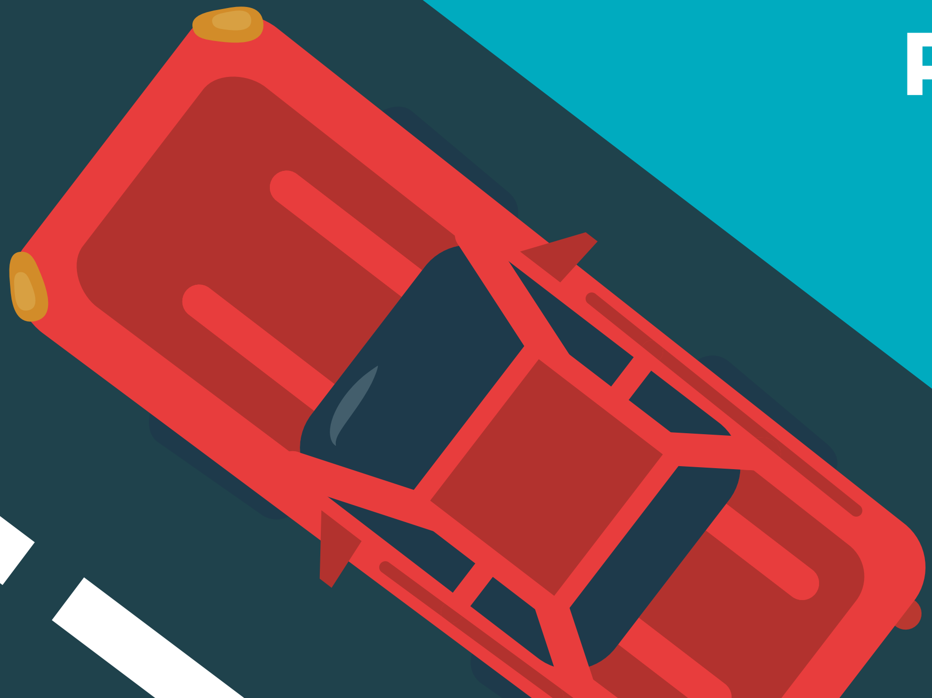
Automobile Insurance Claim Prediction Analysis

Team 6

Wenxuan Yan
Yicheng Jiang
Ziyuan Li
Ziyin Chen



Who?

Help People and Firm Make Better Decisions

1. 2B Clients:

   Insurance Providers

2. 2C Clients:

   Individuals or organizations

Why?

Upward Market Trend

1. Increase Insurance company revenue by matching full coverage sales with high risk clients

2. Helping low risk clients choose the right coverage to save their money

Business Aspect: SWOT

Strengths

Providing ***personalized services*** to both high-risk and low-risk customers. Suggesting the best insurance selection with a ***high model accuracy***. Meanwhile, we are helping our 2B clients to ***promote their products***. The ultimate goal is to build a brand image and focus on market growth throughout the whole value chain.

Weakness

Our ultimate goal is focus on market growth by offering risk advisory service to client. It might has bottleneck in 2C's market growth, since majority potential clients might facing ***high switch costs.***

Opportunities

As ***market revenue*** in auto-insurance keeps growing; insurance firms coming up many new products. We can pattern with upstream suppliers, and help them to market their products in an efficient way. Meanwhile, we help 2C clients to pick the proper product to mitigate risks. Furthermore, there are no many competitors provide such service, and the entry barriers is low.

Threats

Firstly, we include ***sensitive information into our model*** - like race, gender, etc., which can be a potential risk regarding to privacy compliance. When we negotiated with the big insurance companies, they may ***see as a threat*** (especially some agents overestimates the clients' risks and selling high-end products to increase they KPIs).

Data Source & Description

Sagnik Roy

DE Intern @CCD | Former DS Intern @HappyMonk AI | ML Problem Setter @HackerEarth | Former DS Intern @Argoid Analytics | Kaggle Expert | Former GDSC AI/ML Lead

#	Column	Non-Null Count	Dtype
---	------	--------------	-----
0	ID	10000 non-null	int64
1	AGE	10000 non-null	category
2	GENDER	10000 non-null	category
3	RACE	10000 non-null	category
4	DRIVING_EXPERIENCE	10000 non-null	category
5	EDUCATION	10000 non-null	category
6	INCOME	10000 non-null	category
7	CREDIT_SCORE	9018 non-null	float64
8	VEHICLE_OWNERSHIP	10000 non-null	category
9	VEHICLE_YEAR	10000 non-null	category
10	MARRIED	10000 non-null	category
11	CHILDREN	10000 non-null	category
12	POSTAL_CODE	10000 non-null	category
13	ANNUAL_MILEAGE	9043 non-null	category
14	VEHICLE_TYPE	10000 non-null	category
15	SPEEDING_VIOLATIONS	10000 non-null	category
16	DUIS	10000 non-null	category
17	PAST_ACCIDENTS	10000 non-null	category
18	OUTCOME	10000 non-null	category

Structure of Reports

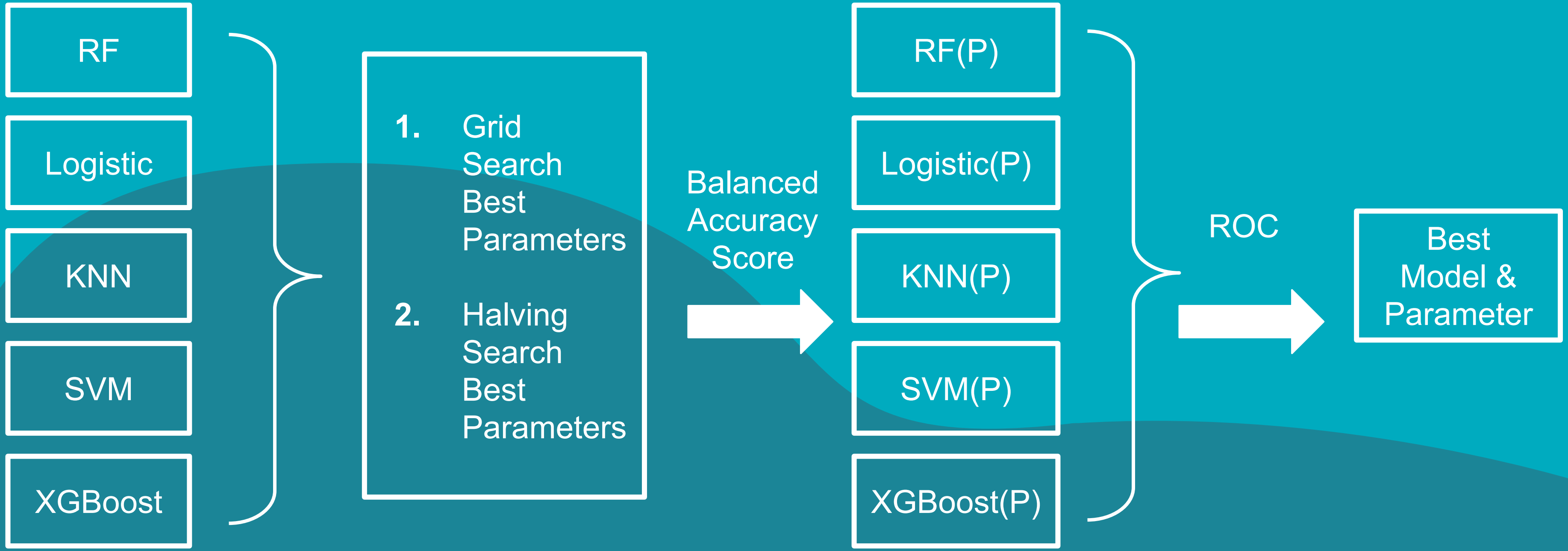
1. Data Exploration
2. Preprocessing Pipeline Building
3. Train Test Split
4. Model Building and Tuning
5. ROC Curve Comparing
6. Best Model Finalizing
7. Real Data Application

1. Random Forest Classifier
2. Logistic Regression Classifier
3. K-nearest Neighbor Classifier
4. Support Vector Machines Classifier
5. XGBoost Classifier

Preprocessing Pipeline

1. Count null values
    a. Two column (‘Credit Score’ & ‘Annual Mileage’) with 1000 null values each
    b. Only 69 overlaid rows → Cannot simply dropna → Apply pipeline
2. Drop ‘ID’ column
3. Separate the train dataset into numerical and categorical
    a. Numerical
        i. Normalize the numerical column with standard scalar and log
        ii. Apply simple imputer and iterative imputer to fill NaN
    b. Categorical - Use OneHotEncoder to fill NaN using most frequent value

Models Building and Tuning



Parameter Tuning

```
y_test.value_counts()

0.0    1342
1.0     658
Name: OUTCOME, dtype: int64
```

	Model	Accuracy_Score	Balanced_Accuracy_Score
0	RandomForest_Grid_Search	0.8405	0.8130
1	RandomForest_Halving_Search	0.8380	0.8096
2	LogisticReg_Halving&Grid	0.8210	0.7876
3	KNN_Halving&Grid	0.8125	0.7662
4	SVM_Grid	0.8360	0.8120
5	SVM_Halving	0.8405	0.8165
6	XGB_Halving&Grid	0.8340	0.8047

This table represents the five sets of models and parameters we will be using for the final ROC evaluation.

Reasons why we excluded Decision Tree Classifier and Random Search

1. Decision Tree
   a. Decision Tree Classifier is a model that requires a lot of training and tuning.
   b. Random Forest Classifier is a resource-efficient analogy to DTC.

2. Random Search
   a. Random Search ONLY selects and tests a random combination of hyperparameter to find the best model
   b. Grid Search looks at EVERY possible combination of hyperparameters in the grid

3. Though we excluded DTC and Random Search, we believe that the wide variety of models we have trained can makeup for that.

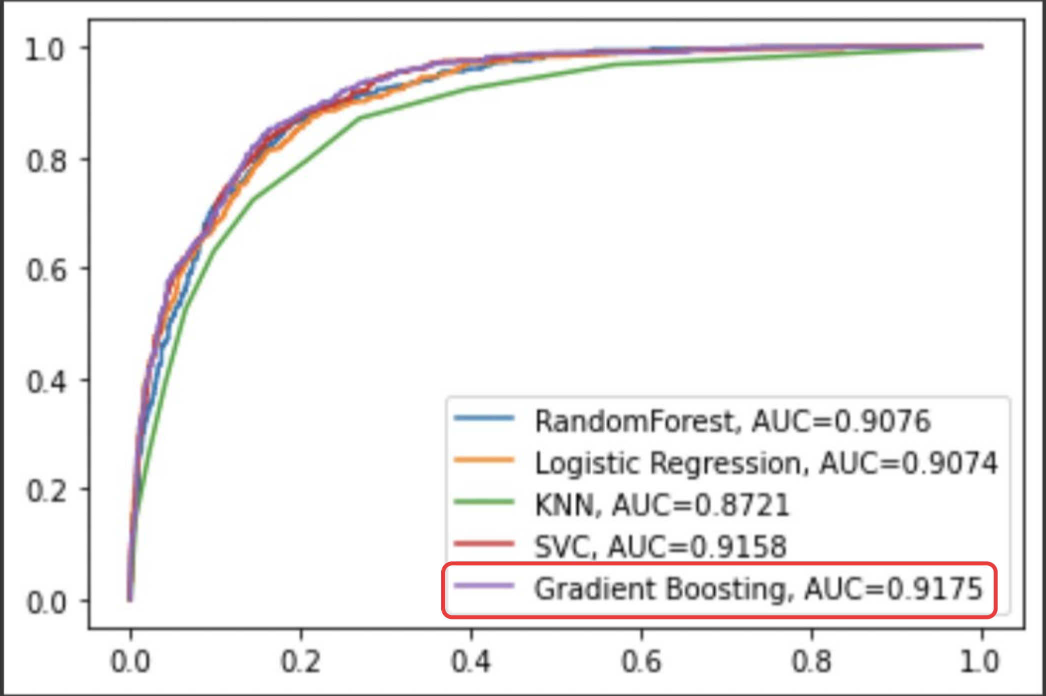
Challenges & Difficulties

- Solvable:
  - Dealing with unbalanced outcome column
  - Choosing which model to fit our dataset
  - Extracting the better parameter
  - Choosing appropriate model performance measurement

- Didn’t Manage to Improve or Not Solvable
  - Linearity in coding
    - Long runtime
    - Colab crashed from time to time
  - Unableness to work on the notebook file simultaneously
    - Different variable naming due to different author
  - Categorical Variable instead of Numerical

ROC Curve Comparing

<matplotlib.legend.Legend at 0x7febcaf12970>



Real Data Application

1. Creating a CSV file based on our team members' realistic situations.
2. Import the file in colab and put it through pipeline
3. Predict the outcome

```
print(raw_ex)
```

```
       ID    AGE GENDER      RACE DRIVING_EXPERIENCE   EDUCATION       INCOME  \
0  111111  16-25   male  minority               0-9y  university  upper class
1  222222  16-25   male  minority               0-9y  university  upper class
2  333333  16-25   male  minority               0-9y  university  upper class
3  444444  16-25   male  minority               0-9y  university  upper class

   CREDIT_SCORE  VEHICLE_OWNERSHIP VEHICLE_YEAR  MARRIED  CHILDREN  \
0           NaN                  1   after 2015        0         0
1           NaN                  1   after 2015        0         0
2           NaN                  0   after 2015        0         0
3           NaN                  1   after 2015        0         0

   POSTAL_CODE  ANNUAL_MILEAGE VEHICLE_TYPE  SPEEDING_VIOLATIONS  DUIS  \
0          NaN           10000   sports car                    1     0
1          NaN           20000        sedan                    0     0
2          NaN            5000        sedan                    0     1
3          NaN           25000        sedan                    1     0

   PAST_ACCIDENTS
0               0
1               1
2               1
3               1
```

```
GradientBoostingClassifier
GradientBoostingClassifier(learning_rate=1.0)
```

```
#Load ourseleves inthe dataset and fit on a svm to see our outcome result
raw_ex = pd.read_csv(data_folder + 'Car_Insurance_Claim_Example.csv')
df_X = df_X.append(raw_ex)
df_X = df_X.tail(4)
X_train1 = preprocessing.transform(df_X)

y_pred = model.predict(X_train1)
y_pred_prob = model.predict_proba(X_train1)[:, 1]
print(y_pred)
```

```
[0. 0. 0. 0.]
```

OUTCOME

Colab Link

https://colab.research.google.com/drive/1gbrkBBWQ6giTCuGd5TjCKrnKdJYe3FTc?usp=sharing

Reference

https://scikit-learn.org/stable/tutorial/statistical_inference/putting_together.html

https://www.statology.org/plot-multiple-roc-curves-python/

https://colab.research.google.com/drive/1sLPqMnYzr5blGzNAUSpkQ3PK8IJ4Y_Mc?usp=sharing#scrollTo=laKnvuqbfS7A

https://colab.research.google.com/drive/1Sk8UJK9R9vYiJR2vLrEe1niexNCYcOKh?usp=sharing

Q&A

Models Building and Tuning

1. Random Forest Classifier
2. Logistic Regression Classifier
3. K-nearest Neighbor Classifier
4. Support Vector Machines Classifier
5. XGBoost Classifier

In each model, we

1. Run the model with default parameters
2. Grid Search & Halving Search
3. Run the model using parameters with highest mean score from each search
4. Compare mean cv score, accuracy score, and ***balanced accuracy score***

```
y_test.value_counts()

0.0    1342
1.0     658
Name: OUTCOME, dtype: int64
```

Deploy into Business Strategy