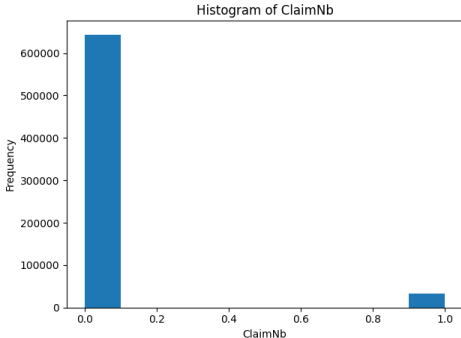
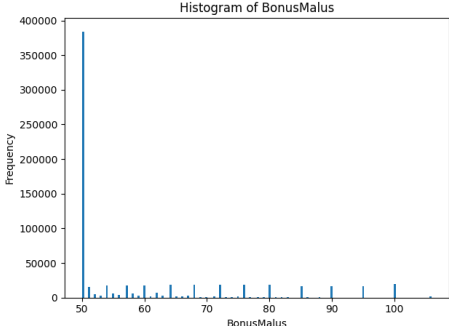
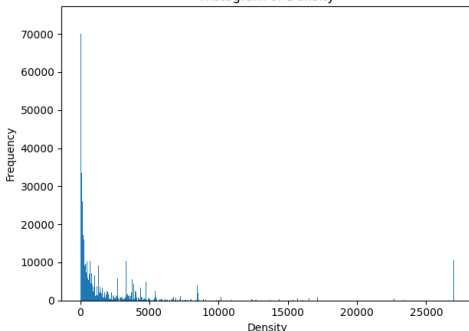


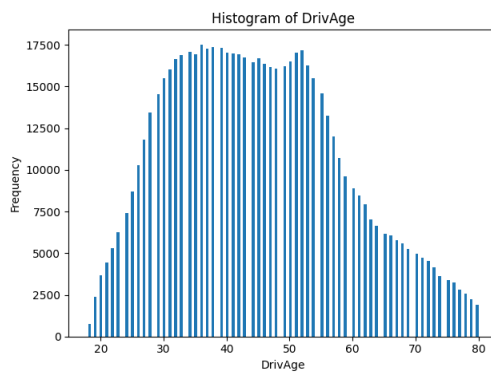
# Results & Findings

## Exploratory Data Analysis (EDA)

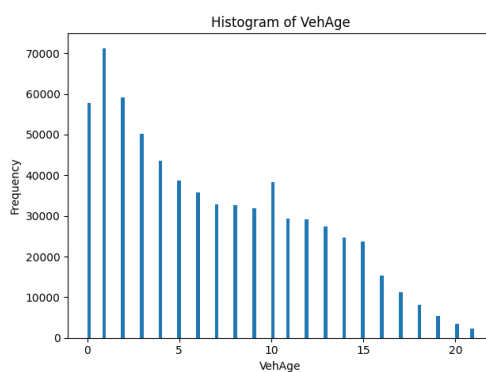
### Distribution of target

In total there are 658k entries with no missing entries. The first dataset has more rows than the second one, making an aggregation of the laims necessary before merging the two datasets to arrive at the target variable

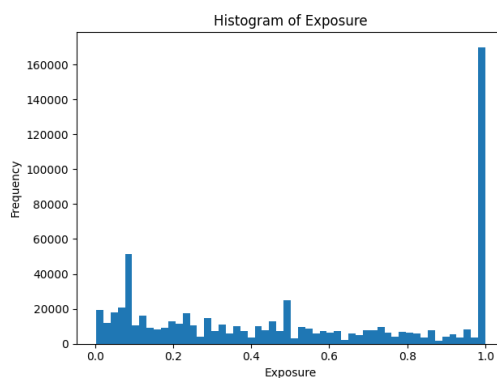
Plot	Explanation
 <p>A histogram titled 'Histogram of ClaimNb'. The x-axis is labeled 'ClaimNb' and ranges from 0.0 to 1.0 with major ticks every 0.2. The y-axis is labeled 'Frequency' and ranges from 0 to 600,000 with major ticks every 100,000. The distribution is highly skewed to the right, with a very tall bar at 0.0 (frequency ~650,000) and a much smaller bar at 1.0 (frequency ~50,000).</p>	<p>Number of claims within the insurance period of 12 months tends naturally to be biased towards zero, as on average, accidents do not occur that often.</p>
Plot	Explanation
 <p>A histogram titled 'Histogram of BonusMalus'. The x-axis is labeled 'BonusMalus' and ranges from 50 to 100 with major ticks every 10. The y-axis is labeled 'Frequency' and ranges from 0 to 400,000 with major ticks every 50,000. The distribution is highly skewed to the right, with a very tall bar at 50 (frequency ~380,000) and many smaller bars extending up to 100.</p>	<p>Average is 59 which is in accordance with the number of claims. as a low number such as 50 indicates no or few claims.</p>
Plot	Explanation
 <p>A histogram titled 'Histogram of Density'. The x-axis is labeled 'Density' and ranges from 0 to 25,000 with major ticks every 5,000. The y-axis is labeled 'Frequency' and ranges from 0 to 70,000 with major ticks every 10,000. The distribution is highly skewed to the right, with a very tall bar at 0 (frequency ~70,000) and many smaller bars extending up to 25,000.</p>	<p>Number of inhabitants per square kilometer in the location of the insured. Heavily biased towards a low density already indicating a bias or attractiveness of the products of the insurance towards a certain demographic</p>

**Plot****Explanation**

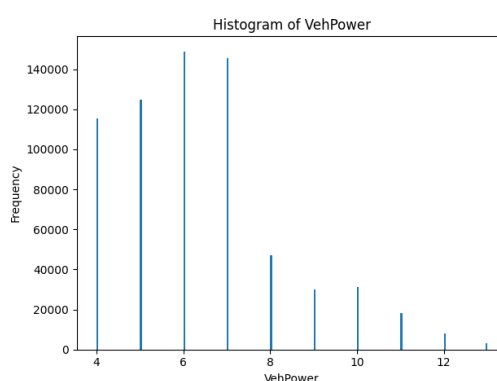
The drivers age is normally distributed with the average age being 45 years

**Plot****Explanation**

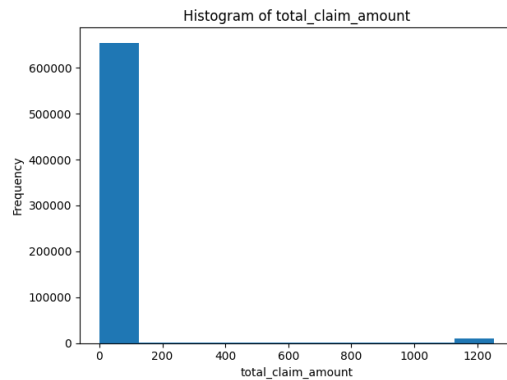
The vehicles age is also skewed to the left indicating that most drivers holding a car insurance have a relatively new car

**Plot****Explanation**

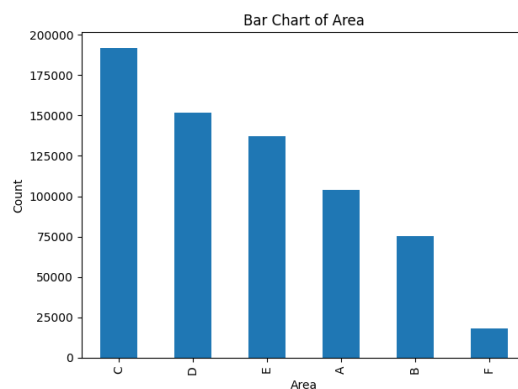
The duration of holding the insurance in years is on average 0.5 meaning half a year though the data is skewed towards one meaning that a lot of customers hold a policy for a year

**Plot****Explanation**

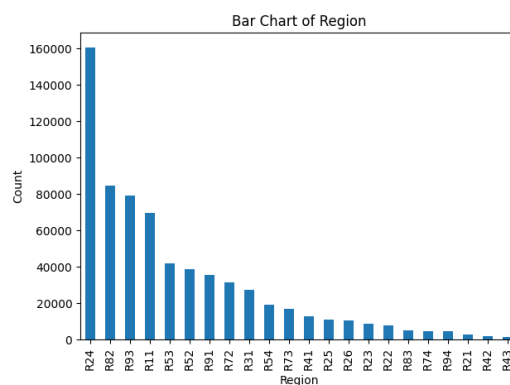
TThe vehicle power is also skewed towards the left meaning most insured cars do not possess a lot of horse power and potentially bear less of a risk which is reflected in the low number of claims

**Plot****Explanation**

The claim amount is also near zero with an average of XX this is as expected for insurance data, as the business case is around the fact of having little claims

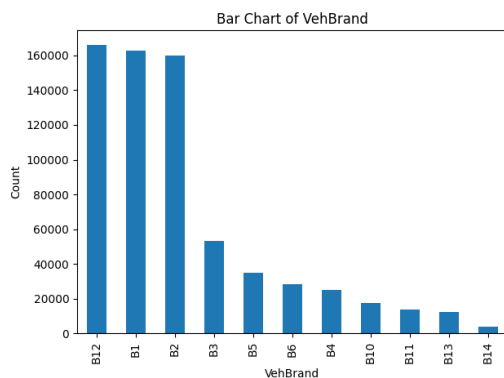
**Plot****Explanation**

The majority of insured come from area C (200k) and the least from F (25k)

**Plot****Explanation**

160k of all insured come from region R24. R24, R82, R93 and R11 account for the majority of regions. I think about 90% of all insured come from these areas

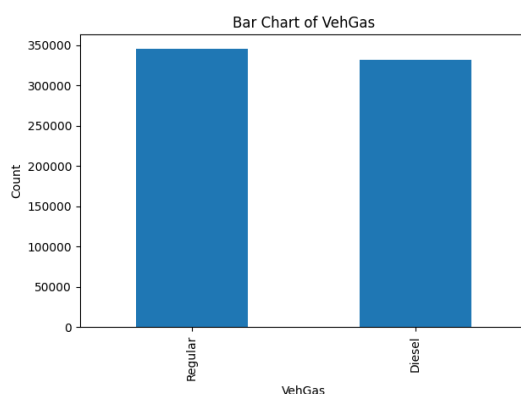
**Plot****Explanation**



The vast majority of cars around 160k each come from VehBrans B1, B2, B12. B3 has around 55k

### Plot

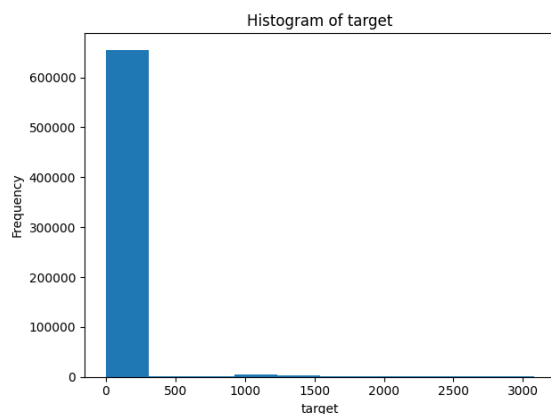
### Explanation



The split between regular and diesel is approx 50:50 which is expected and reflects the global market of types of gas used by vehicles

### Plot

### Explanation

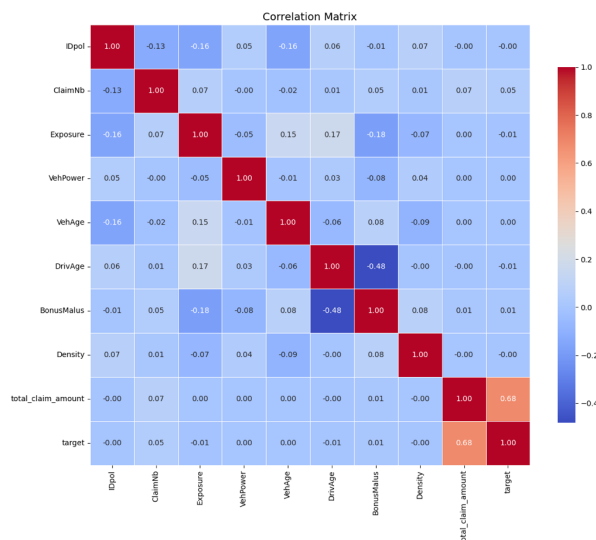


As the target value is defined as the amount claimed per insured divided by the exposure a heavily left skew is expected.

Overall it can be said that nearly all data is biased apart from the drivers age. This was done by comparing average and mean of all features where possible. There is also a lot of categorical data which will result in a large dataset when training the model, which makes further investigation using PCA necessary. It will also cause issues for the machine learning model as skewness will bias the model towards a target value of zero. Due to this imbalance in data it is expected that the models will not perform great in general

### Plot

### Explanation



As the target value is defined as the amount claimed per insured divided by the exposure a heavily left skew is expected.

The correlation matrix reveals a strong positive correlation ( $\approx 0.68$ ) between total\_claim\_amount and target, which is expected since the target variable is defined as the total claim amount divided by the exposure. Interestingly, there's a moderate negative correlation ( $-0.48$ ) between DrivAge (driver's age) and BonusMalus, indicating that younger drivers tend to have higher Bonus-Malus values (i.e., less favorable insurance ratings or higher premiums). This aligns with real-world insurance practice, where less experienced or younger drivers are typically considered higher risk. Aside from this, most features show low or negligible pairwise correlation, suggesting that there is little multicollinearity in the dataset. This can be advantageous for machine learning models, as it implies that most features carry distinct information and thus should be retained unless domain-specific knowledge or feature selection suggests otherwise.

## Feature Engineering

Feature engineering of the categorical values was done through one hot encoding. This increases the number of training features drastically (high dimensionality). That is why I decided to additionally conduct PCA to potentially optimise the number of features for a more accurate modelling of the problem in machine learning. For this I used feature scaling using the standard scaler and was necessary because PCA is sensitive to feature scaling. The results on the PCA can be seen below

Component	Explained Variance
PC1	0.055289892
PC2	0.042729385
PC3	0.037880697
PC4	0.032881772

PC5

0.032087697

Each principal component explains only a small portion of the total variance in the data. Even after 5 components only about 20% of the total variance ( $5.5 + 4.3 + 3.8 + 3.3 + 3.2$ ) is explained. That implies the data is not strongly reducible — no low-dimensional linear subspace captures most of the variance. As a result, applying PCA here may discard useful information, which can explain why your model performance dropped when using PCA features.

## Model Training Summary

### Model Choice

For the modelling task, I explored several approaches: Random Forest, XGBoost, and a Neural Network. While I also briefly considered a Generalised Linear Model (GLM) — a standard method in insurance contexts — I found it challenging to tune effectively for this dataset and ultimately excluded it from further analysis due to suboptimal performance.

- **Random Forest:** A robust ensemble method capable of capturing non-linear relationships in the data. It tends to perform well with limited parameter tuning and provides feature importance insights.
- **XGBoost:** A gradient boosting technique that often yields higher accuracy than Random Forest, especially for complex non-linear problems. It is also considerably faster due to its efficient implementation. And I used it to improve the results obtained with Random Forest.
- **Neural Network:** Particularly suited for modelling highly non-linear interactions. While it requires more tuning and computational resources, it has the potential to learn intricate patterns in the data.

Each model was evaluated on both the full set of features and a reduced feature set derived from PCA, with corresponding performance metrics assessed and documented. I quickly realised that training with selected features did not yield good results which is why I removed the code at some stage.

### Metrics ( $R^2$ , MAE, Loss)

To evaluate model performance, I primarily focused on the  **$R^2$  score (coefficient of determination)**. This metric indicates how well the model's predictions explain the variance in the target variable. An  $R^2$  of 1.0 means perfect predictions, while values close to 0 (or negative) suggest the model isn't capturing the underlying structure of the data.

$R^2$  is particularly helpful in regression tasks like this one, where the goal is to predict a continuous target variable. It offers an intuitive understanding of model quality — essentially answering how much better is the model than simply predicting the mean.

Although additional metrics like **Mean Absolute Error (MAE)** were also calculated during training, the focus remained on maximising  $R^2$ . MAE, which measures the average magnitude of prediction errors, is useful for interpretability (it's in the same units as the target), but  $R^2$  was better suited for comparing models and assessing generalisation performance in this context.

## Hyperparameter Tuning

Hyperparameter tuning was done for all models. A summary of the best model parameters for each machine learning models can be found below.

### Random Forest

#### Metrics

Metric	Train	Validation
MAE	108.20	144.61
$R^2$ Score	0.6808	0.5156

#### Best Parameters

- max\_depth : 10
- min\_samples\_leaf : 2
- min\_samples\_split : 5
- n\_estimators : 50
- random\_state : 42

### XGBoost

#### Metrics

Metric	Train	Validation
MAE	164.81	281.69
$R^2$ Score	0.9890	0.5240

#### Best Parameters

- max\_depth : 3
- n\_estimators : 200
- learning\_rate : 0.1
- subsample : 1.0
- colsample\_bytree : 1.0
- random\_state : 42

### Neural Network

#### Metrics

Metric	Train	Validation
MAE	295.40	381.16
$R^2$ Score	0.6089	0.5267

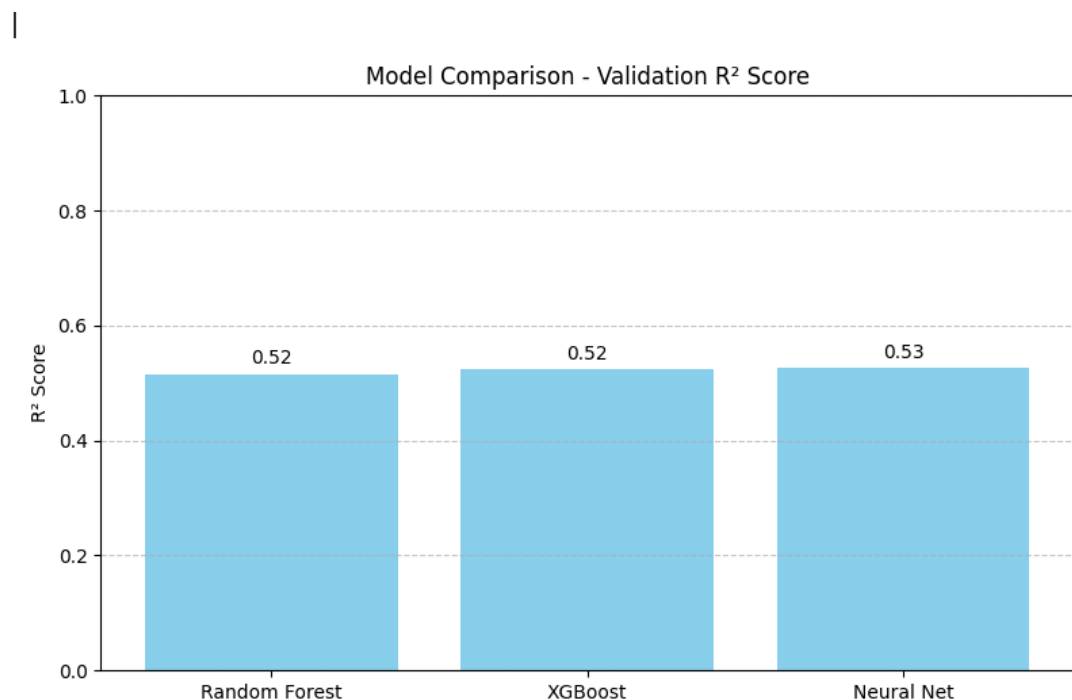
#### Architecture

- Layers: 128 → 64 → 32 → 16 → 8 → 4 → 1
- Dropout: 0.1 after each hidden layer
- Optimizer: Adam

MSE Loss 572M 437M

• Learning rate: 0.0003

### Plot Explanation



| All models reach similar performance, plateauing around an R² of 0.52–0.53. This\_

### 🔍 Observations

- **PCA-based Feature Selection:** Applying PCA led to a significant drop in model performance. While PCA is often used for dimensionality reduction, in this case it likely discarded predictive signals—especially in a dataset with many categorical features and skewed distributions. The top PCA components only explained a small portion of the variance (~5% per component), making PCA unsuitable for this task.
- **XGBoost Tendency to Overfit:** XGBoost achieved a very high training R² (~0.99) but performed only slightly better than the Random Forest on validation data. This indicates overfitting. Although early stopping was used, further hyperparameter tuning (e.g. regularisation terms, smaller trees, or subsample ratios) could help mitigate this effect.
- **Random Forest Stability:** The Random Forest model delivered stable results with a training R² of ~0.68 and a validation R² of ~0.52, indicating a reasonable balance between bias and variance. It's also relatively robust to unscaled and skewed features, which suits the dataset characteristics.
- **Neural Network Performance:** The neural network slightly outperformed



the other models in terms of validation  $R^2$  (approx. 0.53), with a reasonable training  $R^2$  ( $\sim 0.61$ ), showing less overfitting than XGBoost. However, the model requires more complex tuning and is sensitive to scale and architecture. The achieved performance still reflects the limitations of the data.

- **Overall Model Limitations:** Despite hyperparameter tuning, none of the models exceeded a validation  $R^2$  of 0.53. This reflects the underlying difficulty of the problem, particularly due to the **high skewness** of input features and the **target variable**. Since many policyholders have zero claims, the target distribution is heavily imbalanced.
- **Opportunities for Improvement:**
  - Advanced **feature engineering** (e.g. log-transforms, custom risk scores)
  - **Data enrichment** (e.g. external socio-economic data, weather, road conditions)
  - **Segmentation:** Separate models for claimants vs. non-claimants could improve performance due to the zero-inflated nature of the target.
  - **GLMs** and other insurance-specific models were omitted due to limited familiarity but would be worth exploring, especially since they are interpretable and tailored to actuarial tasks.

## Reflections on Model Performance and Limitations

While several models were trained (Random Forest, XGBoost, and Neural Networks), performance across the board remained relatively low, with  $R^2$  peaking at  $\sim 0.53$ . There are several reasons for this:

- **Skewness of Input Features:** Many input variables are heavily skewed, which can challenge models—especially in learning the tails of the distribution. Most policyholders do not submit a claim, making the dataset sparse in high-target cases.
- **Impact on  $R^2$ :**  $R^2$  penalises large errors more heavily. Thus, if a model underpredicts rare, high-value claims, the metric drops significantly—even if it performs well on the majority low-value range. This suggests that the model may perform well on the dense central region but poorly on the business-critical outliers.
- **Choice of Models:** While the selected models are strong general-purpose regressors, domain-specific models like GLMs are most likely better suited for insurance contexts. These were briefly explored but ultimately

omitted due to unfamiliarity.

- **Need for Feature Engineering:** Further improvements could come from more advanced feature engineering, transforming variables to reduce skewness, or even segmenting the modelling task (e.g. claim prediction as a classification + regression hybrid).

In [ ]: