2024

Computer Science Capstone

C964

March 4, 2024

C964: Computer Science Capstone

Part A: Letter of Transmittal

March 4, 2024

CEO Office
Zales Auto Insurance
123 Prototype Dr
Reno, Nevada 89433

Dear Management,

As an auto insurance provider, we deal with risks every day in our operations. Being able to respond to those risks is critical to our viability as a company. In this regard, machine learning is an opportunity to adopt a data-driven approach that can bolster our risk management strategy.

This project proposal seeks your approval for a machine learning application that can predict the likelihood of our customers filing auto insurance claims. Being able to accurately forecast future claims will substantially improve our decision making and risk management capabilities. For example, it will allow us to identify high-risk policy holders and enable us to adjust premiums accordingly. This will lead to significant cost reductions and increases in our profitability.

The total costs of this project will be approximately $300,000. It will take 4 months to develop and deploy the application. Existing customer data will be used to train the machine learning algorithm. This data will not contain any personal identifiable information (PII), so there will not be any ethical or privacy concerns that would need to be considered.

My group has the skills and knowledge necessary to develop this machine learning product. As the team lead, I have the project management experience to ensure the project stays on track and does not exceed the budget. Our team members have the relevant skills in software

development and machine learning to develop this application. I look forward from hearing from you soon.

Sincerely,

XYZ

Part B: Project Proposal Plan

Project Summary

As an auto insurance company, our ability to analyze customers and assess their likelihood of filing expensive insurance claims is paramount in minimizing costs and risk. Failure to do so can have a severe impact on our profitability. Unfortunately, our current approach in analytics is ineffective and error prone. To remedy this, our project seeks to adopt a more data-driven approach by utilizing data science and machine learning. This will be done through the development of a tool that can analyze a customer's data and predict whether that customer will file an insurance claim. This tool will greatly enhance our company's decision making and analysis capabilities. For example, we will be able to more effectively approve or deny coverage, raise or lower insurance premiums, and forecast future costs.

The deliverables of this project are the following:

1. Machine learning tool that can accept customer data and predict whether the customer will be likely to file an insurance claim.
2. User guide that documents the procedure to setup and use the machine learning application.

Data Summary

We will use an online dataset to simulate our existing customer data. This dataset will consist of thousands of entries featuring customer and vehicle information, stored in a CSV format. Entries will contain feature variables such as the car value and the target variable which

is whether the customer filed an insurance claim. This information will be critical in training our machine learning model.

Once the data is acquired, a data dictionary will be created to document details such as data types, column names, and other information regarding the dataset. This will be maintained and updated throughout the development life cycle to reflect the current characteristics of the dataset. The data will then be inspected for anomalies and outliers. These will be handled by modifying the data or removing entries altogether. For example, rows with incomplete data will have missing values substituted with the mean value for that column. The data will also have to be further processed in order to make it suitable for machine learning training. This includes splitting the data into testing and training datasets and encoding categorical columns.

There should not be any ethical or legal concerns to worry about with the dataset. This is because it is open-sourced and does not contain any sensitive information such as personal identifiable information or PII.

Implementation

The methodology that will be used in this project will be SEMMA:

- **Sample:** Dataset that simulates our customers auto insurance data will be acquired for training the machine learning model. This data will be retrieved from an online source such as Kaggle. It will be a good representative sample of our existing customer data, as it is smaller and already in a format suitable for our machine learning program.
- **Explore:** Analysis will be performed on the dataset to better understand the data and general patterns, as well as spot any issues such as missing or invalid data. We will look at each variable and determine the general distribution of values. We

will also look at the relationships between variables. This will allow us to have a better understanding of the dataset that we will be working with.

- **Modify:** Dataset will be modified according to our findings in the exploratory phase. Invalid data will be removed from the dataset. Outliers will be modified with values that are more suitable. This will be done by replacing values with the mean value for that column. For example for vehicle value, any outliers will be replaced with the average value of that column.
- **Model:** Dataset will be split into training and testing sets. The machine learning model will be setup and prepared in the program. The training set will be loaded into the model for training.
- **Assess:** Machine learning model will be analyzed for accuracy and performance. This will be done by scoring the by comparing its predictions with the actual outcomes in the testing set.

Timeline

Milestone or deliverable	Duration (days)	Projected start date	Anticipated end date
Initial Setup	15	March 1	March 15
Data Analysis	7	March 16	March 23
Data Preparation	7	March 24	March 31
Model Training	15	April 1	April 15
Model Refinement	15	April 15	April 30
UI Development	15	May 1	May 15
Testing	15	May 16	May 31
Quality Assurance	15	June 1	June 15

Deployment	15	June 16	June 30
Review	7	July 1	July 7

Evaluation Plan

We will utilize several methods to evaluate the project during development. One method will be performing statistical analysis on the data throughout the project. This will ensure the dataset's integrity and quality. For example whenever the data is modified, statistical operations such as counting the number of rows and viewing distribution of data, will be used to verify data integrity. Any anomalies or extreme deviations would be addressed before continuing development.

To validate the machine learning model and assess its performance, the dataset will be split into training and testing sets. These datasets will be divided into two categories, one containing the feature variables such as customer income and the other containing the target variable, which is whether the customer files an auto insurance claim. The machine learning model will be trained using the training set and scored with the testing set. The score will reflect its accuracy in making predicting the correct target variable outcome. A high score will be an indicator that the machine learning model is performing well.

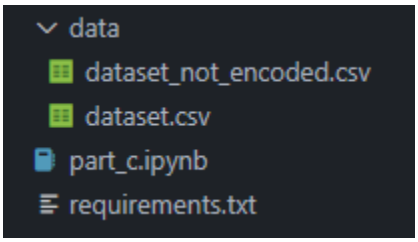
Resources and Costs

Resource	Description	Cost
Developers	Project members with data science and software development expertise	$245,000
Software Resources	Software and tools required for implementing machine learning development	$5,000
Hardware Resources	Hardware required for the system, consisting of computers, servers, etc.	$50,000
	Total	$300,000

Part C: Application

Please see below for the user guide that explains how to use the application online or run it locally. The following is a list of the files submitted for the application.

These files are viewable on GitHub



Part D: Post-implementation Report

Solution Summary

Our company was failing to identify high-risk customers which was leading to increased costs. This was due to the way we analyzed customer data inaccurately. Our project addressed this shortcoming by providing a machine learning tool that would accept customer data and predict the customer's level of risk, which was in the form of their likelihood of filing insurance claims. With this tool, we were able to make better decisions and mitigate costs.

Data Summary

The data was sourced from Kaggle, https://www.kaggle.com/datasets/xiaomengsun/car-insurance-claim-data. This data was downloaded as a CSV file. It was analyzed using Jupyter Notebook and the Pandas library. The analysis performed included inspecting the data for missing values, viewing data types of columns, and calculating the relationships between variables using correlation matrix. After the exploring the data, the data was preprocessed for machine learning. This involved removing entries that had missing elements as well as dropping columns that did not have strong correlation with the target variable. The dataset was also encoded to convert the text datatypes into a numerical format. Once the processing was complete, the dataset was saved into a CSV file as a backup.

Machine Learning

The machine learning model used in this project was the Logistic Regression model. This is a supervised learning method that is well suited to solve binary classification problems. Some examples of this includes predicting whether a customer will churn or whether a patient has a

disease. Logistic regression models work by using sigmoid function to calculate the probability of the target value. In this project, customer data would be entered into the program for the model to analyze and predict whether the customer will file an auto insurance claim.

Development of this project was implemented using a variety of tools and resources. First, data was collected and processed for machine learning using the Pandas python library. It was then split into training and testing sets using the Scikit-learn python package. This package was also used to create and train the logistic regression model using the training dataset. Once the training was completed, the model was evaluated on the testing set for its accuracy. Lastly, the Ipywidgets python library was utilized to create the user interface for inputting customer data and displaying the machine learning results.

One reason logistic regression was chosen for this project was due to its relative simplicity. It does not need to undergo hyperparameter tuning like the KNN algorithm does with its number of neighbors parameter. Along with its ease of use, another reason for choosing logistic regression was its suitability to the problem. This type of machine learning model is commonly used for binary classification problems where the target variable to be predicted consists only of two values. Since our application involves predicting only whether an insurance claim would be filed or not, logistic regression was a good fit.

Validation

The machine learning model was evaluated on its accuracy of making successful predictions. This was done using a testing dataset containing the feature and target variables. The model was scored by comparing its predictions with the actual outcomes in the testing dataset. This scoring was done with the scikit-learn package, specifically using the “score” function. Other metrics

were also used to evaluate the machine learning algorithm. Using scikit-learn's metric module, the model's precision, recall, and f1-score performance was calculated based on its predictions and the actual outcomes in the testing set. The results of all these metrics were above 70%, indicating an acceptable level of performance. Pasted below is an image of these results:

Evaluation

```
score = model.score(X_test, y_test)

print("Accuracy {0:.2f}%".format(score.item() * 100))
```

```
Accuracy 74.15%
```

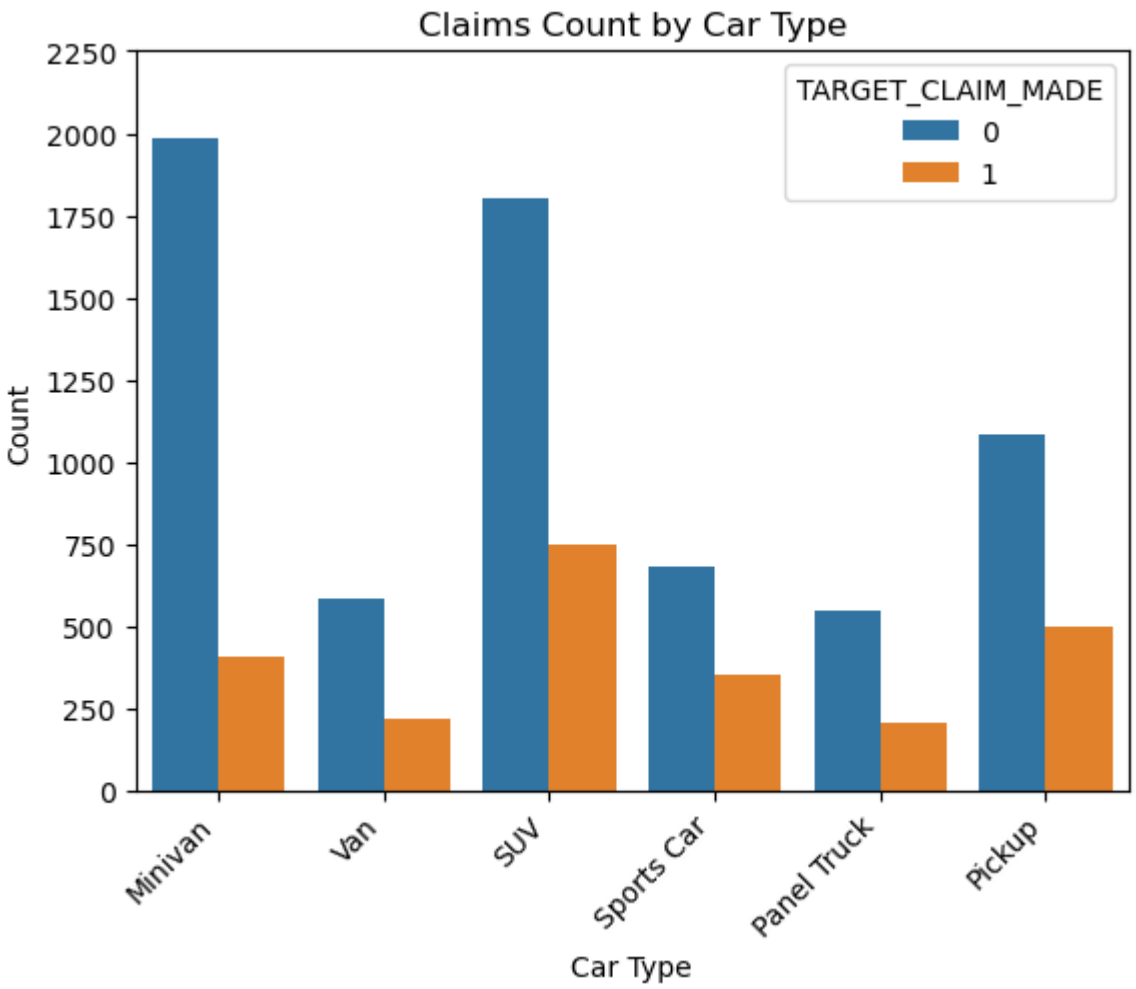
```
from sklearn import metrics
y_pred = model.predict(X_test)
print(metrics.classification_report(y_test,y_pred))
```

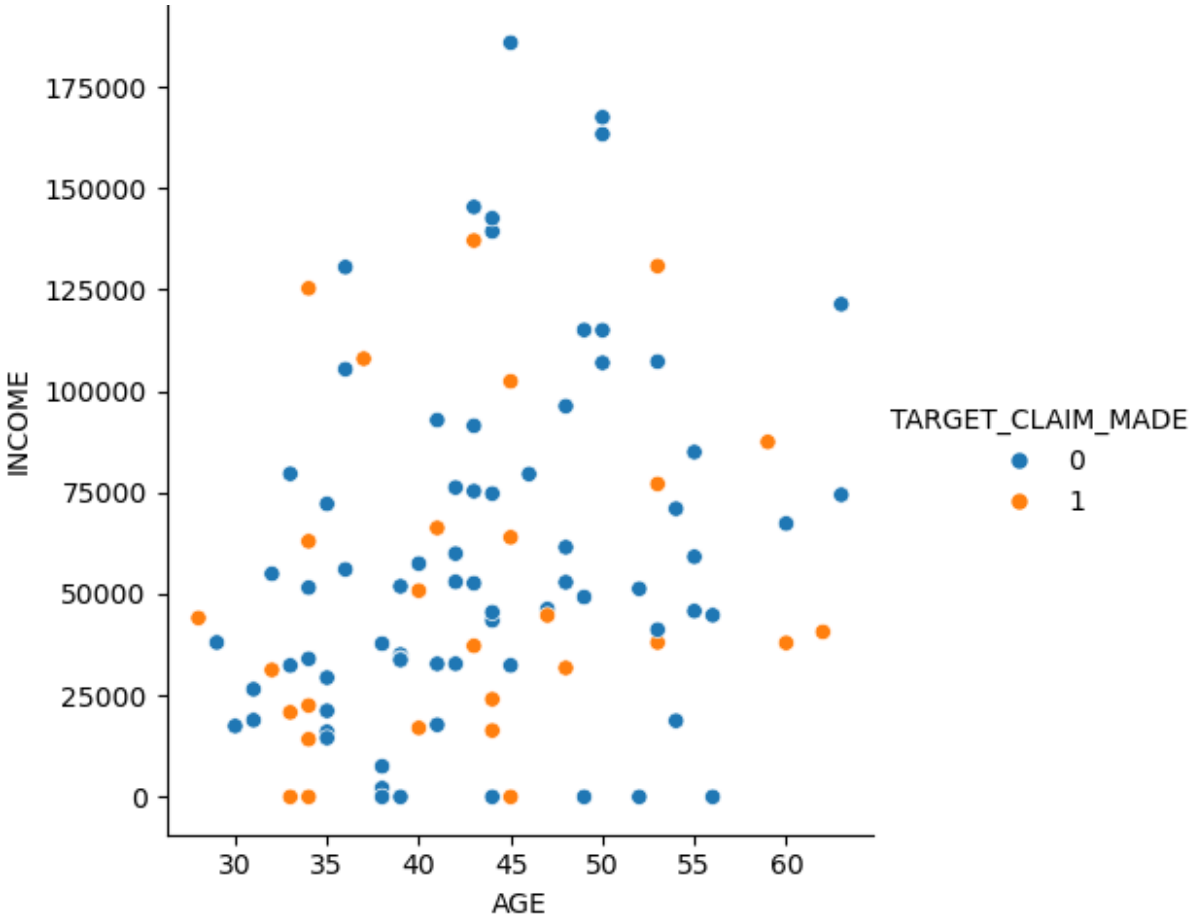
```
              precision    recall  f1-score   support

           0       0.75      0.96      0.85      2007
           1       0.57      0.13      0.22       732

    accuracy                           0.74      2739
   macro avg       0.66      0.55      0.53      2739
weighted avg       0.70      0.74      0.68      2739
```

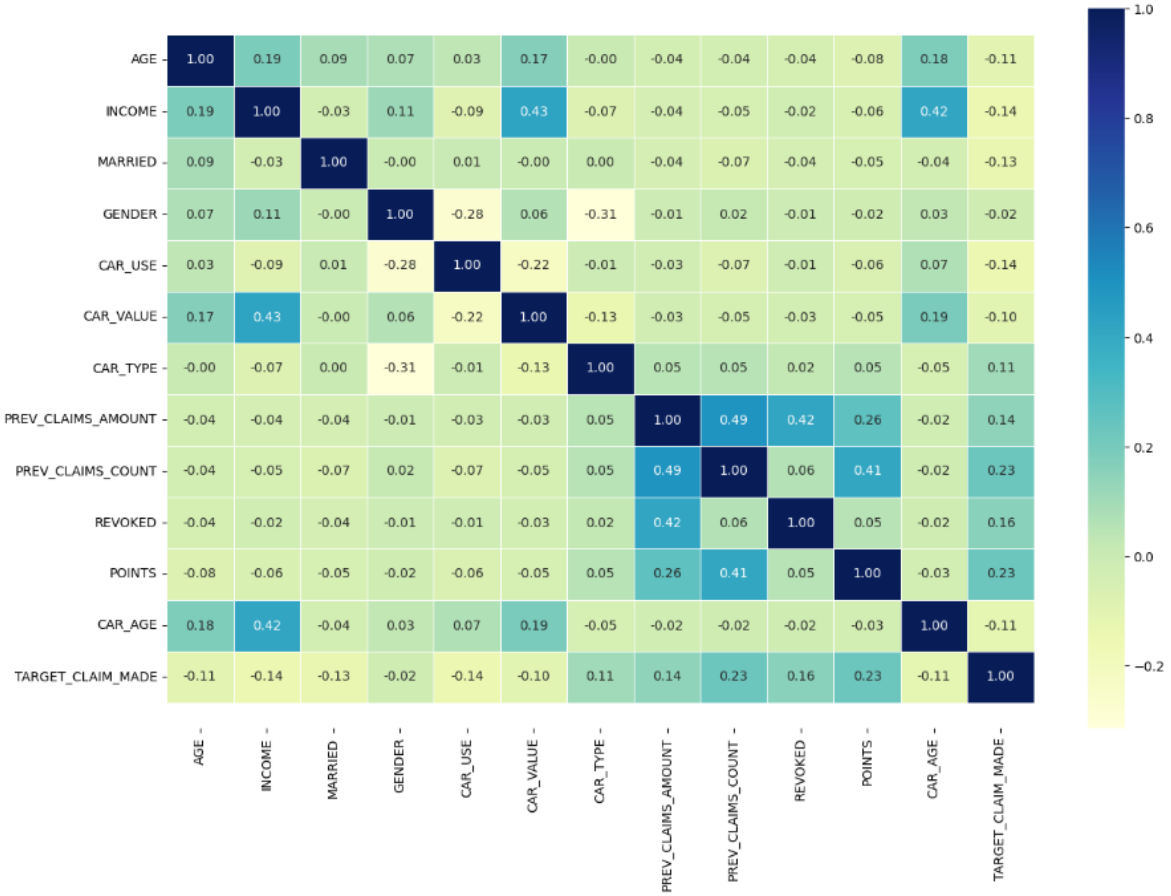
Visualizations

The visualizations can be found in the "part_c.ipynb" jupyter notebook file. Pasted below are the following visualizations:

- Bar chart that shows the frequency of insurance claims
- Heat Map that displays the correlation strength among variables
- Scatter plot that shows the relationship between income, age, and insurance claims


Claims Count by Car Type
TARGET_CLAIM_MADE
0
1
Count
2250
2000
1750
1500
1250
1000
750
500
250
0
Minivan
Van
SUV
Sports Car
Panel Truck
Pickup
Car Type


175000
150000
125000
100000
75000
50000
25000
0
INCOME
30
35
40
45
50
55
60
AGE
TARGET_CLAIM_MADE
0
1

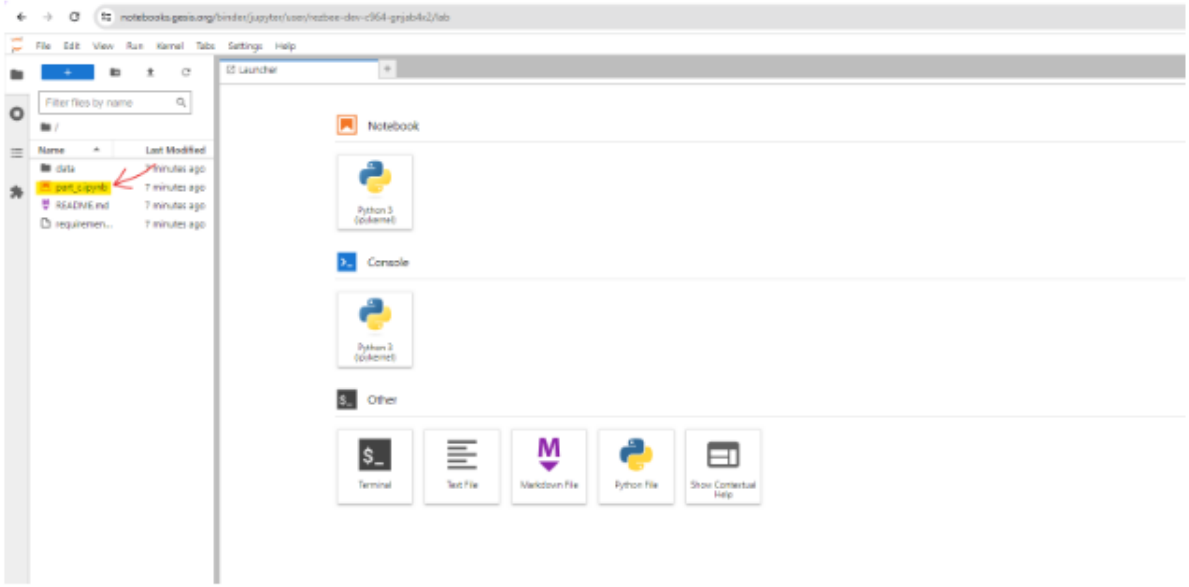

AGE 1.00 0.19 0.09 0.07 0.03 0.17 -0.00 -0.04 -0.04 -0.04 -0.08 0.18 -0.11
INCOME 0.19 1.00 -0.03 0.11 -0.09 0.43 -0.07 -0.04 -0.05 -0.02 -0.06 0.42 -0.14
MARRIED 0.09 -0.03 1.00 -0.00 0.01 -0.00 0.00 -0.04 -0.07 -0.04 -0.05 -0.04 -0.13
GENDER 0.07 0.11 -0.00 1.00 -0.28 0.06 -0.31 -0.01 0.02 -0.01 -0.02 0.03 -0.02
CAR_USE 0.03 -0.09 0.01 -0.28 1.00 -0.22 -0.01 -0.03 -0.07 -0.01 -0.06 0.07 -0.14
CAR_VALUE 0.17 0.43 -0.00 0.06 -0.22 1.00 -0.13 -0.03 -0.05 -0.03 -0.05 0.19 -0.10
CAR_TYPE -0.00 -0.07 0.00 -0.31 -0.01 -0.13 1.00 0.05 0.05 0.02 0.05 -0.05 0.11
PREV_CLAIMS_AMOUNT -0.04 -0.04 -0.04 -0.01 -0.03 -0.03 0.05 1.00 0.49 0.42 0.26 -0.02 0.14
PREV_CLAIMS_COUNT -0.04 -0.05 -0.07 0.02 -0.07 -0.05 0.05 0.49 1.00 0.06 0.41 -0.02 0.23
REVOKED -0.04 -0.02 -0.04 -0.01 -0.01 -0.03 0.02 0.42 0.06 1.00 0.05 -0.02 0.16
POINTS -0.08 -0.06 -0.05 -0.02 -0.06 -0.05 0.05 0.26 0.41 0.05 1.00 -0.03 0.23
CAR_AGE 0.18 0.42 -0.04 0.03 0.07 0.19 -0.05 -0.02 -0.02 -0.02 -0.03 1.00 -0.11
TARGET_CLAIM_MADE -0.11 -0.14 -0.13 -0.02 -0.14 -0.10 0.11 0.14 0.23 0.16 0.23 -0.11 1.00
AGE
INCOME
MARRIED
GENDER
CAR_USE
CAR_VALUE
CAR_TYPE
PREV_CLAIMS_AMOUNT
PREV_CLAIMS_COUNT
REVOKED
POINTS
CAR_AGE
TARGET_CLAIM_MADE
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

User Guide

Links:

- Binder: https://mybinder.org/v2/gh/rezbee-dev/c964/HEAD
- GitHub: https://github.com/rezbee-dev/c964

Running in Binder

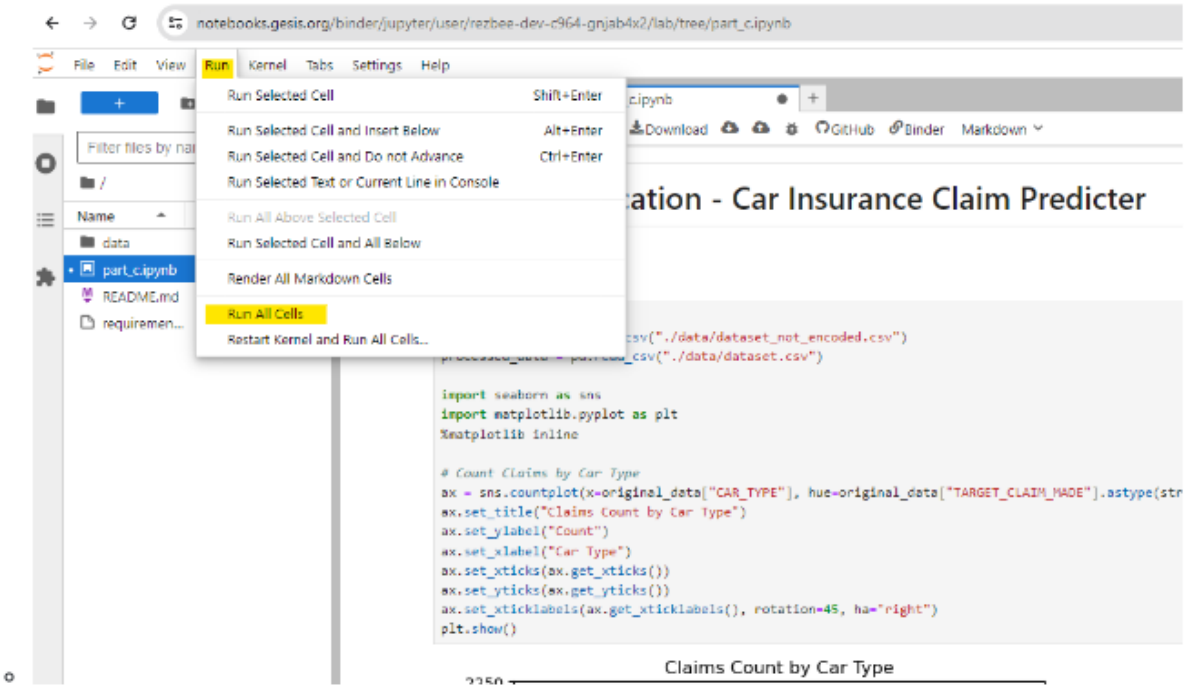
- Navigate to https://mybinder.org/v2/gh/rezbee-dev/c964/HEAD
- Wait for build process to finish (it may take a few minutes)
- Open `part_c.ipynb` file
  - 

Running in local environment

- Install latest version of python
- Download the following files from this repository
  - `requirements.txt`
  - `data/dataset.csv`
  - `data/dataset_not_encoded.csv`
  - `part_c.ipynb`
- Create virtual env folder: `python -m venv .venv`
- Activate virtual env
  - Linux: `source .venv/bin/activate`
  - Windows: `.venv\Scripts\activate`
- Install packages: `pip install -r requirements.txt`
  - Alternatively (if on Windows), run the following
    - `pip install notebook`
    - `pip install ipywidgets`
    - `pip install seaborn`
    - `pip install scikit-learn`
- Start jupyter notebook: `jupyter notebook`
- Navigate to URL notebook is running on

Using the notebook

- Click on `Run > Run all cells`



- Wait for cells to run (it may take a few seconds)
- Scroll to the bottom of the page
- Interact with the application by entering in data, or pressing "Simulate" to prepopulate data
- Click "Predict" to run machine learning model on the data