

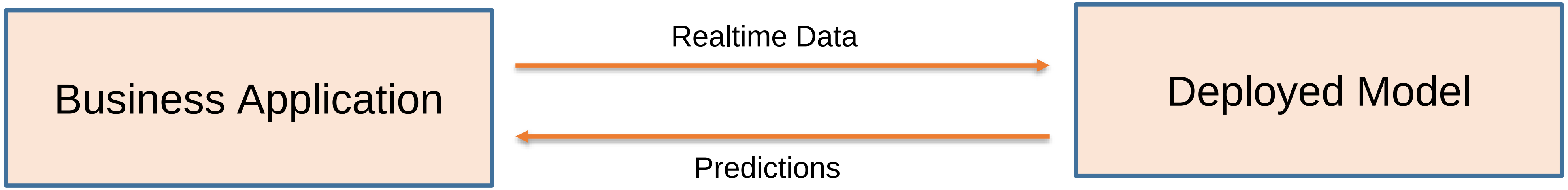
- Model serving refers to use of pre – trained machine learning model and serve the predictions to be used by other users / applications

- Idea is to wrap the prediction code as a production-ready service

- Model Serving can be done in two approaches depending on the use case:
  - **Batch Serving:** feed the model , typically as a scheduled job, with a large amount of data to pre calculate and store the predictions
  - **Online Serving:** Expose the model to be consumed by the users as required

Online Serving

- Deploy the model such that applications can send a request to the model and get a fast response at low latency.



- Possible Options:
  - Micro Service (Java based: Spring Boot /Play, Python based: flask / Django)
  - Third Part Tools: E.g., BentoML. TensorFlow Serving
  - Cloud Services: Azure ML, AWS SageMaker MLOps

Online Serving: Example Use Cases

- Healthcare:
  - Monitor patients' vital signs in real time and access medical histories and doctors' diagnoses to make critical predictions in real time

- Finance:
  - Useful in risk monitoring, like real-time fraud detection, algorithmic trading.

Model Monitoring

- Once deployed and being consumed in production models needs to be continuously monitored for their performance

- Product Recommendation Use Case:
  - Multiple algorithms used to generate recommendations in batch serving mode (pre-calculated)
  - Response for these suggested recommendations are monitored in real-time
  - If the click through rate is low another algorithm would be picked

READING

- D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. **Hidden technical debt in Machine learning systems**. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15). MIT Press, Cambridge, MA, USA, 2503–2511.

- A. L'Heureux, K. Grolinger, H. F. Elyamany and M. A. M. Capretz, "**Machine Learning With Big Data: Challenges and Approaches**," in IEEE Access, vol. 5, pp. 7776-7797, 2017, doi: 10.1109/ACCESS.2017.2696365.