Project Statement

Table of Contents

1. The Problem

Auto insurance industry is witnessing a paradigm shift. Since auto insurance is a homogenous good (difficult to differentiate product A from product B), companies are fighting a price war. On top of that, distribution channel is shifting more from traditional insurance brokers to online purchase. This means that ability for companies to interact through human touch point is limited and customer should be quoted a good price. A good price quote is one which makes customer purchase the policy and helps the company to increase the profits.

Also, the insurance premium is calculated based on more than 50+ parameters. This means that traditional business analytics-based algorithms are now limited in their ability to differentiate among customers based on subtle parameters.

2. Goal

Build a Machine Learning Model to predict whether an owner will initiate an auto insurance claim in the next year.

3. Use Cases

The model shall mainly support the following use cases:

1) **Conquering Market Share:** It should be possible to conquer market share by lowering the prices of the premium for the customers, who are least likely to claim.

2) **Risk Management:** It should be possible to charge right premium from the customer, who is likely to claim insurance in the coming year

3) **Smooth Processing:** It should be possible to reduce the complexity of pricing models, as with majority of transactions happening online and with more customer attributes available (thanks to internet and social media), time is ripe to harness the power of data and build complex ML models

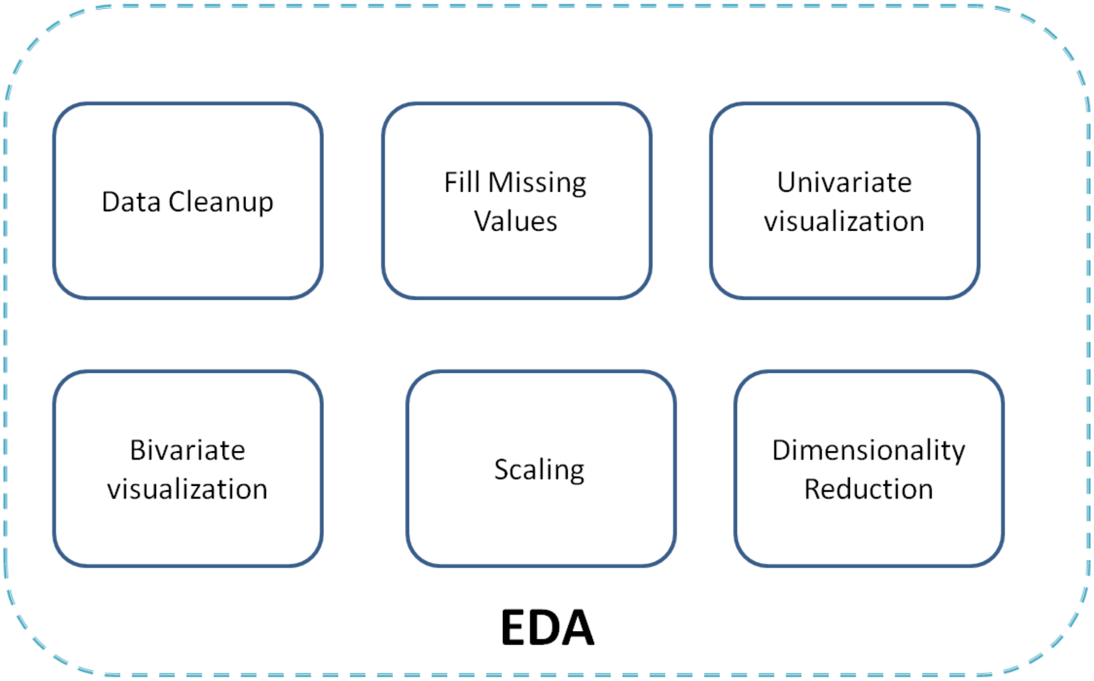
4) **Increased Profits:** As per industry estimate 1% reduction in claim can boost profit by 10%. So, through ML model we can identify and deny the insurance to driver who will make the claim. Thus, ensuring reduced claim out go and increased profit.

Part of the model development is to identify and prioritize above use cases.

4. Process Flow

The ML model shall mainly consist of two phases:

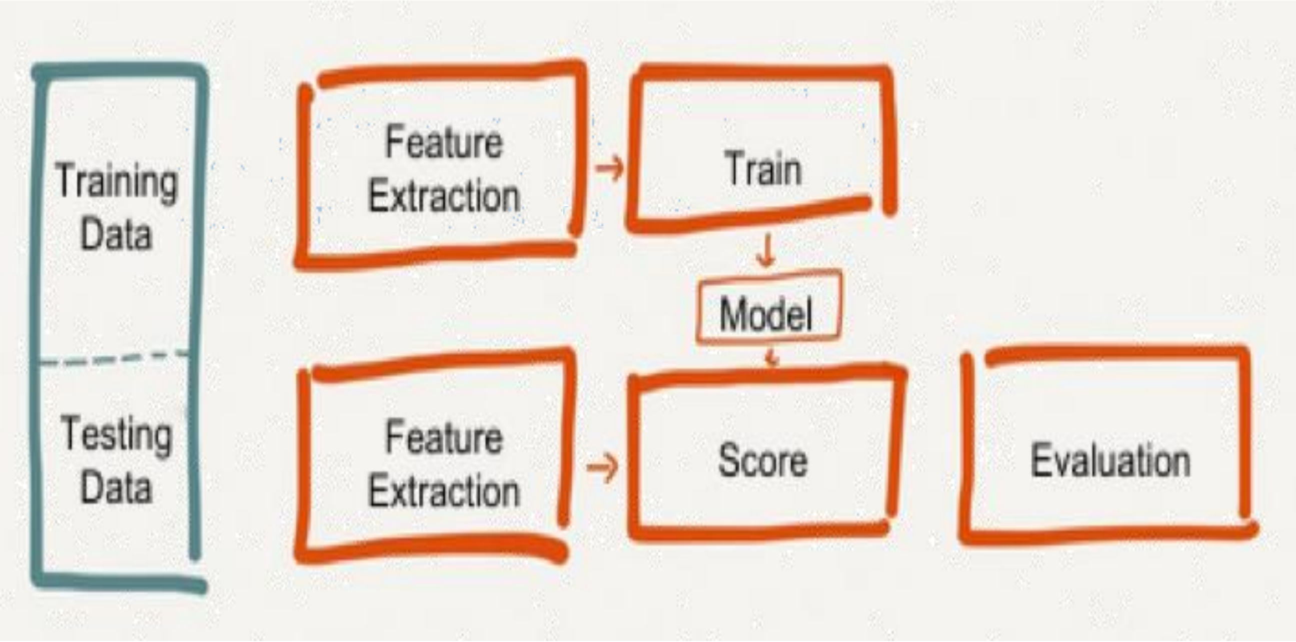
1) **EDA (Exploratory Data Analysis):** This phase will include analysing datasets to summarize their main characteristics, often with visual methods. A statistical model can be used, but primarily EDA will be used for seeing what the data can tell us beyond the formal modelling or hypothesis testing task.
Following tasks can be performed as a part of EDA:
    - Scaling/Normalization – so that entire data is on same scale
    - fill the missing values
    - feature selection & engineering. This will ensure right input to model



2) **Machine Learning Modelling:** After EDA, modelling comes into process. The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The term “ML model” refers to the model artefact that is created by the training process.

   The training data must contain the correct answer, which is known as a target or target attribute. The learning algorithm finds patterns in the training data that maps the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

   You will use the ML model to get predictions on new data for which you will not know the target.



Following tasks can be performed as a part of modelling:

- Start with basic model but eventually move towards ensemble
- Use Deep Learning with sklearn MLPClassifier and check if neural network model is better than traditional models
- Arrival at a model with best f1-score

5. Dataset Dimension

The project involves the use of a dataset with 600k training data and 57 features/data.

6. Target Environment

You will use Edureka's CloudLab, a cloud-based *Jupyter Notebook*, which is pre-installed with Python packages on the cloud-lab environment to work on this Project. It is offered by Edureka as a part of the course, where you can execute all the demos and work on real-life projects in a fluent manner.
You'll be accessing the CloudLab via your browser, which requires minimal hardware configuration.

7. Deliverables

Following are the deliverables (.ipynb files), which needed to be developed with respect to Exploratory Data Analysis:

1. Write at least 3 important inferences from the data above
2. Is the data balanced? Meaning are targets 0 and 1 in right proportion?
3. How many categorical features are there?
4. How many binary features are there?
5. Write inferences from data on Interval variables.
6. Write inferences from data on ordinal variables.
7. Write inferences from data on binary variables.
8. Check if the target data is proportionate or not. Hint: Below than 30% for binary data is sign of imbalance
9. What should be the preferred way in this case to balance the data?
10. How many training records are there after achieving balance of 12 %?
11. Which are the top two features in terms of missing values?
12. In total how many features have missing values?
13. What steps should be taken to handle the missing data?
14. Which interval variables have strong correlation?
15. What's the level of correlation among ordinal features?
16. Implement Hot Encoding for categorical features

17. In nominal and interval features which features are suitable for StandardScaler?
18. Summarize the learnings of ED

Following are the deliverables (.ipynb files), which needed to be developed with respect to Modelling:

1. The Simple LogisticRegression Model seem to have high accuracy. Is that what we need at all? What is the problem with this model?
2. Why do you think f1-score is 0.0?
3. What is the precision and recall score for the model?
4. What is the most important inference you can draw from the result?
5. What is the accuracy score and f1-score for the improved Logistic Regression model?
6. Why do you think f1-score has improved?
7. For model LinearSVC play with parameters – dual, max_iter and see if there is any improvement
8. For -- SVC with Imbalance Check & Feature Optimization & only 100K Records is there improvement in scores?
9. XGBoost is one the better classifiers -- but still f1-score is very low. What could be the reason?
10. What is the increase in number of features after onehotencoding of the data?
11. Is there any improvement in scores after encoding?
12. If not missing a positive sample is the priority which model is best so far?
13. If not marking negative sample as positive is top priority, which model is best so far?
14. Do you think using AdaBoost can give any significant improvement over XGBoost?
15. MLPClassifier is the neural network we are trying. But how to choose the right no. of layers and size
16. At what layer size we get the best f1-score?