INSY 446: Data Mining for Business Analytics (Winter 2024)

Individual Project – Worth 20%

Due date: 12 April (11:59pm)

This project is to be done individually. All the coding involved in this project must be in Python. **You can only use techniques/algorithms covered in this class**. However, you are allowed to use parameters, attributes, etc. that are not covered in this class as long as they belong to the techniques/algorithms covered in the class. For example, *Random Forest* is the algorithm covered in the class but *warm_start=true* is the parameter that is not covered in the class. You are allowed to use *warm_start=true* with *Random Forest*. Meanwhile, *XGBoost* is the algorithm that is not covered in the class, so you are not allowed to use *XGBoost* in this project. Similarly, the use of libraries like *TensorFlow*, *Keras* and *PyTorch* is not allowed for this individual project.

Your task in this assignment is to use the Kickstarter dataset to:

1. **Develop a classification model** (i.e., a supervised-learning model where the target variable is a categorical variable) to predict whether the variable "state" will take the value "successful" or "failure." After you obtain the final model, explain the model and justify the predictors you include/exclude.

** ***Important***: The classification task is assumed to be done at the time each project is launched. In other words, we execute the model to predict whether a new project is going to be successful or not, at the moment when the project owner submits the project. Therefore, the model should only use the predictors that are available at the moment when a new project is launched.

2. **Develop a clustering model** (i.e., an unsupervised-learning model which can group observations together) to group projects together. After you obtain the final clusters, explain the characteristics that you observe in each cluster.

****** For all tasks, only include observations where the variable "state" takes the value "successful" or "failure" (i.e., all observations with other states should be dropped).

Note that you will be graded based on both the performance of the model and the explanations/justifications you provide. You also need to clearly articulate how realistic or useful your model would be in a business context.

There are two deliverables for this assignment:

1) Summary Report
   - The report must be submitted in .pdf format.
   - The report must not exceed 3 double-spaced pages, **including everything**. Page margins must measure 1" around. Please use a 12-point Times New Roman font.

- Name the file as follows: "Lastname_Firstname_IndividualProject"
- The report must contain the explanations/justifications of each model that you submit. You may submit only one model per task.
- The report is due by Friday, April 12, 2024, at 11:59pm.

2) Python Code (**in .py format only**)

Along with the report, please also submit Python code (in .py format) that you use to develop your models. The code should be complete with informative comments and able to run fully without any errors or modifications (besides the file path).

Grading Criteria

For the classification model, you will be graded based on the relative performance of your model (compared to your classmates' models). Please note that you will be heavily penalized if you use an invalid predictor in your model (predictors are considered invalid if they can only be realized after the prediction starts). The small portion of the grade is dedicated to the explanations/justifications of your model, and how useful your model is in the business context.

For the clustering model, you will be graded based on the insights you obtain from the clustering task. For example, if your model generates two clusters with a high cluster separation and low cluster cohesion but one cluster essentially represents successful projects while another cluster essentially represents failed projects, then the insights gained are fairly limited.

Hints

1) Make sure to understand the predictors that you plan to include in your model. If the information in the Data Description section below is not clear, please visit kickstarter.com to obtain additional information.
2) If your model performs too well even on the test dataset (e.g., your classification accuracy is 1), you either accidentally include the target variable as a predictor in your model or one of your predictors is perfectly correlated with the target variable.
3) Good feature engineering can significantly increase the performance of the classification model.

Structuring your Code for Grading Purposes

In this assignment, we will evaluate your model using the future dataset that is not available to you. To facilitate the grading process, please write a separate code that allows the grader to apply your classification model based on the grader's "grading" dataset. Specifically, please write a separate code that reads an input file named "Kickstarter-Grading.xlsx" (this file has exactly the same structure as "Kickstarter.xlsx", which is the dataset of this project). Should you do any preprocessing, the relevant code should also be applied to the grading set. Then, generate the accuracy score with this grading dataset, using the model that was developed based on the original data you were given. This script should essentially allow the grader to test the performance of your model with the grading data. Be sure that your script does not train a new model with this new data, and that the model used to generate the accuracy score is developed based on the original data.

If this instruction is still not clear, please feel free to email me for further clarifications or schedule an office hour session to chat with me.

The penalty for failing to do this is up to 10% of your total grade for the individual project.

For illustration, the sample code ("IndividualProject-Sample.py") and the sample grading dataset ("Kickstarter-Grading-Sample.xlsx") are provided. The content of the real grading dataset is different than that of the one provided. Do **NOT** measure the performance of your model based on the sample grading dataset.

Data Description

The dataset in this project is scraped from Kickstarter, which is a popular crowdfunding platform. There are 45 variables in total. The table below contains a short description of each variable.

Column	**Name Description**
id	Unique identifier for projects
name	Project
goal	Goal amount requested by the project
pledged	Amount pledged at time of data scrape
state	Status of the project (successful, failed, etc)
disable_communication	If communication with project owners was disabled
country	Origin country of project
currency	Currency of origin country
deadline	End date of project funding period
state_changed_at	date and time the project state was modified to current state
created_at	Date and time project was created
launched_at	Date and time project was launched
staff_pick	If the project was a staff pick
backers_count	Number of backers
static_usd_rate	The conversion rate of project country currency to USD
usd_pledged	Amount pledged in USD
category	Category of project
spotlight	If the project was featured on kickstarter spotlight page
name_len	Length of project name in word count
name_len_clean	Length of project name in word count sans non- key words (such as "for" "and" etc.)
blurb_len	Length of project blurb in word count
blurb_len_clean	Length of project blurb in word count sans non- key words
deadline_weekday	Weekday of deadline date

state_changed_at_weekday	Weekday of state change
created_at_Weekday	Weekday of creation date
launched_at_weekday	Weekday of launch date
deadline_month	Month of the project deadline
deadline_day	Day of the project deadline
deadline_yr	Year of the project deadline
deadline_hr	Hour of project deadline
state_changed_at_month	Month of latest state change
state_changed_at_day	Day of latest state change
state_changed_at_yr	Year of latest state change
state_changed_at_hr	Hour of latest state change
created_at_month	Month of creation date
created_at_day	Day of creation change
created_at_yr	Year of creation change
created_at_hr	Hour of creation change
launched_at_month	Month of launch date
launched_at_day	Day of launch date
launched_at_yr	Year of launch date
launched_at_hr	Hour of launch date
create_to_launch_days	Number of days between project creation and the public launch date
launch_to_deadling_days	Number of days between the launch date and the deadline
launch_to_state_change_days	Number of days between launch date to the latest status change

The file is in excel (.xlsx) format. You may get encoding errors when you try to convert it to .csv format. You can import the file directly without converting by using **read_excel** function (instead of **read_csv** that we typically use in class).