INF438 – Bases de Données Avancées
Projet de Session

Titre du projet : Système Analytique pour Tournois d'E-sport et Analyse de Performance des Joueurs

Objectif : Intégration de données Batch & Streaming sur une architecture Lakehouse Azure

1. Contexte du Projet
Dans le cadre du cours, les étudiant.e.s réaliseront un projet complet d'ingénierie des données en s'appuyant sur une architecture Lakehouse sur le plateforme Microsoft Azure.

Le projet consiste à utiliser les données de matchs de League of Legends ou Dota 2 afin de construire un système capable de (d'):

- analyser les performances des joueurs,
- modéliser des statistiques avancées,
- produire des analyses en temps réel via une ingestion streaming,
- entraîner un modèle de *machine learning* simple pour la prédiction.

2. Services Obligatoires à Utiliser
Services Azure (obligatoires)

- Azure Data Lake Storage Gen2 – Stockage Lakehouse (Bronze / Silver / Gold)
- Azure Data Factory – Pipelines Batch & Orchestration
- Azure Databricks – Transformation, analyse et modélisation ML
- Power BI – Création de visualisations et dashboard

Services optionnels (bonus)

- **Azure Event Hubs** – Ingestion temps réel
- **MLflow (Databricks)** – Suivi d'expériences ML

3. Livrables du Projet

3.1. Architecture Lakehouse (Obligatoire)
Bronze – Données brutes

- Stockage des fichiers téléchargés depuis Kaggle (CSV/JSON)
- Nommage suggéré : bronze/raw_matches_YYYYMMDD.csv

Silver – Données nettoyées

- Nettoyage (valeurs manquantes, formats, types)
- Standardisation des colonnes
- Format : Delta Tables (Parquet-based)

Gold – Données analytiques

- Agrégations, features avancées, indicateurs métier
- Tables optimisées pour la BI et le ML
- Format : Delta Tables (Parquet-based)

Livrables

- Captures d'écran de la structure ADLS Gen2
- Diagramme d'architecture Bronze → Silver → Gold
- Exemples de fichiers de chaque couche

3.2. Pipeline Batch (Obligatoire)

Avec Azure Data Factory

- Pipeline ingestion Kaggle → Bronze
- Pipeline transformation Bronze → Silver
- Pipeline agrégation Silver → Gold
- Configuration d'un *trigger* (quotidien ou hebdomadaire)

Avec Tableicks

- Notebook de nettoyage (PySpark)
- Notebook de feature engineering
- Création et gestion de tables Delta

Livrables

- Fichiers JSON ou captures d'écran des pipelines
- Notebooks exportés (.ipynb)
- Logs d'exécution

3.3. Simulation Streaming (Obligatoire – Version simple)

Python

- Script qui rejoue les événements des matchs à partir des données Batch
- Écriture en mini-batch dans la couche Bronze
- Horodatage simulé (timestamp replay)

Livrables

- Script Python
- Preuve d'écriture dans Bronze
- Logs générés

3.4. Transformation & Fusion des Données (Obligatoire)

Transformations Bronze → Silver

- Gestion des valeurs nulles
- Conversions de types
- Suppression des doublons
- Détection et traitement des outliers
- Standardisation des noms de colonnes

Transformations Silver → Gold

- Agrégations par joueur :
    - KDA moyen, win rate, etc.
- Analyses par match :
    - durée, schémas de victoire
- Analyses Champion/Hero
- Features temporelles :
    - heure, jour, tendance

Livrables

- Documentation du logic de transformation
- Extraits de code PySpark

3.5. Analyses de Données (Obligatoire)

Analyses demandées

- **Top 10 joueurs** (KDA, win rate)
- **Évolution des performances** dans le temps
- **Meta Analyse** des champions / héros
- **Analyse de durée du match** et corrélations

Livrables

- Requêtes SQL complètes
- Notebook PySpark d'analyse
- Résultats présentés sous forme de tableau ou graphique

3.6. Modèle de Machine Learning (Obligatoire — modèle simple)

Modèle (au choix du groupe)

- Régression logistique
- Random Forest
- Decision Tree
- Autre modèle simple

Pipeline ML

- Feature selection
- Train/Test split
- Entraînement du modèle
- Hyperparamètres
- Évaluation du modèle
- Tracking MLflow

Livrables

- Notebook d'entraînement
- Importance des features
- Matrice de confusion et métriques
- Captures d'écran MLflow

3.7. Documentation Finale (Fichier de format PDF)

Contenu attendu

1. Introduction

- Contexte
- Jeux de données choisis
- Objectifs du projet
- Services AZURE utilisés

2. Architecture

- Schéma détaillé Lakehouse
- Rôle des couches Bronze/Silver/Gold
- Flux Batch et Streaming

3. Implémentation

- Configurations Azure
- Pipelines Data Factory
- Transformations Databricks
- Gestion Delta Lake
- Simulation streaming

4. Analyse & ML

- Requêtes SQL
- Résultats d'analyse
- Méthode et résultats ML

5. Difficultés & Solutions

6. Conclusion & Travaux futurs

Livrables finaux

- Code complet (scripts, notebooks)
- README.md
- Diagramme d'architecture
- Vidéo de démonstration

4. Barème d'Évaluation

Critère	**Poids**
Architecture Lakehouse (Bronze–Silver–Gold)	**25%**
Pipelines Batch (ADF + Databricks)	**20%**
Simulation Streaming	**15%**
Analyse des données (SQL / PySpark)	**20%**
Modèle ML	**5%**
Documentation finale	**15%**
Total	**100%**

5. Points Importants

1. Azure – Coût et limites

- Utiliser **Azure for Students**
- Éteindre les clusters Databricks après usage
- Auto-termination : 120 minutes
- Taille minimale recommandée : *Standard_DS3_v2*

2. Simulation Streaming

- Une API temps réel n'est pas obligatoire
- Le replay batch basé sur timestamp est suffisant
- Event Hubs = bonus

3. Exigences Techniques

- PySpark obligatoire
- Minimum 5 requêtes analytiques SQL
- Gestion d'erreurs
- Commentaires dans le code (FR ou TR)

4. Format de Remise

- Archive : Groupe3.zip
- Dépôt via moodle.gsu.edu.tr
- Aucune soumission par e-mail n'est acceptée

5. Jeu de données à utiliser

- Dota 2 Pro League : https://www.kaggle.com/datasets/bwandowando/dota-2-pro-league-matches-2023