Capstone Project

Project Title: Credit Card Churn Prediction

Organization: EXL

Track: Python + AI/ML + AWS Cloud Integration

Team: Individual

Submission Mode: GitHub + AWS + PPT Presentation

---

1. Business Scenario

EXL's Credit Card Analytics Division has observed a **decline in retention** among specific customer segments. The internal data science team wants a **proof-of-concept (POC)** solution to:

- Predict **which customers are likely to churn** based on behavioral and demographic data.
- Identify **early warning signals** using data science and visual analytics.
- Host the entire solution on **AWS Aurora MySQL + EC2**, simulating a production environment.

---

2. Key Business Definitions

What is **Churn**?

In this project, a customer is considered to have churned if:

- They **have not transacted for 3+ months**, and
- Their **average monthly spend** drops below 50% of their last known average, and
- They are marked as **"Churn = Yes"** in the dataset.

> This simulates real-world KPIs used by banks and financial firms to label inactive credit card customers.

---

3. Mandatory Dataset and Encoding Guidelines

Dataset File: `exl_credit_card_customers.csv`

- A sample 10-record CSV will be provided.
- Participants can expand with synthetic data (if required) but must **use the same columns and structure**.

Categorical Encoding:

- **All categorical variables must be encoded using One-Hot Encoding only.**
- No Label Encoding or manual mapping allowed.
- This ensures uniformity in feature space for all submissions.

---

4. Git & Branching Strategy

All submissions must be version controlled with the following Git structure:

```
main        → Final, clean working code
dev         → Main development branch
feature/eda  → EDA, visualization files
feature/ml   → Model training and testing
feature/aws  → AWS integration and scripts
presentation → PPT and documentation
```

Each participant must push their code to a **GitHub repo** and share access with reviewers.

---

5. Cloud Infrastructure Requirements

Participants must deploy their solution using **AWS Cloud Services** (free tier compatible):

Step-by-Step AWS Setup:

Step 1: Launch Aurora MySQL DB via Amazon RDS

- Choose Amazon Aurora (MySQL Compatible)
- DB name: `exl_churn_db`
- Create table: `customers`
- Import `exl_credit_card_customers.csv` into this table

Step 2: Launch EC2 Ubuntu Server

- Install Python 3.9+
- Setup virtual environment
- Install required libraries:
  - pandas, numpy, seaborn, scikit-learn, mysql-connector-python
- Clone your GitHub repository into the EC2 instance

Step 3: Execute Full Pipeline

- Load customer data from Aurora into Pandas DataFrame
- Perform preprocessing, model training (Random Forest regression only)
- Store predictions back into `churn_predictions` table in Aurora

---

6. Detailed Task Breakdown

1: Data Acquisition and Exploration

1. Load dataset into Pandas from local CSV
2. Upload data to Aurora MySQL
3. Run SQL queries from Python to validate upload

4. Perform full Exploratory Data Analysis:
    - Age distribution, tenure, card type analysis
    - Churn count vs non-churn
    - Correlation matrix

Note: Store EDA visualizations in the `/feature/eda` branch

2: Feature Engineering and Modeling

1. Perform One-Hot Encoding on all categorical features
2. Normalize numerical features using MinMaxScaler (mandatory)
3. Train Random Forest model only
4. Evaluate model using:
    - Confusion Matrix
    - Accuracy, Precision, Recall
    - Top 3 influential features by coefficient values

Note: Save model and results in the `/feature/ml` branch

3: AWS Execution + Final Reporting

1. Connect to Aurora from EC2 instance
2. Load test data from Aurora, run predictions, store back in `churn_predictions`
3. Prepare Final PowerPoint (5 slides max):
    - Problem Statement
    - EDA Summary
    - Model Output
    - AWS Architecture
    - Final Recommendations
4. Push final code and PPT to GitHub `/presentation` branch

5. Conduct internal presentation with manager panel

7. Evaluation Metrics

Component	Weight
Data Understanding	10%
Consistency of Encoding	10%
EDA & Visual Storytelling	15%
Model Accuracy & Rigor	20%
AWS RDS + EC2 Integration	20%
Git Workflow & Branching	10%
Presentation Readiness	15%

8. Submission Checklist

- [ ] Aurora MySQL setup with data loaded
- [ ] EC2 instance running full pipeline
- [ ] Random Forest model implemented with consistent preprocessing
- [ ] GitHub repo with all branches as specified
- [ ] PowerPoint presentation in `/presentation` branch
- [ ] Screenshots/logs of AWS components

Final Project Directory Structure

Project directory structure for your EXL Credit Card Churn Prediction project:

```
exl-credit-churn-analysis/
│
├── data/
│   ├── raw/
```

```
│   │   └── exl_credit_card_churn_data  # Original dataset
│   └── processed/
│       └── churn_cleaned.csv          # After cleaning & preprocessing
│
├── scripts/
│   ├── data_loader.py              # Script to load data from Aurora MySQL RDS
│   ├── data_cleaner.py             # Data cleaning: handle nulls, outliers, etc.
│   ├── feature_engineering.py        # Encoding, feature creation
│   ├── model_training.py           # Model training and evaluation
│   ├── model_predict.py            # Predict on new data
│   └── model_serializer.py         # Serialize model using pickle
│
├── model/
│   ├── churn_model.pkl             # Final trained model
│   └── model_metrics.txt           # Accuracy, precision, recall, F1-score
│
├── aws/
│   ├── rds_setup.md                # Step-by-step AWS RDS setup guide (Aurora MySQL)
│   └── ec2_connection.md             # Instructions to connect EC2 & access scripts
│
├── git/
│   └── git_flow.md                 # Git commands, branching (feature/* → dev), PR, merging
│
├── requirements.txt                # Python packages needed
├── README.md                       # Overview, objectives, and how to run the project
└── .gitignore                    # Files to ignore (e.g., .pyc, __pycache__)
```

SAMPLE - exl_credit_card_churn_data.csv

```
CustomerID,Gender,Age,Tenure,Balance,NumOfProducts,HasCrCard,IsActive
Member,EstimatedSalary,Churn
CUST0001,Male,45,3,120000.5,2,1,1,50000,0
CUST0002,Female,33,1,34000.0,1,0,1,62000,1
CUST0003,Male,52,6,58000.0,3,1,0,85000,0
CUST0004,Female,40,4,0.0,1,1,0,42000,1
CUST0005,Male,28,2,99000.0,2,0,1,37000,1
CUST0006,Female,35,5,40000.75,2,1,1,90000,0
CUST0007,Male,60,7,150000.0,4,1,0,76000,0
CUST0008,Female,48,3,85000.0,3,0,0,66000,1
CUST0009,Male,39,2,22000.2,1,1,1,58000,0
CUST0010,Female,50,4,76000.9,2,1,1,47000,1
```