Assignment 3

Introduction

In this assignment, you will be using the provided Car Insurance dataset and the machine learning algorithms you have learned in this course in order to :

1. Analyse the data and create a report to to indicate how you build and fine-tuned your models
2. Predict the *age of policy holder*
3. Predict *if a policy holder will lodge a claim*

Here is a video walkthrough for this assignment which can also be found in MS Teams -> General Channel -> Class Material folder.

Datasets

In this assignment, you will be given two datasets train.csv and test.csv . It is part of the assignment that you dig into each column and try to understand how you can use them to achieve the assignment goal. **Here is the column descriptions:**

Variable	Description
policy_id	Unique identifier of the policyholder
policy_tenure	Time period of the policy
age_of_car	Normalized age of the car in years
age_of_policyholder	Normalized age of policyholder in years
area_cluster	Area cluster of the policyholder
population density	Population density of the city (Policyholder City)
make	Encoded Manufacturer/company of the car
segment	Segment of the car (A/ B1/ B2/ C1/ C2)
model	Encoded name of the car
fuel_type	Type of fuel used by the car
max_torque	Maximum Torque generated by the car (Nm@rpm)
max_power	Maximum Power generated by the car (bhp@rpm)
engine_type	Type of engine used in the car
airbags	Number of airbags installed in the car
is_esc	Boolean flag indicating whether Electronic Stability Control (ESC) is present in the car or not.
is_adjustable_steering	Boolean flag indicating whether the steering wheel of the car is adjustable or not.
is_tpms	Boolean flag indicating whether Tyre Pressure Monitoring System (TPMS) is present in the car or not.
is_parking_sensors	Boolean flag indicating whether parking sensors are present in the car or not.
is_parking_camera	Boolean flag indicating whether the parking camera is present in the car or not.
rear_brakes_type	Type of brakes used in the rear of the car
displacement	Engine displacement of the car (cc)
cylinder	Number of cylinders present in the engine of the car
transmission_type	Transmission type of the car
gear_box	Number of gears in the car

steering_type	Type of the power steering present in the car
turning_radius	The space a vehicle needs to make a certain turn (Meters)
length	Length of the car (Millimetre)
width	Width of the car (Millimetre)
height	Height of the car (Millimetre)
gross_weight	The maximum allowable weight of the fully-loaded car, including passengers, cargo and equipment (Kg)
is_front_fog_lights	Boolean flag indicating whether front fog lights are available in the car or not.
is_rear_window_wiper	Boolean flag indicating whether the rear window wiper is available in the car or not.
is_rear_window_washer	Boolean flag indicating whether the rear window washer is available in the car or not.
is_rear_window_defogger	Boolean flag indicating whether rear window defogger is available in the car or not.
is_brake_assist	Boolean flag indicating whether the brake assistance feature is available in the car or not.
is_power_door_lock	Boolean flag indicating whether a power door lock is available in the car or not.
is_central_locking	Boolean flag indicating whether the central locking feature is available in the car or not.
is_power_steering	Boolean flag indicating whether power steering is available in the car or not.
is_driver_seat_height_adjustable	Boolean flag indicating whether the height of the driver seat is adjustable or not.
is_day_night_rear_view_mirror	Boolean flag indicating whether day & night rearview mirror is present in the car or not.
is_ecw	Boolean flag indicating whether Engine Check Warning (ECW) is available in the car or not.
is_speed_alert	Boolean flag indicating whether the speed alert system is available in the car or not.
ncap_rating	Safety rating given by NCAP (out of 5)
is_claim	Outcome: Boolean flag indicating whether the policyholder file a claim in the next 6 months or not.

Please Note:

- Do not trust the column descriptions! check the values yourself, and clean the data if needed. Please check the "train.csv" file for the column names, as there might be spelling errors in the images above; the column names will be the same as what you see in the "train.csv" file.
- You can use the **training** dataset (but not the test) for training machine learning models, and you can use the test dataset to evaluate your solutions to avoid over/under-fitting.
- This assignment specification is deliberately left open to encourage students to submit innovative solutions.
- Your model will be evaluated against a different set of datasets (available for tutors, but not for students)
- You must submit your code (.py) and a report (.pdf) as mentioned later on this page
- You should use the following requirements.txt similar to what you did for Assignment 1; any library outside of this file is forbidden.
- The due date is **22/04/2024 at 20:00**

Part I: Regression

In this part of the assignment you need to predict the "age_of_policyholder". You can use all columns of the dataset except " **age_of_policyholder"** which you are supposed to predict. You can keep your prediction as *a float number* without any penalty.

- The minimum requirement for ***Mean Square Error*** *for this part should be less than* ***95.00*** on the *marking test dataset* ***

- You should analyse and select features that you think would improve your machine learning models (and filter out those that may not). You can also combine multiple features and create new ones.

Part II: Classification

Predict if a policy holder will lodge a claim ("is_claim=1") in the next 6 months. You can use all columns of the dataset except " **is_claim"** which are supposed to predict. In the dataset, "is_claim" column can be either 0 or 1; 1 indicates that in the next 6 months the customer will lodge a claim.

- The minimum requirement for ***Macro Average Precision for this part is 0.52*** on the marking test dataset***. Please note that the distribution of "is_claim" is not uniform. As such, most records have 'is_claim=0'. So it is easy to achieve high accuracy for this problem, but what about " average precision and recall"?

- You should analyze and select features that you think would improve your machine learning models (and filter out those that may not). You can also combine multiple features and create new ones.

****the dataset will not be public and will be used by tutors to test your models. Try not to overfit your machine-learning model on the provided datasets. The marking test dataset will follow the same schema/format of the provided training and test datasets. Please consider missing values or any possible data type issues and do proper exception handling.

Submission

You must submit two files:

- A python script z{id}.py
- A report named z{id}.pdf

Python Script and Expected Output files

Your code must be executed in CSE machines using the following command with three arguments:

```
$ python3 z{id}.py train.tsv test.tsv
```

Your program should create 2 files on the same directory as the script:

- z{id}.PART1.output.csv
- z{id}.PART2.output.csv

For the first part of the assignment: " z{id}.PART1.output.csv " stores the predicted age for all of the records in the test dataset (not the training dataset).

the file should be formatted exactly as (you can keep age as Float or Integer):

```
policy_id,age
P1,25.4
P2,37.5
...
```

For the second part of the assignment: " z{id}.PART2.output.csv " stores the predicted values (is_claim) for the test dataset (not the training dataset) and it should be formatted exactly as follow:

```
policy_id,is_claim
P1,0
P2,1
...
```

Is_claim: 0 means the customer will not lodge a claim, and 1 means the customer will lodge a claim

IMPORTANT: Each line in the output files corresponds to a policy in the test.tsv file (same size and same order). It must have only two columns and the column names should match the above sample outputs.

Marking Criteria

You will be marked based on:

- **(12 marks - 6 marks for each section)**
  - Your code must run and perform the designated tasks on CSE machines without problems and create the expected files correctly, otherwise you will lose marks for the section. Please make sure your output files have the same number of lines as in the "test.csv" file.
  - Your submission will also be marked based on how well it performs on the test dataset (a different dataset not available for students) based on the following metrics: Mean Square Error (Part I), Micro Avg Precision (Part II).
    - 6 marks if your model is in the top 10%.
    - 5 marks if your model is in the top 20%.
    - 4 marks if your model is in the top 50%.
    - 3 marks if your model is in the top 80%.
    - 2 Marks if you only pass the baseline
    - 0 Mark for invalid output files, or not passing the baseline
- **(3 marks)** A report
  You should provide a report, containing 3 sections (max 3 pages). Each section should highlight the most important information in a few bulletpoints (along with statistics and charts if applicable). Here are the three sections you should have in the report (1 mark per section):
    - Discuss any pre and post processing you completed in the dataset;

    - Discuss why you chose particular machine learning algorithms and tuned them
    - Discuss how your solution provides business value to an organisation working in the insurance industry
- The late penalty is 5% per day, and submissions after day 5 will not be marked.
- You will be penalized (1 mark per minute) if your models take more than 2 minutes to train and generate output files in CSE lab machines.
- Your assignment will not be marked (i.e., you will receive zero mark for Assign3) if any of the following occur:
    - If it generates hard-coded predictions
    - If it also uses the second dataset (test) to train the model
    - If it does not run on CSE machines with the given command (e.g., python3 zid.py train.tsv test.tsv)
      Do NOT hard-code the dataset names
    - If it does not create the expected output files (matching the same)
    - You will get zero mark if you merge/concat the train and test datasets in your code . Especially if you are using oversampling techniques this will considered as a CHEATING (zero mark) because in oversampling if you add the test dataframe it will generate samples from test data which is used in training; and it means testing data and training data are the same. In order to avoid this we suggest you use a function like ***def preprocess(df) -> df*** , and pass both dataframe one by one like ***train_df = preprocess(train_df), test_df = preprocess(test_df)***

FAQ

- **Can we define our own feature set?**
  Yes, you can define any features; make sure your features do not rely on the test datasets. You can add/modify/remove any columns from both datasets, but you should not touch or use the column you are predicting to create another column. You can remove any rows of the training dataset, but no rows from the test dataset. DO NOT REMOVE ANY ROWS OF THE TEST DATA SET; THIS will be considered as A KIND OF CHEATING AND WILL RESULT IN ZERO MARK.
- **Will I get penalized for "Warnings" thrown by my code?**
  No, you will not get penalized

Plagiarism

This is an *individual assignment* . The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such offense may include negative marks, automatic failure of the course, and possibly other academic disciplines. Assignment submissions will be checked using plagiarism detection tools for both code and the report and then the submission will be examined manually.

Do not provide or show your assignment work to any other person - apart from the teaching staff of this course. If you knowingly provide or show your assignment work to another person for any reason, and work derived from it is submitted, you may be penalized, even if the work was submitted

without your knowledge or consent. Pay attention to that is **also your duty to protect your code artifacts** . if you are using an online solution to store your code artifacts (e.g., GitHub) then make sure to keep the repository private and do not share access to anyone.

Reminder: Plagiarism is defined as using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several online sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity
- UNSW Plagiarism Procedure

Make sure that you read and understand this. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible for ensuring that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using one (e.g., Github and similar). Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW.