NOVA IMS Information Management School

MDSAA

Master's Degree Program in

Data Science and Advanced Analytics

Big Data Analytics

Multi-Class Text Classification of Udemy Courses using Spark NLP & MLlib

Jude, Gbenimako, Number: 20240700

Group 40

CHALLENGE (unforeseen): ***"Due to personal reasons, my group members discontinued the course. As a result, I carried out the project independently(alone) and I did let the professor know about it."*** ***Thank you.***

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

June 2025

Index

1. Introduction & Background

Problem Statement and Objectives

Online education has significantly expanded, providing learners diverse courses across various subjects. This exponential growth has created a significant challenge in accurately categorizing educational content such as books, videos etc. Misclassification can lead to poor user experience and lower engagement rates. Categorizing course titles accurately ensure enhanced discoverability and improves user experience. This project leverages Natural Language Processing (NLP) and machine learning (ML) techniques implemented via Spark NLP and Spark MLlib to classify Udemy courses into distinct categories.

The key objectives include:

- Preprocessing Udemy course titles effectively.
- Extracting meaningful text features via NLP techniques.
- Developing accurate ML classification models.
- Evaluating model performance using robust metrics.
- Identifying the best-performing classification model.
- Testing the predictive accuracy of the classification models using single and batch examples.

Research Motivation

Efficient categorization using NLP and ML techniques ensures users find relevant courses effortlessly, enhancing learner satisfaction and platform usability.

2. Data Collection & Preprocessing

a. Data Sources

The Udemy dataset used is publicly accessible on Kaggle and the link is available on the notebook and also contained here, containing detailed course information such as course titles, categories, and unique identifiers.

b. Data Characteristics

The dataset primarily comprises **semi-structured** textual data. It is composed of 98,104 rows and 13 columns with attributes such as ID, title, and category among others. The categories with ranked count (Table: 1 on notebook) are Development (9945), Business (9912), IT & Software (9888), Teaching & Academics (9763),

Personal Development (9692), Design (9263), Health & Fitness (8559), Lifestyle (7426), Finance & Accounting (7324), Marketing (6277), Office Productivity (3955), Music (3854), Photography & Video (2246). This count distribution of category in the dataset through ranking counts, show Class imbalance which were investigated later.

c. Data Cleaning & Preprocessing Steps

To verify data quality, basic exploratory analysis was performed to understand text characteristics. It includes renaming selected relevant columns for the text classification implementation (id = course_id, tittle = course_title and category = subject) for clarity, usability and consistency. With a check, there were no duplicates or missing values.

From Table 1 and Fig 1 (class distribution plot on notebook), class imbalance is evident as the difference between the largest (Development (9945) and smallest class (Photography & Video (2246).) is more than 2.5x which is a significant gap. Therefore, I employed merging and sampling technique to mitigate the Risk of Bias classification towards the dominant classes. Similar subject categories were merged to consolidate related classes into fewer groups to improve performance (see Table 2). Marketing has distinct features and was kept as a category while other categories were merged into Business, Creative Arts, Education, Health & Lifestyle, and Tech.

A plot for WordCloud for all course titles (Fig 3) visualization showed the most dominant words across all categories are course, learn, complete, and beginners. These are not relevant words but will inform the feature engineering method choice to unweight their significance. A WordCloud per subject category plot (Fig 4-9) showed dominant words across course titles in each subject category, providing visual insight into keyword prominence per subject category, indicating meaningful textual patterns influencing categorization. These plots are contained in the appendix of this report.

Using some data preprocessing feature extraction methods, the course titles were cleaned of unwanted characters and adjusted to ensure that the resulting text is clean, consistent, and optimized for further NLP processing. Some of these methods applied are:

i) Data Preprocessing

- **Cleaning**: Lowercasing, punctuation removal, and trimming whitespace using the Regex function
- **Tokenization**: used to segment text into meaningful tokens.
- **Stopword Removal**: Commonly used English words with minimal analytical value were eliminated to retain meaningful tokens.

ii) Feature Extraction

- **Bigram**: Generated bigrams from the filtered_tokens column using NGram(n=2) to capture some context by looking at pairs of words, instead of just individual words (unigrams)
- **CountVectorizer**: Transformed course titles into numerical vectors for use in classifiers. It captures the frequency of tokens in each course title including bigrams provides additional context (e.g., "machine learning" vs. "machine" and "learning") separately.
- **TF-IDF**: Generated features emphasizing rare and significant terms. Performs feature importance selections by allocating high weights to textual features that are rare and distinctive (e.g., "blockchain", "yoga", "photography") which will contribute significantly to correct categorization and gives lower weights to common but uninformative terms (e.g., "course", "learn"). This improves model performance, enhances interpretability and trustworthiness of the model.

Other methods such as HashingTF, Word2vec (to capture semantic relationships between words) and Custom stopword removal were also explored but added no improvements on the model and therefore not implemented to avoid redundancy and inefficient pipeline.

3. Methodology & Tools

Machine Learning Techniques

First, I implemented stratified sampling (Stratified Split) on subject column after merging. Models were trained and evaluated using an 80/20 train-test split. 20% of each subject was sampled for the test set the remaining 80% was used as training data. This approach preserves class balance and avoids introducing bias due to skewed splits. Then Label Encoding with StringIndexer was fitted only on the training set (trainDF) to avoid data leakage. The assigned unique numeric labels are Business(0), Tech(1), Education(2), Health & Lifestyle(3), Creative Arts(4), Marketing(5). I also, use the decoding function (PySpark UDF (User Defined Function)) to ensures that predicted results are interpretable in their original context, such as "Business" or "Creative Arts", instead of raw label values like 0.0 or 4.0. To optimize performance, the transformed train dataFrame (trainDF) is cached in memory.

After defining the final text preprocessing pipeline for the text classification, I Built a Pipeline with all the transformers and the base/initial estimator as logistic regression which was employed due to its fast training and works well with large-scale data in

Spark and supports multi-class classification natively in Databricks. It is also one of the most accurate classifiers. The pipeline is as defined below:

lr = LogisticRegression(featuresCol="vectorizedFeatures", labelCol="label", maxIter=20, regParam=0.1, elasticNetParam=0.0)

pipeline = Pipeline(stages=[tokenizer, stopwords_remover, ngram, sql_transform, vectorizer, idf, lr])

Using the same pipeline but different classifier, Naive bayes and Random Forest were other ML classification algorithms used as benchmarks for comparison.

Model Evaluation

Model Evaluation Summary

Metric	Logistic Regression	Naive Bayes	Random Forest
Accuracy	0.7973	0.8055	0.2780
Precision	0.8017	0.8067	0.6897
Recall	0.7973	0.8055	0.2780
F1 Score	0.7980	0.8052	0.1906

The evaluation was conducted using accuracy, precision, recall, and F1 Score metrics. Interpretation: Naive Bayes performs best across all key metrics (Weighted F1 score = 0.8052 and Macro F1 Score 0.8025 – a balanced performance across all classes), slightly better than Logistic Regression. It was simple and fast. Logistic Regression (Weighted F1 score = 0.7980, Macro F1-score = 0.7961) is very close to Naive Bayes and could be preferred in scenarios requiring flexibility for better/improved accuracy due to correlated features. Random Forest underperforms significantly in both accuracy and recall, despite having moderately high precision but very low F1 score of 0.1906 and not good for an unbalanced dataset as per the project. This suggests it may correctly classify fewer samples overall but is somewhat confident in its few correct predictions. Recommendations: We recommend the use of Logistic Regression or Naive Bayes for production or deployment. Exclude Random Forest (for poor performance) for this text classification task, unless additional parameter tuning is/are made.

Querying & Storage

Data processing and querying were handled using Apache Spark DataFrames, facilitating scalable and efficient computation. This enables scalable and efficient handling of the large Udemy dataset with over 98,000 records. Spark's distributed in-memory computation allowed fast preprocessing and feature engineering for text classification. Data was stored and managed within the Databricks Lakehouse, ensuring fault tolerance and scalability. This setup supported the entire machine learning pipeline from data ingestion to model training and evaluation in a reproducible and efficient manner.

Visualization

Seaborn and Matplotlib provided visual insights through confusion matrices and Percentage confusion matrices (clearly highlighting areas of model strength and confusion) and exploratory data visualizations, highlighting model performance clearly. These visualizations were implemented in Databricks except for wordcloud plots (providing visual insight into keyword prominence per subject category, indicating meaningful textual patterns influencing categorization.) which were generated using Google Colab to optimize my Cluster & Runtime Restrictions. Key insights derived:

- Common terms across subject categories such as "complete,", "beginner" "guide," and "learn" indicated potential refinement areas for stopword removal.
- Visualization techniques clearly identify subject category specific keyword prominence and areas of model strength and confusion.

4. Conclusion

Based on the strong model results and predictions, the project demonstrated the effectiveness of Naïve baye combined with TF-IDF in handling text-based categorization tasks. However, the performance variance across different course subjects suggests potential for improvement in preprocessing and feature engineering. This project successfully leveraged Spark NLP and MLlib to categorize Udemy course data, providing actionable insights into effective feature selection and model performance. The adopted methodologies improved course discoverability as can be seen by the correct **predictions in classifying new single/batch of course titles (see Fig 11 appendix)**. The Insights from this analysis can directly enhance course discoverability and recommendation systems.

5. Incorporation of Feedback from discussion for Project Enhancements

Following the project discussion/defense, valuable feedback was received, highlighting critical areas for improvement to align with Big Data Analytics standards. The professor specifically pointed out two primary enhancements required for the project:

1. **Dataset Expansion:**

Initially, the analysis was conducted on a relatively small dataset consisting of only **3,678 rows**, significantly below the threshold typically considered substantial for big data analytics. It was probably overfitting with an **F1 Score of 0.9897 and 0.9911 for Naïve Bayes and Logistic Regression respectively**. The professor mentioned that a high F1 on small dataset is likely due to easier, less diverse data, possible overfitting, or cleaner labeling.

In response, a larger and more robust dataset (13 subject categories) comprising **98,104 rows** was sourced and integrated into the project. This substantial increase allowed for more representative modeling, improved accuracy, and the ability to derive deeper and more reliable insights reflective of real-world scenarios. So, the l**ower F1 on large dataset (0.8052= Naïve Bayes and 0.7980 = Logistic Regression)** likely **r**eflects real-world complexity, noise, imbalance, and harder classification challenges.

2. **Visualization Enhancement via WordCloud:**

Visualization, particularly through WordCloud, was recommended by the professor to better interpret and communicate the key textual features within each course category. However, technical constraints in the Databricks environment initially posed challenges.

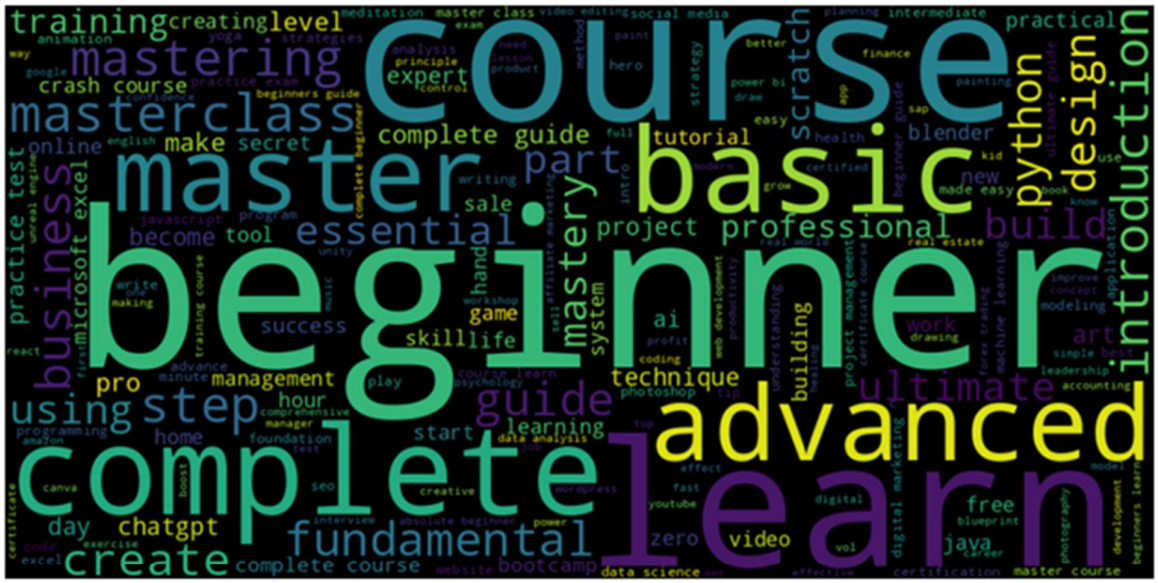
To effectively overcome these limitations, the WordCloud visualization component was implemented using Google Colab. This allowed the creation of insightful visual representations of prominent keywords and terms within each category, significantly improving the interpretability and presentation of textual data.

These enhancements have substantially elevated the analytical rigor and clarity of this project, aligning it with best practices in big data analytics and significantly enriching the interpretative and communicative value of the final outcomes.

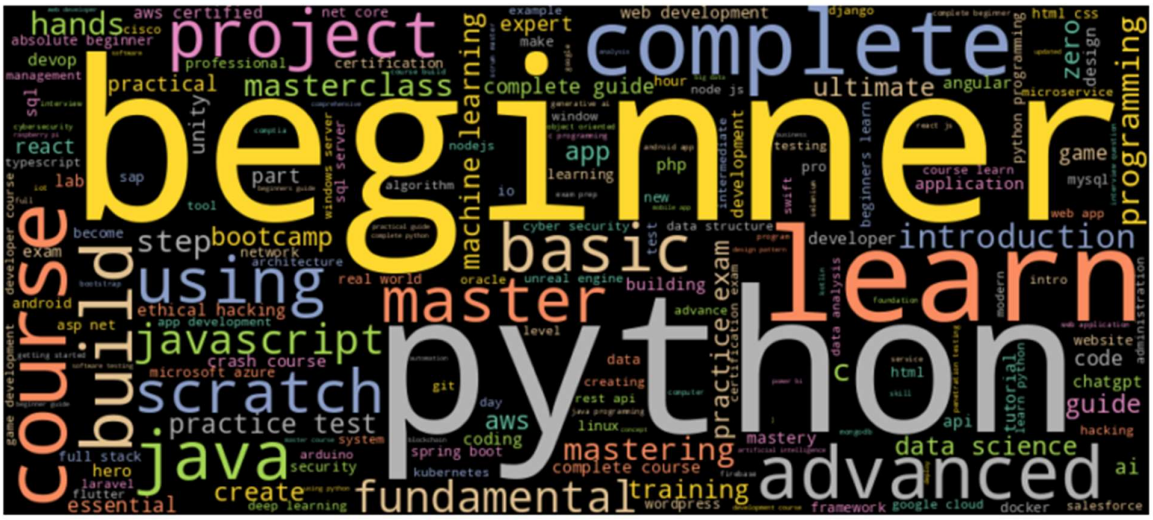
6. Appendices:

Fig 3-9

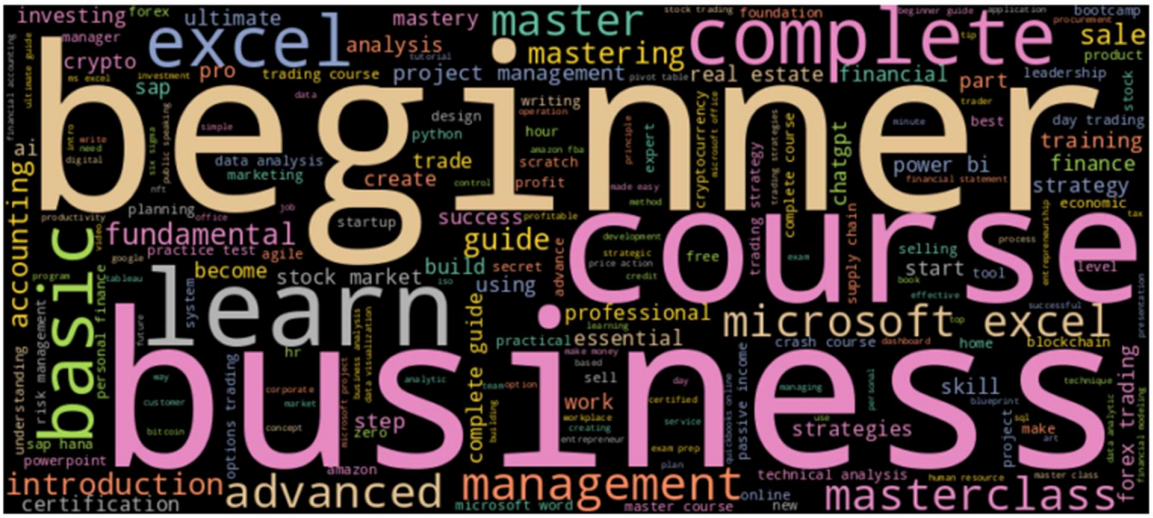
General WordCloud of All Course Titles



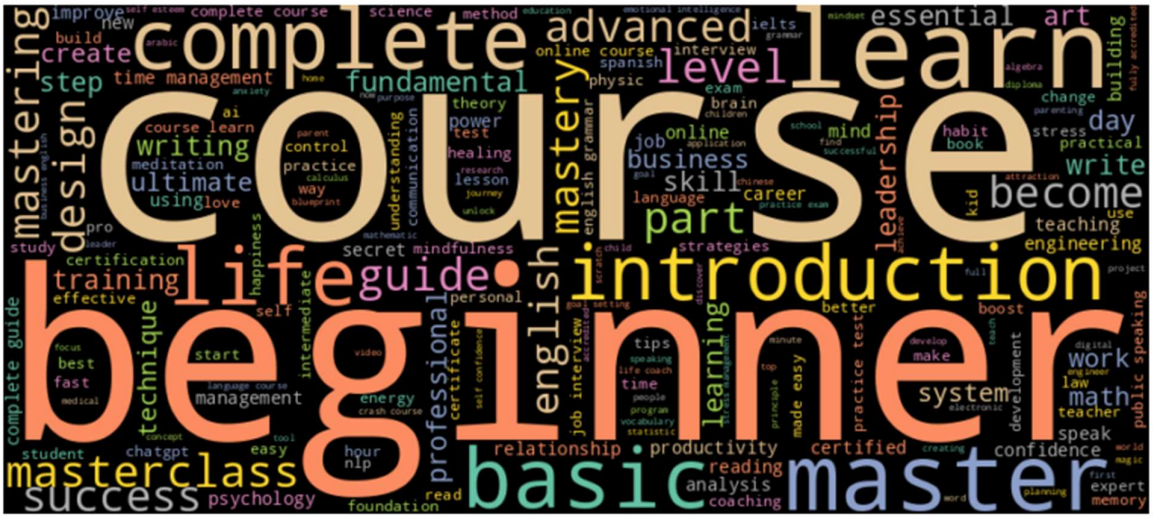
WordCloud for Tech Courses



WordCloud for Business Courses



WordCloud for Education Courses



WordCloud for Creative Arts Courses

Fig 11: Batch of new course titles prediction outputs

Predict a Batch of New Course Titles

Markdown

15 hours ago (2s) 159 Python

```python
batch_titles = [
    ("Business Strategy: Grow Your Consulting Firm",),
    ("Master Python Programming from Zero to Hero",),
    ("Complete Guide to Academic Research and Writing",),
    ("Yoga and Meditation for Stress Relief and Wellness",),
    ("Photoshop & Illustrator: Graphic Design Essentials",),
    ("Social Media Marketing with Facebook and Instagram",),
    ("Machine Learning for Business Decision Making",)
]

# Create DataFrame
batch_df = spark.createDataFrame(batch_titles, ["course_title"])

# Predict
batch_pred = lr_model.transform(batch_df)
batch_pred = batch_pred.withColumn("predicted_subject", label_to_subject_udf(batch_pred["prediction"]))

# Show result
batch_pred.select("course_title", "prediction", "predicted_subject", "probability").show(truncate=False)
```

(3) Spark Jobs

batch_df: pyspark.sql.dataframe.DataFrame = [course_title: string]

batch_pred: pyspark.sql.dataframe.DataFrame

```
+--------------------------------------------------+----------+-----------------+------------------------------------------------------------------------------------------------------------------------
---+
|course_title                                      |prediction|predicted_subject |probability
|
+--------------------------------------------------+----------+-----------------+------------------------------------------------------------------------------------------------------------------------
---+
|Business Strategy: Grow Your Consulting Firm      |0.0       |Business         |[0.6761452340243487,0.06360243439868714,0.10036432116995599,0.062412571273770104,0.05424468035273291,0.04323075878050534]
|
|Master Python Programming from Zero to Hero       |1.0       |Tech             |[0.026911605590558836,0.9206030019601054,0.017537658778269617,0.01240168254390305,0.01428662798275711,0.00825942314440624
2]  |
|Complete Guide to Academic Research and Writing   |2.0       |Education        |[0.33270310442116535,0.05571350600185713,0.4792596658525297,0.048061911134397,0.05296938738553583,0.03129242520451494]
|
|Yoga and Meditation for Stress Relief and Wellness|3.0       |Health & Lifestyle|[0.005183335056236825,0.004610566189244182,0.011186736897596505,0.972198802744052,0.004156005049596367,0.00266455406327417
62]|
|Photoshop & Illustrator: Graphic Design Essentials|4.0       |Creative Arts    |[0.03926234179606391,0.028055780488134077,0.04062440573926942,0.0288587218981915,0.8497743049145624,0.013424445163778808]
|
|Social Media Marketing with Facebook and Instagram|5.0       |Marketing        |[0.010060955929785585,0.007027195819877142,0.009616309616429258,0.00710017276482809,0.00809222168167012,0.958103144187409
8]  |
|Machine Learning for Business Decision Making     |0.0       |Business         |[0.7556646472922035,0.15843236104264047,0.03674851943277516,0.022618648509348745,0.02016744256534438,0.006368381157687609]
|
+--------------------------------------------------+----------+-----------------+------------------------------------------------------------------------------------------------------------------------
```