Assignment 3 Report

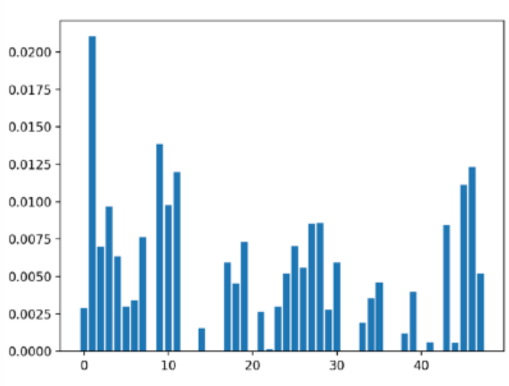
COMP9321 Data Services Engineering – Machine Learning

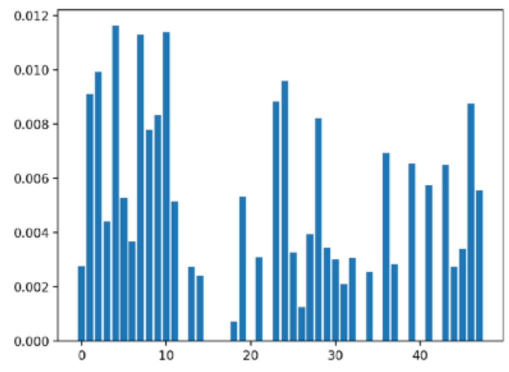
Part I – Pre and Post Processing in the Dataset

A. Data Cleansing

- Sanitise data: ensure all values in correct format by checking *dtype*, and columns are lowercase with underscores instead of spaces
- Remove *policy_id* column from the data frame since we are depending on this for output and not for any correlation
- Convert age of car to months only
- Separate *max_torque* and *max_power* into two columns for each unit: *torque (Nm) / power (bhp)* and *rpm*.
- Change categorical values to numerical values by creating a nested dict which acts as a catalogue mapping categorical to numerical values, and replacing all strings with integers
  - Note that *is_claim* is already in this format (1 = yes, 0 = no)

B. Feature Selection





Method 1: SelectKBest

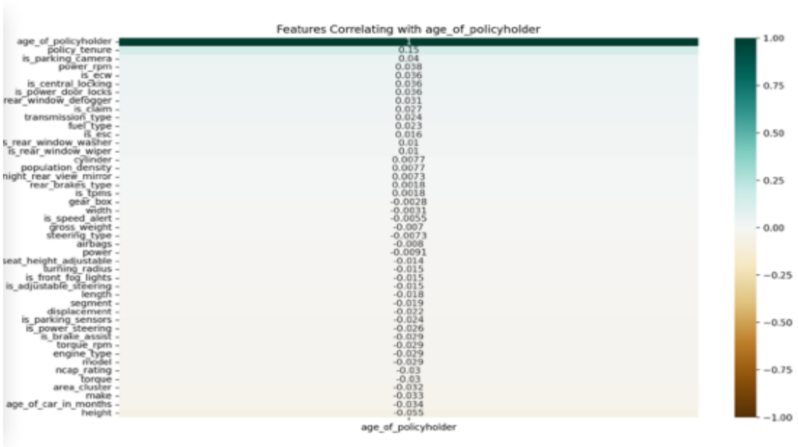
- Uses *SelectKBest* library from *sklearn.feature_selection* to select the best features
- Each execution of the function can be plotted on a bar chart (see images on the right: top = *age_of_policyholder*, bottom = *is_claim*)
- Is inconsistent, generates different features for each call
  - Execute the function multiple times and average the top calls in order to find the best features
  - By performing this strategy, we generate more reliable results than method 2

Method 2: Correlation Heatmap

- Creates the correlation data frame and plots this on a heat map using *seaborn*
- Is much more consistent, generates the same features for each call
- However is much more unreliable, features with highest correlation did not reflect better results when fitted on the models. Potentially misleading data
- See next page for heat map visualisation

C. Outlier Management

- Reallocated outliers to the mean of the column values
- Filled *NaN* values with the mean of the column values
- Had only a slight effect on increasing F1-score (classification), but a much larger effect on reducing mean squared error (regression)



Part II — Choice of Machine Learning Algorithms and Tuning

A. Model Selection and Parameter Tuning

- For regression I put these models into a list and used a for loop to fit the data, calculated the mean square error and compared all models
  - The model which generated the lowest MSE was what I selected in the end
  - *GradientBoostingRegressor()* yielded the lowest MSE out of all models
  - Tested tuning parameters:
    - *learning_rate*: most optimal value is 0.05
    - *n_estimators*: most optimal value is 92
    - All other parameters had their default values being the most optimal
- Classification was the same idea, except for calculating F1-score instead
  - F1-scores were cross-validated to identify instances of overfitting
  - The model which generated the highest F1-score was what I selected in the end
  - GradientBoostingClassifier() yielded the highest F1-score out of all models
  - Tested tuning parameters:
    - *learning_rate*: most optimal value is 0.12
    - *n_estimators*: most optimal value is 131
    - All other parameters had their default values being the most optimal
  - Note that *RidgeClassifier()* yielded the lowest F1 score out of all models, but also had the least amount of overfitting (since F1 scores between test data and cross-validation was very close)

B. Resampling

- Reduces class imbalance for classification
- Before classification, I oversampled using SMOTE:
  - Oversampling increases the number of samples in the minority class to match the majority class, or the ratio provided in sampling_strategy
  - SMOTE generates based on existing data to reduce the risk of overfitting
- Increased the value of the F1-score substantially

C. Cross-Validation

- Designed to prevent overfitting in classification and ensure model's performance is consistent
- Used *KFold* cross validation. Value k=10 is a good baseline value for situations where choosing an appropriate value of k is hard (due to imbalance of classes)
- There exists an F1-score / overfitting trade-off between classification models. Some models yield higher F1-scores but risk severe overfitting. Others yield much lower F1-scores which don't even pass the benchmark, but reduce overfitting substantially
  - Function prioritises higher F1-scores to pass benchmark over overfitting

D. Standardisation

- Standardisation helps eliminate skewness in data by transforming numerical data to within a restricted range (e.g. [-1, 1])
- Attempts to standardise training dataset unfortunately made MSE and F1-scores drastically worse for regression and classification respectively

Part III — Business Value to Organisations in Insurance Industry

- Businesses need to communicate updates to stakeholders to maintain investments, profits and environmental and social governance
  - Internal stakeholders include: employees, contractors, board of directors, data analysts, managers of property and equipment
  - External stakeholders include: customers, suppliers, investors with part-ownership in company, the Government, environmental and social activists
- The more information and insights a business receives, the more the business can refine its decisions and perform strategies in the future to increase profits and corporate responsibility, which maintains positive relations with external and internal stakeholders
- Businesses nowadays rely on well-managed information systems to perform their operations, especially when dealing with personal information from clients. The insurance industry is no exception from this, as more and more people rely on insurance to cover unexpected costs
- This project uses machine learning to help generate precise and timely insights for companies to make more informed decisions, especially for the following purposes:
  - Marketing: insights from machine learning help the company to identifying and targeting particular market groups (e.g. by age), to increase the likelihood of policyholders that will lodge a claim
  - Risk: insurance companies prioritise minimising risk in their industry, and machine learning in data analytics reduces risk of various errors, making results more reliable and precise
  - Human Resources and Time: machine learning is more time efficient, and removes the need for humans to manually calculate scores for every single data entry (of which there are millions). This is also useful since companies require insights to be delivered by specific deadlines and on demand
  - Integrations: machine learning tools can integrate with other systems such as openAI and GitHub, which vastly increases the value information systems can deliver for business analytics. They can also be implemented in RESTful APIs.