Projects for Artificial Intelligence and Machine Learning

This document describes the projects for Artificial Intelligence and Machine Learning. They are valid only for the academic year 2022/23. Projects are mandatory, whether you take the midterm/final or the oral exam. The project will contribute to 30% of your total grade in the class. A further 10% will be assigned on the day of the final/exam with a theoretical/technical question about the project.

Instructions

Choosing your project

You can work in groups of at most 3 people. The team's "captain" must send an email to gitaliano@luiss.it and ispinelli@luiss.it with the subject [PROJECT22/23] and cc the components of the team.
The email should contain:

First Project Preference

Name Surname student id of member 1 ("captain")
Name Surname student id of member 2
Name Surname student id of member 3

Second Project Preference

You must send the email by November 02, 2022. If you do not send an email by that date, you will be assigned to a project and to a team by the instructor.

When to submit your project.

You must submit your project at least 7 days before the exam date when you want to take the exam or register your grade. The first two exam dates are December 16 2022, and January 16 2023. You must submit your project by December 09, 2022, or by January 09, 2023.

What to submit for your project.

Each group must submit via mail to gitaliano@luiss.it and ispinelli@luiss.it, the URL of a GitHub repository. The repository's name should be the student id of the “captain”.

The repository must contain:

1) a “README.md” file with the following information:

• Title and Team members
• [Section 1] Introduction – Describe your project briefly.
• [Section 2] Methods – Describe your proposed ideas (e.g., features, algorithm(s), training overview, design choices, etc.) and your environment so that:
    o A reader can understand why you made your design decisions and the reasons behind any other choice related to the project
    o A reader should be able to recreate your environment (e.g., conda list, conda envexport, etc.)
    o It may help to include a figure illustrating your ideas, e.g., a flowchart illustrating the steps in your machine learning system(s)
• [Section 3] Experimental Design – Describe any experiments you conducted to demonstrate/validate the target contribution(s) of your project; indicate the following for each experiment:
    o The main purpose: 1-2 sentence high-level explanation
    o Baseline(s): describe the method(s) that you used to compare your work to
    o Evaluation Metrics(s): which ones did you use and why?
• [Section 4] Results – Describe the following:
    o Main finding(s): report your final results and what you might conclude from your work
    o Include at least one placeholder figure and/or table for communicating your findings
    o All the figures containing results should be generated from the code.
• [Section 5] Conclusions – List some concluding remarks. In particular:
    o Summarize in one paragraph the take-away point from your work.
    o Include one paragraph to explain what questions may not be fully answered by your work as well as natural next steps for this direction of future work

2) single notebook called “main.ipynb” with ALL the code used for the project.
   The notebook must have the following characteristics:

• Text and code cells must alternate from start to finish. The text cell above must describe the contents of the code below and its output so that a reader can easily follow up on your implementation. In particular:
    - o You must explain what you will do and why you chose to do so.
    - o You must explain the outputs of the cell (if any) with particular attention to describing figures such that a reader already knows what he is going to see.

3) An additional folder named “images” contains the figures displayed in the “README.md”.

Academic Integrity

You must write the code by yourself. The abuse of copy-paste will be taken into account during the evaluation. Any code that, for some (nonsensical) reason, is not written by yourself must be referenced (with a link to the original code).

Copying the projects from other teams is also strictly forbidden. Your code will be validated by anti-plagiarism software. In the unlikely event of two projects being very similar, we will follow the Netflix Prize rules: only the first project published on GitHub will get the grade, and the other will get nothing.

Datasets

All the datasets can be found (zipped) on the Luisslearn platform inside the folder Datasets in the section Project.

1) Credit Prediction

The greatest financial company in the world has collected bank details and credit-related information from all over the globe. Your task is to design a data-driven solution to reduce the manual burden ad divide the user into three credit score brackets: Poor, Standard and Good. Your solution will help the company to develop targeted products for each group.

Dataset:

Variables descriptions:

- Id: unique value assigned to each user
- age, occupation

- annual income, monthly salary, number of bank accounts, number of credit cards, number of loans
- Type of loan: types of loan taken by a person (a person may have multiple loans of different types
- Delay_from_due_date: average number of days delayed from the payment date
- Num_of_Delayed_Payment: average number of payments delayed by a person
- Changed_Credit_Limit: percentage change in credit card limit
- Num_Credit_Inquiries: number of credit card inquiries
- Credit_Mix: classification of the mix of credits
- Outstanding_Debt: remaining debt to be paid (in USD)
- Credit_Utilization_Ratio: utilization ratio of credit card
- Credit_History_Age: age of credit history of the person
- Payment_of_Min_Amount: Represents whether only the minimum amount was paid by the person
- Amount_invested_monthly: monthly amount invested by the customer (in USD)
- Payment_Behaviour: payment behaviour of the customer
- Monthly_Balance: monthly balance amount of the customer (in USD)

Specific Task

- Perform an Explanatory data analysis (EDA) with visualization.
- Generate a training and test set. The test set should be used only at the end.
- Preprocess the dataset (remove outliers, impute missing values, encode categorical features with one hot encoding, not necessarily in this order)
- Test at least 3 different classifiers. First, create a validation set from the training set to analyze the behaviour with the default hyperparameters. Then use 10-fold cross-validation to find the best set of hyperparameters. You must describe every hyperparameter tuned (the more, the better).
- Select the best architecture using the right metric (is the dataset balanced?)
- Finally, compute the performances of the test set.

2) Social media shares

Your company entrusts you to help the social media department analyse its communications' success. They want to develop a tool that predicts the number of shares on social media given the contents and the supposed publication time. Therefore they have collected a dataset to help you design the best machine-learning solution.

Dataset

Variables descriptions:

- tokens_title: Number of words in the title
- tokens_content: Number of words in the content
- unique_tokens: Rate of unique words in the content
- non_stop_words: Rate of non-stop words in the content
- non_stop_unique_tokens: Rate of unique non-stop words in the content
- hrefs: Number of links
- self_hrefs: Number of links to other articles published by our company
- imgs: Number of images
- videos: Number of videos
- token_length: Average length of the words in the content
- keywords: Number of keywords in the metadata
- lifestyle: Topic Lifestyle
- entertainment: Topic Entertainment
- bus: Topic Business
- socmed: Topic Social Media
- tech: Topic Tech
- world: Topic World
- kw_min_min: Worst keyword (min. shares)
- kw_max_min: Worst keyword (max. shares)
- kw_avg_min: Worst keyword (avg. shares)
- kw_min_max: Best keyword (min. shares)
- kw_max_max: Best keyword (max. shares)
- kw_avg_max: Best keyword (avg. shares)
- kw_min_avg: Avg. keyword (min. shares)
- kw_max_avg: Avg. keyword (max. shares)
- kw_avg_avg: Avg. keyword (avg. shares)
- self_reference_min_shares: Min. shares of referenced articles in Mashable
- self_reference_max_shares: Max. shares of referenced articles in Mashable
- self_reference_avg_shares: Avg. shares of referenced articles in Mashable
- monday: Was the article published on a Monday?

- tuesday: Was the article published on a Tuesday?
- wednesday: Was the article published on a Wednesday?
- thursday: Was the article published on a Thursday?
- friday: Was the article published on a Friday?
- saturday: Was the article published on a Saturday?
- sunday: Was the article published on a Sunday?
- weekend: Was the article published on the weekend?
- LDA_00: Closeness to LDA topic 0 (LDA: a statistical model for discovering the abstract topics, aka topic modelling.)
- LDA_01: Closeness to LDA topic 1
- LDA_02: Closeness to LDA topic 2
- LDA_03: Closeness to LDA topic 3
- LDA_04: Closeness to LDA topic 4
- global_subjectivity: Text subjectivity
- global_sentiment_polarity: Text sentiment polarity
- global_rate_positive_words: Rate of positive words in the content
- global_rate_negative_words: Rate of negative words in the content
- rate_positive_words: Rate of positive words among non-neutral tokens
- rate_negative_words: Rate of negative words among non-neutral tokens
- avg_positive_polarity: Avg. polarity of positive words
- min_positive_polarity: Min. polarity of positive words
- max_positive_polarity: Max. polarity of positive words
- avg_negative_polarity: Avg. polarity of negative words
- min_negative_polarity: Min. polarity of negative words
- max_negative_polarity: Max. polarity of negative words
- title_subjectivity: Title subjectivity
- title_sentiment_polarity: Title polarity
- abs_title_subjectivity: Absolute subjectivity level
- abs_title_sentiment_polarity: Absolute polarity level
- is_popular: Whether or not the article was among the most popular ones based on shares in social media.

Specific Task

- Perform an Explanatory data analysis (EDA) with visualization.
- Generate a training and test set. The test set should be used only at the end.
- Preprocess the dataset (remove outliers, encode categorical features with one hot encoding, not necessarily in this order)
- Test at least 3 different regressors. First, create a validation set from the training set to analyze the behaviour with the default hyperparameters. Then use 10-fold

cross-validation to find the best set of hyperparameters. You must describe every hyperparameter tuned (the more, the better).
- Select the best architecture using the right metric
- Finally, compute the performances of the test set.
- Now select a subset of the attributes retaining only the most relevant. Train again your best model and compare the performances.

3) Customer Segmentation

One of the largest department stores on earth has collected data regarding its subsidiary in Brazil. They want to segment their customers to develop a targeted email campaign. As a marketing team member with a vast knowledge of machine learning, your job is to identify the ideal number of partitions and assign each user in the dataset to one of them. The collected dataset has a lot of information. It consists of a list of orders with information regarding the users, the sellers, payments, products and geolocation data. Not everything is important for performing customer segmentation, but it may be interesting to analyze in your Explanatory Data Analysis (EDA). To accomplish your goal, you should develop a strategy known as RFM (recency, frequency and monetary value):

- **Recency value** refers to the time since a customer's last purchase.
- **Frequency value** refers to the number of times a customer has made a purchase.
- **Monetary value**: refers to the total amount a customer has spent purchasing products

Dataset

Variables descriptions:

- order_id: unique order identifier
- customer_id: the key to the orders dataset. Each order has a unique customer_id
- customer_unique_id: the unique identifier of a customer.
- customer_city: customer city name
- customer_state: customer state
- order_item_id: sequential number identifying the number of items included in the same order.
- product_id: product unique identifier
- price: item price

- freight_value: item freight value item (if an order has more than one item, the freight value is split between items)
- payment_type: method of payment chosen by the customer.
- payment_installments: number of instalments chosen by the customer.
- payment_value: transaction value.
- order_status: the order status (delivered, shipped, etc).
- order_purchase_timestamp: purchase timestamp.
- order_approved_at: payment approval timestamp.
- order_delivered_carrier_date: order posting timestamp. When it was handled by the logistic partner.
- order_delivered_customer_date: actual order delivery date to the customer.
- order_estimated_delivery_date: the estimated delivery date informed to the customer at the purchase moment.
- product_id: unique product identifier
- product_category_name: root product category, in Portuguese.
- product_category_name_english: root category of product, in English
- product_description_lenght: number of characters extracted from the product description.
- seller_id: seller unique identifier
- seller_city: seller city name
- seller_state: seller state

Specific Task

- Perform an Explanatory data analysis (EDA) with visualization using the entire dataset.
- Preprocess the dataset (remove duplicates, encode categorical features with one hot encoding, not necessarily in this order).
- Use at least 3 clustering techniques to perform market segmentation based on the RFM scores
- Identify the proper number of clusters, and evaluate different options.
- Describe the properties of the clusters you have identified.
- Describe the properties of the customers belonging to each cluster

4) Recommender System

As a member of the data science team at a prestigious fast fashion firm, your job is to help increase revenues through all possible (legal) means. The online platform is going strong. However, you would like to improve the recommender system on your platform. Your plan is to test different recommendation systems and pick the most suitable one.

Dataset

It contains 3 files:

recsys_customers.csv: users' metadata

- customer_id: unique customer identifier
- fashion_news: customer subscribed to the newsletter
- club_member: customer part of the special club member
- age: customer age

recsys_transactions.csv: “training data” consisting of the purchases of each customer Duplicate rows correspond to multiple purchases of the same item.

- t_dat: purchase date
- customer_id: unique customer identifier
- article_id: unique article identifier

recsys_articles.csv: articles metadata. Most attributes have “code” and “name” descriptions. For example, product_type encodes the type of the product with a number meanwhile, 'product_type_name' has a textual description. In your pipeline, use numerical attributes. However, the textual descriptions may help perform the Explanatory Data Analysis.

- article_id: unique article identifier
- prod_name: article name
- product_type & product_type_name
- product_group & product_group_name
- colour_group & colour_group_name

- perceived_colour_value & perceived_colour_value_name
- perceived_colour_master_id & perceived_colour_master_name
- department & department_name
- index & index_name
- index_group & index_group_name
- section & section_name
- garment_group & garment_group_name

Specific Task

- Perform an Explanatory data analysis (EDA) with visualization using the entire dataset..
- Preprocess the dataset (impute missing values, encode categorical features with one hot encoding).
- Use at least 2 recommender system pipelines (collaborative, content-based, hybrid) and tune their hyperparameters
- Compare the performances and select the optimal solution describing your decision process to pick the best algorithm.