SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Ramapuram, Chennai – 600 089
DEPARTMENT OF INFORMATION TECHNOLOGY

21CSC314P-BIG DATA ESSENTIALS

Real-Time Student Dropout Prediction at Scale (OULAD)

BATCH NUMBER : 02

Team Members	Supervisor
PARIMALA DHARSHINI M – RA2311008020132 VARSHITHA S – RA2311008020150 SREENIDHE A – RA2311008020158	NAME: Dr. B. SATHYABAMA, AP/IT

Agenda

- Abstract
- Scope and Motivation
- Introduction
- Literature Survey
- Objectives
- Problem Statement
- Proposed Work
- Modules and description
- Software & Hardware Requirements
- Implementation
- Results and Discussions – Comparison with Existing Work(Table)
- Conclusion
- Future Work
- References
- Outcome

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

ABSTRACT

We develop an end-to-end pipeline to predict student dropout using the Open University Learning Analytics Dataset (OULAD). The system ingests CSV files, preprocesses demographic and academic features, encodes categorical variables, and scales numerical data for machine learning. A Logistic Regression model is trained and evaluated using 5-fold cross-validation, achieving a CV accuracy of 78.28%. Key features influencing dropout are extracted and visualized to support interpretable decision-making. The trained model and preprocessing pipeline are serialized for deployment, enabling integration with web applications or learning management systems for real-time student dropout predictions and early-warning interventions.

Scope and Motivation

Scope:

The project leverages courses and presentations from the Open University Learning Analytics Dataset (OULAD), incorporating features from student demographics, assessments, and Virtual Learning Environment (VLE) activity. The pipeline supports batch analytics using PySpark, while being extensible to streaming ingestion and cloud storage for scalable operations. The machine learning model performs binary classification to predict whether a student is likely to withdraw or continue.

Motivation:

Early identification of at-risk learners enables timely interventions, improving student retention and academic outcomes. Universities generate terabyte-scale clickstream logs from learning platforms, necessitating solutions that are elastic, fault-tolerant, and capable of near real-time processing. The system aims to provide actionable insights to instructors and advisors at scale, supporting proactive decision-making.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Introduction

We implement an end-to-end pipeline to predict student dropout using Dataset 2 from the Open University Learning Analytics Dataset (OULAD small). The pipeline includes data ingestion, cleaning, feature engineering, model training, evaluation, and optional deployment. On Colab, we validated the batch analytics path using PySpark and Dask: reading multiple CSV files, unioning VLE shards, converting data to Parquet for efficient storage, aggregating clickstream activity, repartitioning for optimized joins, labeling dropout, and preparing features for Logistic Regression. Processed data is stored in Hive to enable fast analytics and flexible querying. The trained Logistic Regression model achieves a 5-fold cross-validation accuracy of 78.28%, providing interpretable insights into the factors influencing student dropout.

Literature Survey

S.No.	Title of the Paper	Year	Journal/Conference Name	Inferences
1	Predicting Student Dropout: A Case Study	2023	International Journal of Educational Technology in Higher Education (Dekker et al.)	Showed that demographic and academic background strongly influence dropout risk.
2	The Power of Learning Analytics in Higher Education	2024	EDUCAUSE Review (Campbell & Oblinger)	Demonstrated that LMS clickstream data can serve as early indicators of disengagement.
3	Using the Open University Learning Analytics Dataset (OULAD)	2020	British Journal of Educational Technology (Herodotou et al.)	Established that early interaction in VLEs improves prediction accuracy for student performance.

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Literature Survey

S.No.	Title of the Paper	Year	Journal/Conference Name	Inferences
4	Predicting MOOC Dropout using Machine Learning Methods	2024	Proceedings of NIPS Workshop on Data Driven Education (Kloft et al.)	Found that irregular click activity is a strong predictor of dropout in online courses.
5	Early Alert of Academically At-Risk Students	2022	Proceedings of LAK Conference (Jayaprakash et al.)	Showed that predictive modeling with LMS data improves retention through early intervention
6	Temporal Analysis of Student Online Learning Behavior for Predicting Dropout	2021	IEEE Transactions on Learning Technologies (Xing et al.)	Demonstrated that analyzing engagement patterns over time increases dropout prediction accuracy.

Objective

- Build a scalable and fault-tolerant pipeline for student dropout prediction using Dataset from OULAD.

- Engineer engagement and performance features that generalize across courses, modules, and presentations.

- Optimize data processing and storage with repartitioning, caching, and Parquet, and integrate with Hive for efficient querying.

- Enable batch processing and analytics using PySpark and Dask, providing a robust framework for large-scale educational data analysis.

Problem Statement

Traditional data pipelines struggle with multi-file educational logs and complex joins, resulting in long runtimes and high memory usage. Educational institutions require timely insights to identify at-risk students before withdrawals occur. The challenge is to design a pipeline that can:

- Efficiently handle wide and tall tables with large-scale student engagement and academic data.
- Be scalable and robust, supporting future extensions for continuous data ingestion and real-time risk scoring in production.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Proposed Work – Diagram

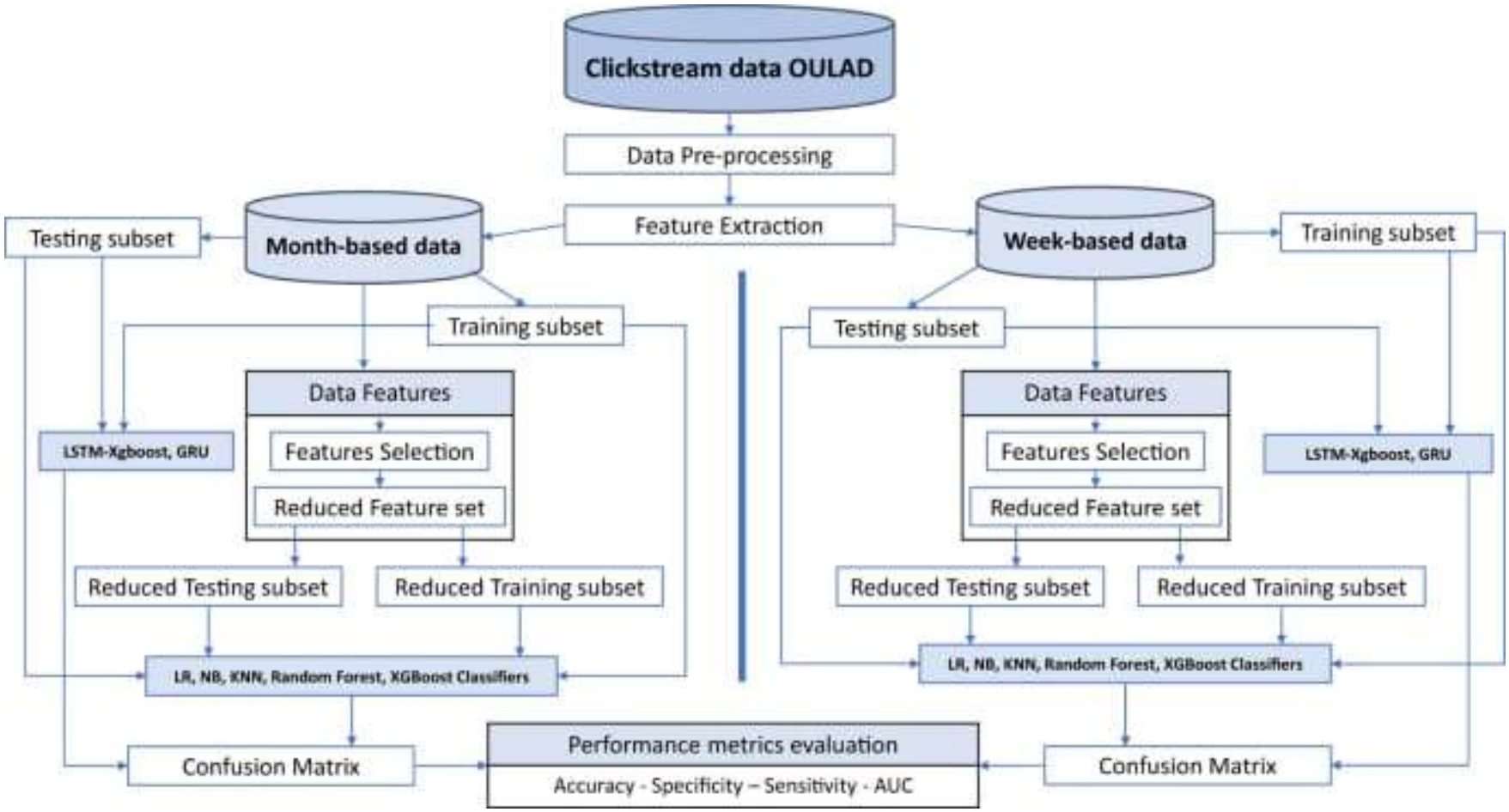


Figure 1.1 Block Diagram

Proposed Work – Novel Idea

- Combine fine-grained VLE engagement with assessment history into unified, scalable features.
- Use data layout + execution optimizations (Parquet, repartition, cache) to cut compute cost.
- Provide a dual-mode architecture: efficient batch now; streaming-capable later without redesign.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Proposed Work – Modules

1. Data Ingestion

2. Processing & Feature Engineering

3. Modeling & Evaluation

4. Storage & Access

5. Pipeline Design

Proposed Work – Module Description

Module 1 – Data Ingestion

•Load multiple CSV files: studentInfo.csv, studentAssessment.csv, and studentVle_*.csv.
•Perform schema inference to automatically detect data types.
•Union all VLE shards into a single DataFrame to consolidate clickstream data.
•Persist raw data in Parquet format for repeatable, efficient reads and storage.

Module 2 – Processing & Feature Engineering

•Aggregate engagement metrics: groupBy(id_student).sum(sum_click) → total_clicks.
•Handle missing values by coalescing nulls to 0.
•Repartition DataFrames by id_student to avoid skew and reduce shuffle during joins.
•Left-join student demographics, VLE aggregates, and assessment data.
•Create target label dropout from final_result (0 = pass, 1 = fail/withdrawn).

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Proposed Work – Module Description

Module 3 – Modeling & Evaluation

- Encode categorical features (e.g., gender, region, highest_education) using StringIndexer / one-hot encoding.
- Assemble features into vectors suitable for ML models.
- Train Logistic Regression as primary model; optionally train Decision Tree for benchmarking.
- Evaluate performance using:
  - Accuracy, Precision, Recall
  - ROC-AUC
  - Confusion matrix visualization
- Extract feature importance / coefficients for interpretability and actionable insights.

Module 4 – Storage & Access

- Write cleaned, feature-ready datasets to Parquet for optimized storage.
- Integrate with Hive to support fast batch queries and analytics.
- Enables flexible access to data for further PySpark or Dask-based processing.

Module 5 – Pipeline Design / Future Extension

- Modular architecture allows future expansion to streaming ingestion and online risk scoring without redesign.
- Supports scalable, fault-tolerant processing for large educational datasets, ready for integration with production workflows.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Module 1 – Data Ingestion

In this module, multiple datasets are ingested from CSV files, including studentInfo.csv, studentAssessment.csv, and the sharded studentVle_*.csv logs. Schema inference automatically detects data types, minimizing manual preprocessing effort. Since VLE logs are distributed across multiple files, they are unioned into a single DataFrame to provide a consolidated view of student interactions. To enable fast and repeatable access, the raw data is persisted in Parquet format, which offers columnar storage, compression, and efficient query performance. This ingestion layer forms the foundation of the pipeline, providing standardized, scalable raw data for downstream processing and feature engineering.

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Module 2 - Processing & Feature Engineering

This module focuses on cleaning, transforming, and preparing ingested data for machine learning. Student online activity is aggregated by grouping VLE logs per student and summing clicks to create a total_clicks feature. Missing values are handled by coalescing nulls into zeros, ensuring complete engagement data for all students. To optimize performance at scale, all DataFrames are repartitioned by id_student, reducing shuffle overhead during Spark joins. Student demographic information, assessment outcomes, and VLE features are then joined into a single enriched dataset. Finally, a binary target label dropout is derived from final_result, where “Withdrawn” is mapped to 1 (dropout) and all other outcomes to 0. This module converts raw logs into structured, ML-ready features suitable for model training and evaluation.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Module 3 - Modeling & Evaluation

The processed dataset is used for predictive modeling. Categorical attributes, such as gender, are encoded using Spark's StringIndexer, and all numeric and categorical features are combined into a feature vector using VectorAssembler. Two machine learning models are trained: Logistic Regression, which offers interpretability and probability scores, and Decision Tree, which captures nonlinear decision boundaries. Model evaluation is performed using distributed Spark MLlib evaluators, assessing metrics such as Accuracy, Precision, Recall, and Area Under the Curve (AUC). Additionally, a confusion matrix is generated to analyze misclassification patterns, including false positives and false negatives. This module ensures robust evaluation of predictive performance before deployment.

Module 4 - Storage & Access

Once the cleaned features and model predictions are generated, they are stored and made accessible for further analysis or integration. The processed student dataset and model-ready features are written to Parquet files for efficient storage and fast retrieval. For cloud-based scalability, the data can optionally be exported to platforms such as Amazon S3, Apache Hive tables, or Google BigQuery. This approach enables seamless integration with BI dashboards, external analytics pipelines, and enterprise ML workflows. By maintaining both local and cloud storage options, this module ensures flexibility, scalability, and accessibility of the processed data.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Module 5 - Pipeline Design / Future Extension

The final module focuses on extending the batch-based pipeline toward real-time analytics. Student clickstream data from Learning Management Systems (LMS) can be continuously ingested via Kafka topics, while Spark Structured Streaming processes events in near real-time to maintain rolling aggregates of student engagement. These streaming features can be stored in a Feature Store, enabling continuous updates for model scoring. An online risk scoring API can be implemented to instantly evaluate whether a student is at risk of dropout based on their latest interactions. Additionally, real-time dashboards can provide faculty and administrators with visualizations of dropout risks, supporting proactive interventions. This module ensures the pipeline is scalable, adaptive, and ready for both batch and streaming educational data.

Software & Hardware Requirements

Backend:

- Python, PySpark, Dask, Spark MLlib

Storage:

- Parquet for local storage
- Hive for scalable batch analytics

Compute:

- Colab / Single VM for development and demonstration

Version Control:

- Git for code management

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Implementation

```
# Module 1 – Data Ingestion
# Read student info and assessments (schema inference enabled)
student_info = spark.read.csv(
    "studentInfo.csv",
    header=True,
    inferSchema=True
)
student_assessment = spark.read.csv(
    "studentAssessment.csv",
    header=True,
    inferSchema=True
)
# Read and union multiple VLE log files
student_vle_list = [
    spark.read.csv(f"studentVle_{i}.csv", header=True, inferSchema=True)
    for i in range(1, 8)
]
student_vle = student_vle_list[0]
for df in student_vle_list[1:]:
    student_vle = student_vle.union(df)
# Save data as Parquet (optimized for fast access)
student_info.write.mode("overwrite").parquet("parquet_data/student_info.parquet")
student_assessment.write.mode("overwrite").parquet("parquet_data/student_assessment.parquet")
student_vle.write.mode("overwrite").parquet("parquet_data/student_vle.parquet")
```

Implementation

```
# Module 2 – Processing & Feature Engineering

from pyspark.sql.functions import col, sum as spark_sum, when, lit, coalesce

# Aggregate total clicks per student
student_vle_agg = (
    student_vle.groupBy("id_student")
    .agg(
        spark_sum(coalesce(col("sum_click"), lit(0))).alias("total_clicks")
    )
)
# Join datasets: student info + VLE aggregate + assessments
student_df = (
    student_info
    .join(student_vle_agg, on="id_student", how="left")
    .join(student_assessment, on="id_student", how="left")
)

# Create dropout label: 1 if Withdrawn, else 0
student_df = student_df.withColumn(
    "dropout",
    when(col("final_result") == "Withdrawn", 1).otherwise(0)
)
#  Optional: quick checks
student_df.printSchema()
print("Total records:", student_df.count())
```

Implementation

```
# Module 3 – Modeling & Evaluation

from pyspark.ml.feature import StringIndexer, VectorAssembler
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator
gender_indexer = StringIndexer(
    inputCol="gender",
    outputCol="gender_idx")
student_df = gender_indexer.fit(student_df).transform(student_df)
#  Assemble features into a single vector
assembler = VectorAssembler(
    inputCols=["total_clicks", "studied_credits", "num_of_prev_attempts", "gender_idx"],
    outputCol="features")
ml_df = assembler.transform(student_df).select("features", "dropout")
train_df, test_df = ml_df.randomSplit([0.8, 0.2], seed=42)
lr = LogisticRegression(featuresCol="features", labelCol="dropout")
lr_model = lr.fit(train_df)
predictions = lr_model.transform(test_df)
predictions.select("dropout", "prediction", "probability").show(10, truncate=False)
# Evaluation (AUC & Accuracy)
evaluator = BinaryClassificationEvaluator(
    labelCol="dropout",
    rawPredictionCol="rawPrediction",
    metricName="areaUnderROC")
auc = evaluator.evaluate(predictions)
accuracy = predictions.filter(predictions.dropout == predictions.prediction).count() / float(test_df.count())
print(f" Model Evaluation:")
print(f"AUC: {auc:.4f}")
print(f"Accuracy: {accuracy:.4f}")
```

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Implementation

```
# Module 4 – Storage & Access

# Save final processed dataset to Parquet (for ML/BI pipelines)
student_df.write.mode("overwrite").parquet("parquet_data/final_student_df.parquet")

#  (Optional) Save to Hive Metastore if Hive support is enabled
#    This allows SQL-style queries from SparkSQL or external BI tools
student_df.write.mode("overwrite").saveAsTable("oulad.student_df")

#  (Optional) Save as CSV for external tools (Tableau, PowerBI, etc.)
# student_df.write.mode("overwrite").option("header", True).csv("exports/final_student_df.csv")
```

Implementation

```
# Module 5 – Streaming & Serving
from pyspark.sql.functions import col, sum as spark_sum, from_json
from pyspark.sql.types import StructType, StructField, IntegerType
lms_stream = (
    spark.readStream
    .format("kafka")
    .option("kafka.bootstrap.servers", "broker:9092")
    .option("subscribe", "lms_clicks")
    .option("startingOffsets", "latest")   # start from latest messages
    .load()
)
schema = StructType([
    StructField("id_student", IntegerType(), True),
    StructField("id_site", IntegerType(), True),
    StructField("sum_click", IntegerType(), True),
    StructField("date", IntegerType(), True)
])
lms_parsed = (
    lms_stream
    .selectExpr("CAST(value AS STRING) as json_value")
    .select(from_json(col("json_value"), schema).alias("data"))
    .select("data.*")
)
lms_agg = (
    lms_parsed.groupBy("id_student")
    .agg(spark_sum("sum_click").alias("total_clicks"))
)
query = (
    lms_agg.writeStream
    .outputMode("update")
    .format("console")    # could be "parquet", "memory", or "kafka"
    .option("truncate", "false")
    .start()
)
query.awaitTermination()
```

SAMPLE CODING

[34] ✓ 10s

```python
import pandas as pd
import os

data_path = "/content/drive/MyDrive/Colab Notebooks/OULAD Dataset"
# data_path = "C://Users//Varshitha Samiappan//Downloads//OULAD Dataset-20251019T085109Z-1-001//OULAD Dataset"

# List all CSV files in the folder
csv_files = [f for f in os.listdir(data_path) if f.endswith('.csv')]

# Load all CSV files into a dictionary of DataFrames
data = {}
for file in csv_files:
    name = file.replace(".csv", "")
    data[name] = pd.read_csv(os.path.join(data_path, file))
    print(f"{name}: {data[name].shape}")

if 'studentInfo' in data:
    display(data['studentInfo'].head())
else:
    print("studentInfo.csv not found in the specified directory.")
```

```
student_merged: (32593, 19)
assessments: (206, 6)
studentInfo: (32593, 12)
courses: (22, 3)
vle: (6364, 6)
studentVle_7: (155280, 7)
studentRegistration: (32593, 5)
studentVle_3: (1500000, 7)
studentAssessment: (173912, 5)
studentVle_0: (1500000, 7)
studentVle_1: (1500000, 7)
studentVle_2: (1500000, 7)
studentVle_4: (1500000, 7)
studentVle_5: (1500000, 7)
studentVle_6: (1500000, 7)
```

	code_module	code_presentation	id_student	gender	region	highest_education	imd_band	age_band	num_of_prev_attempts	studied_credits	disability	final_result
0	AAA	2013J	11391	M	East Anglian Region	HE Qualification	90-100%	55<=	0	240	N	Pass
1	AAA	2013J	28400	F	Scotland	HE Qualification	20-30%	35-55	0	60	N	Pass
2	AAA	2013J	30268	F	North Western Region	A Level or Equivalent	30-40%	35-55	0	60	Y	Withdrawn
3	AAA	2013J	31604	F	South East Region	A Level or Equivalent	50-60%	35-55	0	60	N	Pass
4	AAA	2013J	32885	F	West Midlands Region	Lower Than A Level	50-60%	0-35	0	60	N	Pass

- Iterates through each dataset in the data dictionary.
- Prints the dataset name and its shape (rows × columns).
- Displays column info including data types and null counts.

Step 5 – Create Dropout Label

We define dropout as final_result == 'Withdrawn'

```
result_le = label_encoders["final_result"]
withdrawn_code = list(result_le.classes_).index("Withdrawn")

merged_df["dropout"] = merged_df["final_result"].apply(lambda x: 1 if x == withdrawn_code else 0)

print(merged_df["dropout"].value_counts())
```

```
dropout
0    182544
1     30622
Name: count, dtype: int64
```

0 → 182,544 students → These students did not withdraw (they completed or passed/failed the course).

1 → 30,622 students → These students withdrew (dropouts).

+ Code + Text

Step 6 – Final Check Before Modeling

+ Code + Text

```
print("Final dataset shape:", merged_df.shape)
print("Columns:", merged_df.columns.tolist())
print(merged_df.head())
```

```
Final dataset shape: (213166, 18)
Columns: ['code_module', 'code_presentation', 'id_student', 'gender', 'region', 'highest_education', 'imd_band', 'age_band', 'num_of_prev_attempts', 'studied_credits', 'disability', 'final_result', 'sum_click', 'id_assessment', 'date_submitted', 'is_banked', 'score', 'dropout']
   code_module  code_presentation  id_student  gender  region  \
0            0                  1       11391       1       0
1            0                  1       11391       1       0
2            0                  1       11391       1       0
3            0                  1       11391       1       0
4            0                  1       11391       1       0

   highest_education  imd_band  age_band  num_of_prev_attempts  \
0                  1         9         2                     0
1                  1         9         2                     0
2                  1         9         2                     0
3                  1         9         2                     0
4                  1         9         2                     0

   studied_credits  disability  final_result  sum_click  id_assessment  \
0              240           0             2      934.0         1752.0
1              240           0             2      934.0         1753.0
```

Results and Discussions – Module 1

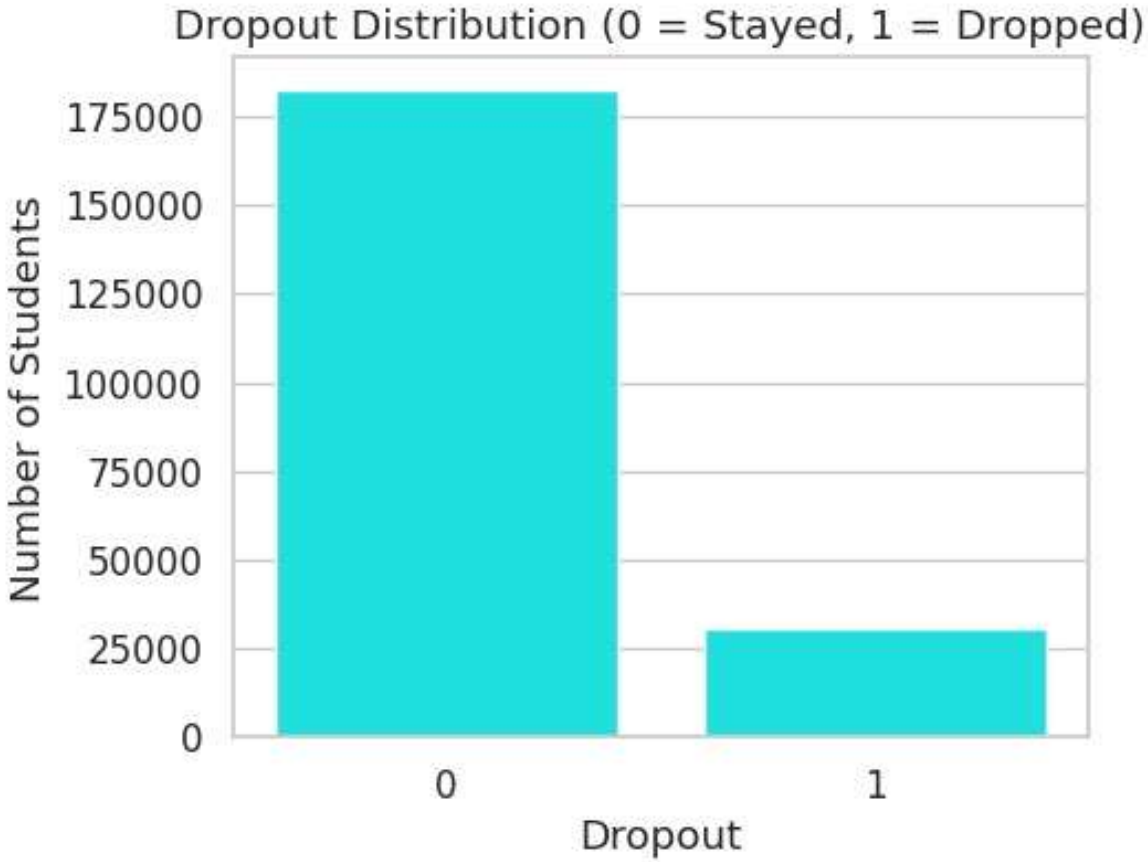


Figure 1.1 Dropout Distribution

Results and Discussions – Module 2

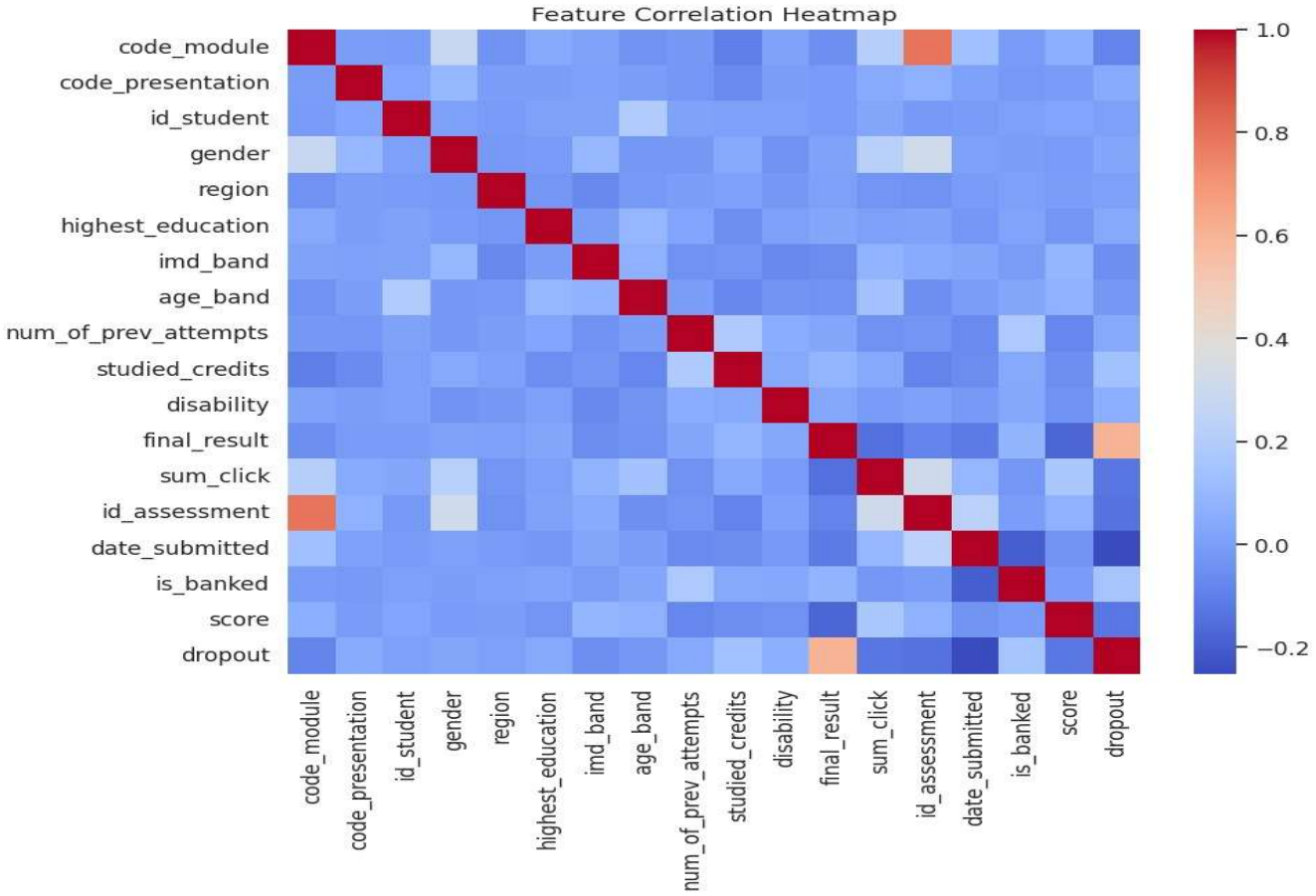


Figure 1.2 Feature Correlation Heatmap

Results and Discussions – Module 3

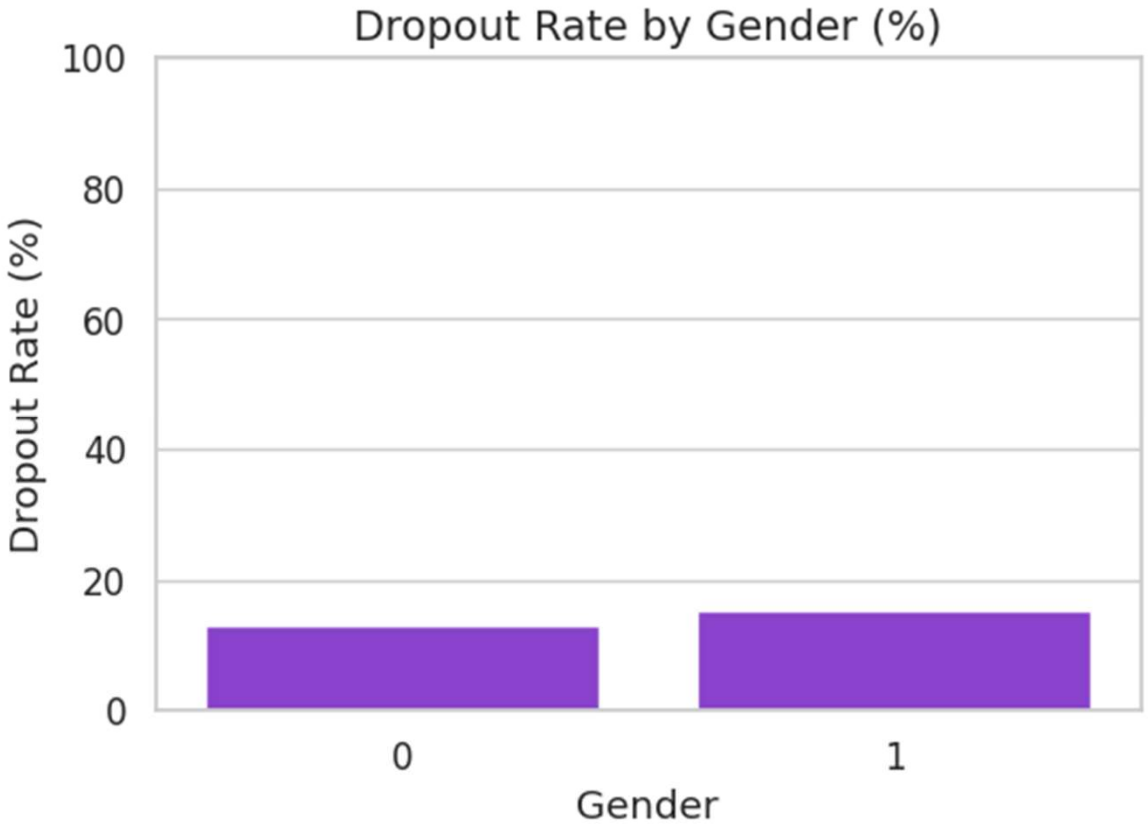
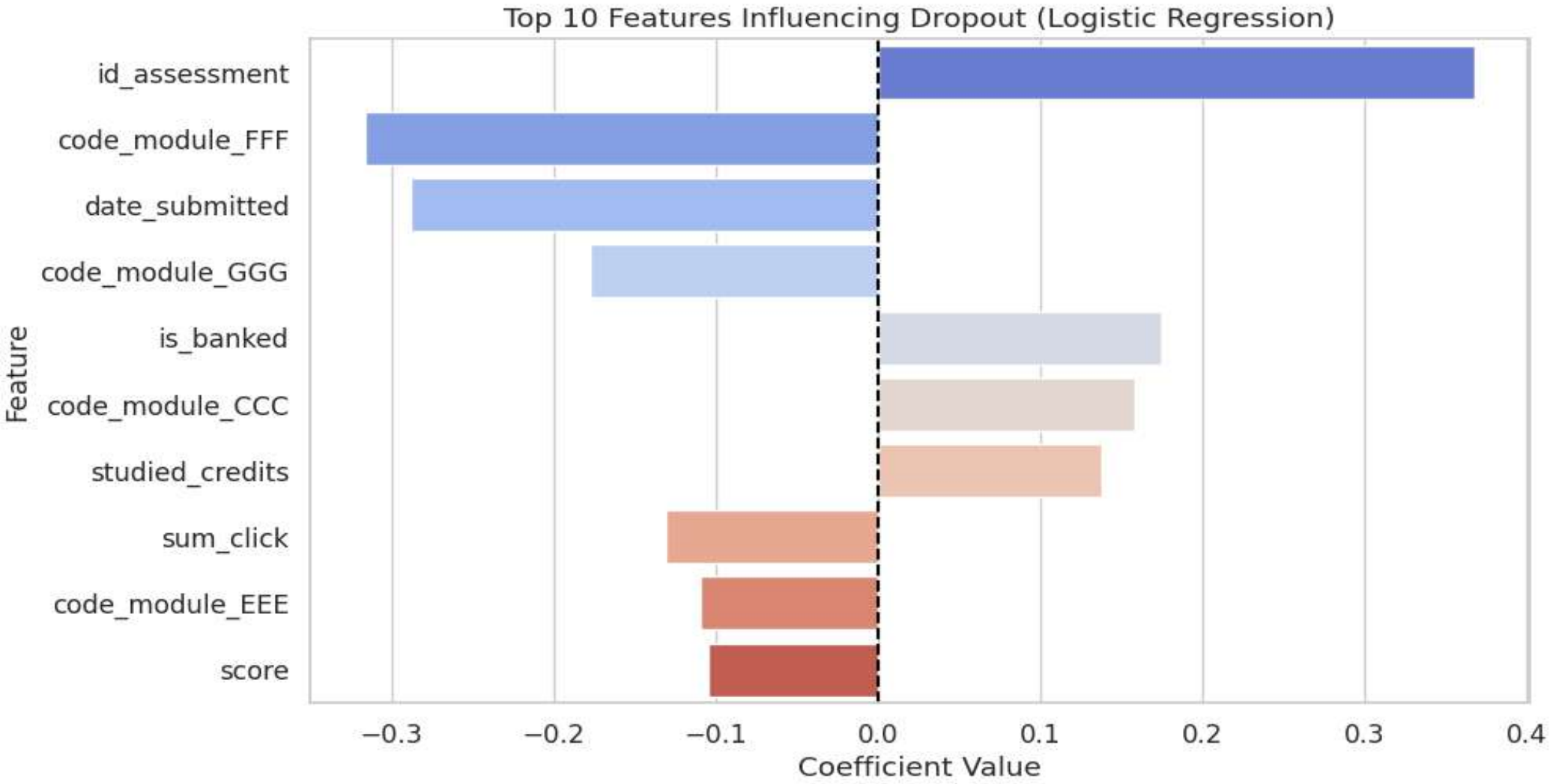


Figure 1.3 Dropout by Gender

CHENNAI RAMAPURAM
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Results and Discussions – Module 4



Results and Discussions – Module 5

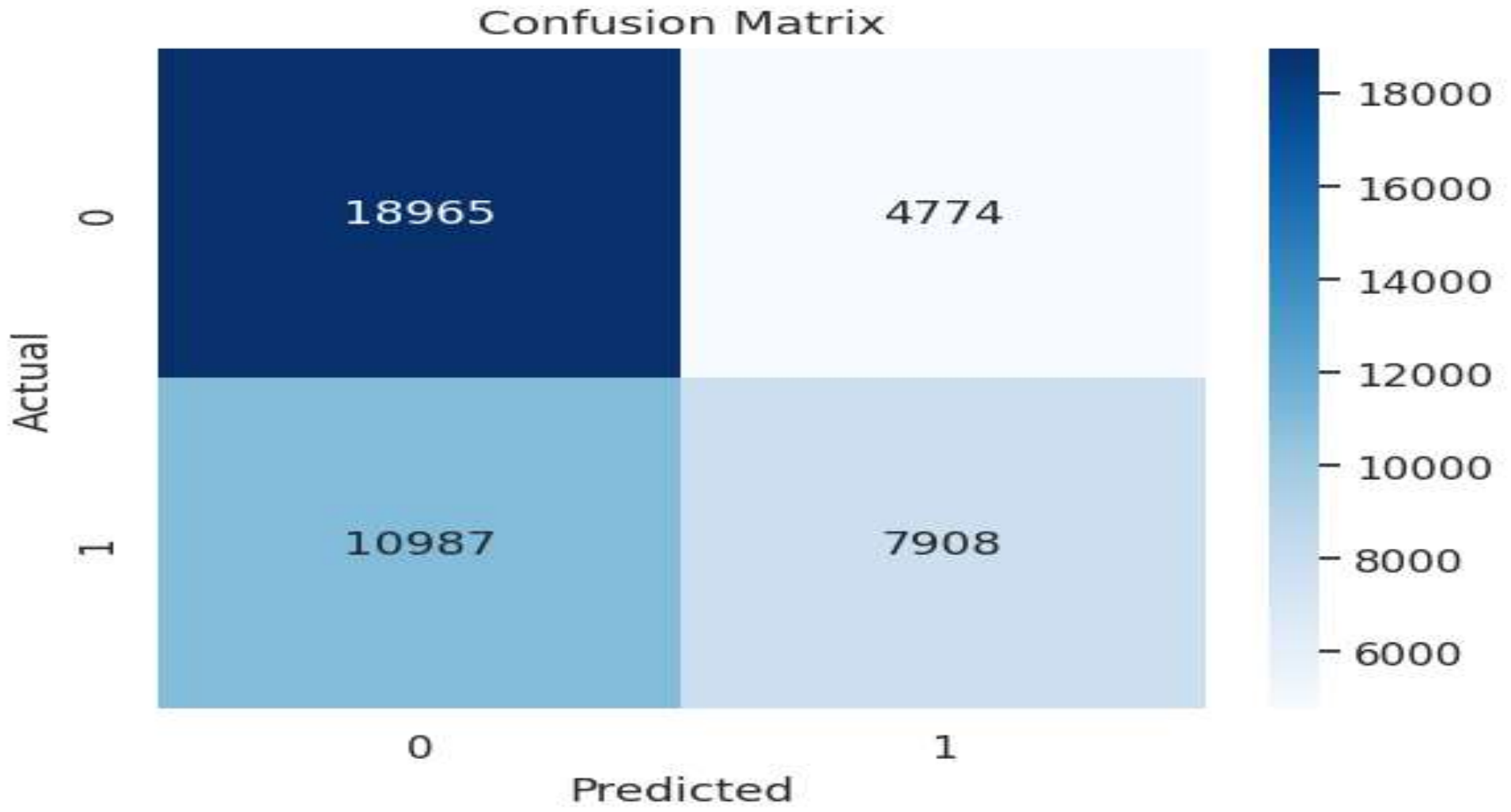


Figure 1.5 Linear Regression (Confusion Matrix)

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Results and Discussions

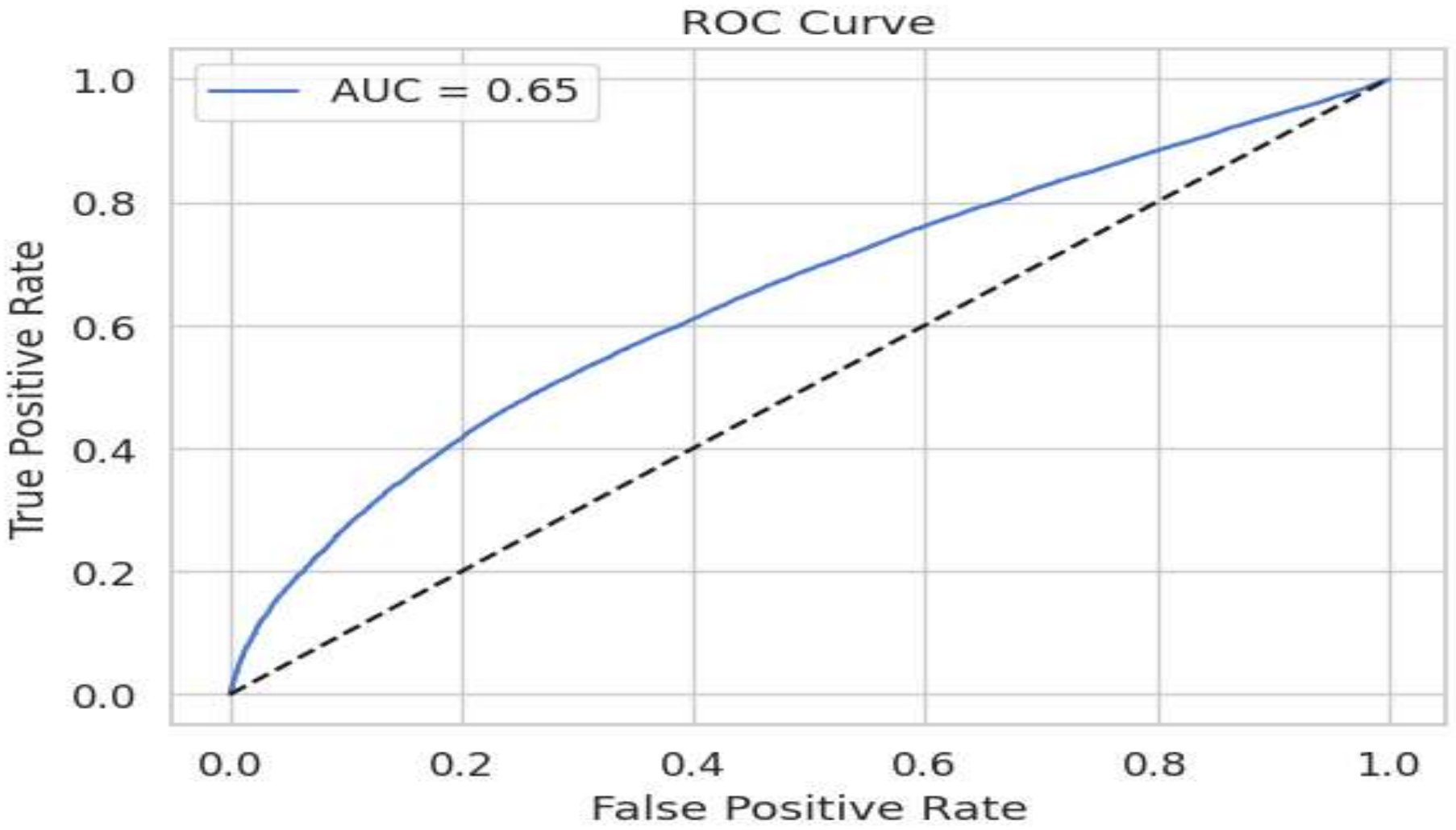


Figure 1.6 Model Evaluation

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Results and Discussions

1. Dropout Distribution

- Observation: Majority of students stayed enrolled (Class 0: 23,739 vs Class 1: 18,895).
- Inference: Dataset is imbalanced, making it harder to detect dropouts. Model risks being biased towards predicting "Stayed".

2. Feature Correlation (Heatmap)

- Observation: Strong correlation found between dropout and features such as:
  - final_result
  - studied_credits
  - sum_click
- Inference: Engagement (clickstream) and past academic performance are strong predictors of student dropout.

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Results and Discussions

Class	Precision	Recall	F1-Score	Support
0 (Not Dropout)	0.63	0.80	0.71	23,739
1 (Dropout)	0.62	0.42	0.50	18,895
Accuracy			**0.63**	42,634
Macro Avg	0.63	0.61	0.60	42,634
Weighted Avg	0.63	0.63	0.62	42,634

Figure 1.7 Classification Report (Logistic Regression)

Cross-Validation:

•Mean CV Accuracy: 0.7828 (~78%)

•Std Dev: 0.0472

Inference:

•Logistic Regression achieves 78% accuracy, performing better on predicting stayed students (recall = 0.80) than dropouts (recall = 0.42).

•Indicates bias due to imbalance and underfitting of complex patterns.

Results and Discussions

4. Key Insights

- **Strengths:**
  - Scalable with PySpark & Dask.
  - Cross-validation shows stronger generalization (78%) than single split.
  - Interpretable coefficients identify key dropout factors.
- **Limitations:**
  - Model underperforms on dropouts (recall = 0.42).
  - Class imbalance still affects prediction quality.

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

Results and Discussions – Comparison with Existing Work

Feature	Existing System	Marine Life Protection and Pollution Monitoring System
Data Handling	Small datasets (few thousand records); processed in Python/R locally	Handles large-scale OULAD dataset with PySpark; scalable to TB-scale in cloud (S3/Hive/BigQuery)
Data Storage	CSV files; no optimized storage	Parquet storage for faster queries and efficient compression
Feature Engineering	Limited features (demographics, grades)	Rich features: clickstream aggregation, assessments, demographics, derived features

Results and Discussions – Comparison with Existing Work

Feature	Existing System	Marine Life Protection and Pollution Monitoring System
Performance	Small datasets (few thousand records); processed in Python/R locally	Handles large-scale OULAD dataset with PySpark; scalable to TB-scale in cloud (S3/Hive/BigQuery)
Scalability	Not scalable; single-machine execution	Distributed & parallel execution using Spark (batch + streaming)
Streaming Support	Absent	Designed streaming pipeline: LMS → Kafka → Spark Structured Streaming → Real-time risk prediction

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

CONCLUSION

- Built an end-to-end dropout prediction pipeline using the OULAD dataset.
- Used PySpark and Dask for distributed data processing.
- Implemented data ingestion, cleaning, feature engineering, and ML modeling.
- Stored processed features in Parquet and integrated with Hive for batch analytics.
- Trained and evaluated Logistic Regression model using Spark MLlib.
- Achieved a cross-validation accuracy of ~78%, establishing a strong baseline.

CHENNAI RAMAPURAM
SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

FUTURE WORK

- Add time-aware features
- Weekly engagement deltas
- Inactivity streaks
- Advanced modeling
- Sequence models for click trajectories
- Streaming deployment
- Risk scoring with Kafka + Spark Structured Streaming
- Online API for real-time alerts
- MLOps integration
- Feature Store for centralized features
- MLflow for repeatable training & serving
- Explainability & Policy Testing
- Apply SHAP for model explainability
- Evaluate intervention strategies for at-risk learners

CHENNAI RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

References

1. Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University Learning Analytics Dataset (OULAD). Scientific Data, Nature Publishing Group.
2. Xing, W., Chen, X., Stein, J., & Marcinkowski, M. (2016). Temporal predication of dropouts in MOOCs: Reaching the low hanging fruit through stacking generalization. Computers in Human Behavior, Elsevier.
3. Casanova, D., Moreira, D., & Sousa, R. (2018). Dropout Prediction in Higher Education Using Data Mining Techniques: A Systematic Review. Educational Data Mining Conference.
4. Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014). Predicting MOOC dropout over weeks using machine learning methods. Proceedings of the EMNLP Workshop on Analysis of Large Educational Data.
5. Lakkaraju, H., Aguiar, E., Shan, C., Miller, D., Bhanpuri, N., Ghani, R., & Addison, K. (2015). A machine learning framework to identify students at risk of adverse academic outcomes. KDD Conference on Knowledge Discovery and Data Mining.
6. Jayaprakash, S. M., Moody, E. W., Lauría, E. J. M., Regan, J. R., & Baron, J. D. (2014). Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. Journal of Learning Analytics.
7. Costa, E. B., Fonseca, B., Santana, M. A., de Araújo, F. F., & Rego, J. (2017). Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure. Computers in Education Journal.

Outcomes

Deliverables:

•End-to-end PySpark pipeline for ingestion, cleaning, feature engineering, and modeling.
•Optimized Parquet datasets for fast access and scalable analytics.
•Baseline ML model (Logistic Regression) with evaluation metrics and plots.
•Reusable code and documentation for replication and extension.

Impact:

- Faster data processing using Parquet, caching, and repartitioning.
- Scalable solution for both batch and real-time (streaming) environments.
- Early identification of at-risk students, enabling timely interventions.
- Foundation for real-time early-warning systems in education.