BOOZ ALLEN HAMILTON INTERNAL

BAH Machine Learning Presentation
Team 3
Analytics Avengers
Members: Ahmed Sodeinde
Erick Orellana Morales
Rami Tello

April 2022

WEB AGE SOLUTIONS INC

TEAM 3:
Project Car Insurance Claim Prediction

Our Official Project for the BAH Machine Learning Program

WEB AGE SOLUTIONS INC

The Team

AHMED SODEINDE

Machine Learning Engineer

ERICK ORELLANA MORALES

Machine Learning Engineer

RAMI TELLO

Machine Learning Engineer

Booz | Allen | Hamilton

Project

Predicting if a car insurance claim will be filed within 6 months based on data related to the policy holder

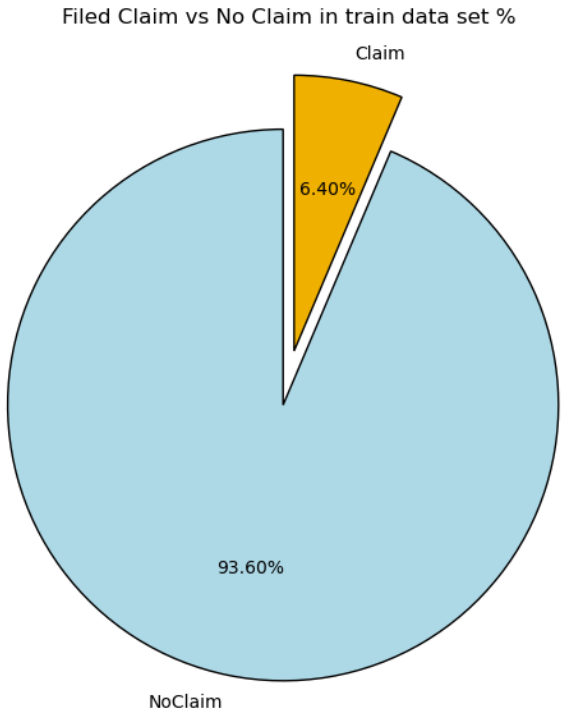
1. Introductions
2. Project Objectives
3. Project Overview
4. Architecture/Design diagram
5. Tools/Software Environment used
6. Challenges/Issues/Problems/Your stories
7. What have you learned/learned?
8. What else can we do on the project? - future plan- any design and implementation ideas?
9. Demo
10. Questions and answers

Project Goals & Business Objectives

- Project Summary:
  - Building a model that could be used to predict if a car insurance policy holder is likely to submit a claim over the next 6 months primarily based on information about the vehicle
- Business Use Case:
  - Having a prediction for the claims that may be filed can help make predictions about accepting new policy holders/renewing policies, pricing for policies, and making decisions about resources needed to handle the volume of claims
- Out of scope:
  - The type of claim (collision with another vehicle, collision with a stationary object etc)
  - Predicted cost of the claim
  - The type of driver
- Result:
  - It is difficult to predict whether a car insurance claim will be filed based on attributes in the available data. These attributes are largely about the car and does not have any information about the driver.

The Dataset

- CSV files
- Approximately 52,500 observations
- 41 attributes
- No duplicates or null values
- Imbalanced Dataset



Architecture and Design

Data Collection

S3 to store and read in new data

Pipeline to transform data

SageMaker Endpoint for inference

S3 to store predictions

Report outcomes for Decision Making

Technologies and Environment Used

- Python based
  - Libraries used:
    - pandas
    - numpy
    - sklearn
    - imblearn
    - seaborn
    - SageMaker
    - boto3
    - XGBoost
- AWS
  - IAM
  - S3
  - SageMaker
    - Notebooks
    - Studio
    - Registry
    - Endpoints

Amazon SageMaker

The Baseline Models (Dummy and Logistic)

- Baseline Model (Dummy Classifier)
  - Benchmark

```
Summary of performance:

The model's accuracy is:  0.9360322228290552
The model's F1 score is:  0.0
```

- Logistic Regression
  - Recursive Feature Elimination

	precision	recall	f1-score	support
0	0.50	1.00	0.66	11446
1	0.00	0.00	0.00	11586
accuracy			0.50	23032
macro avg	0.25	0.50	0.33	23032
weighted avg	0.25	0.50	0.33	23032

The Models (XGBoost)

- Pre-Hyperparameter Tuning

	precision	recall	f1-score	support
0	0.94	0.98	0.96	10963
1	0.16	0.04	0.07	756
accuracy			0.92	11719
macro avg	0.55	0.51	0.51	11719
weighted avg	0.89	0.92	0.90	11719

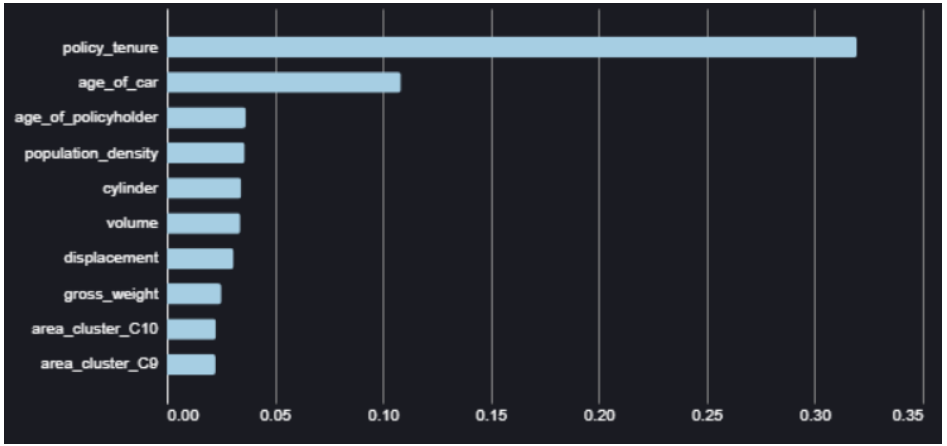
- With Hyperparameter Pruning

	precision	recall	f1-score	support
0	0.94	0.99	0.96	10963
1	0.16	0.04	0.06	756
accuracy			0.93	11719
macro avg	0.55	0.51	0.51	11719
weighted avg	0.89	0.93	0.90	11719

The Models (AutoML - Autopilot)

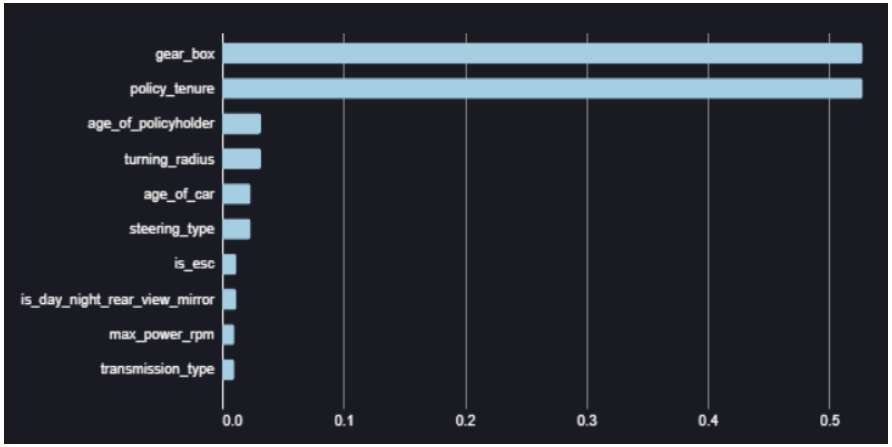
- Ensemble

F1 Objective	Accuracy	BalancedAccuracy	AUC	Precision	Recall	LogLoss	InferenceLatency	Algorithm
0.174	0.77	0.588	0.647	0.113	0.38	0.545	0.196	WeightedEnsemble



- Hyperparameter Optimization

F1 Objective	LogLoss	Recall	Precision	AUC	Accuracy	BalancedAccuracy	Algorithm
0.164	0.65	0.685	0.093	0.652	0.552	0.614	XGBoost



Demo

Experiment Conclusion

H_0: Car features cannot be used to predict claims being filed

H_a: Car features can be used to predict claims being filed

- Failed to reject the null Hypothesis
  - The data set did not have the signal to prove that filing claims can be predicted based on car features

Challenges and Learning

Challenges

Data Quality and Lineage

- The data used from Kaggle doesn't have a verifiable source to ensure the quality of the data
- ML Model performance is limited to the quality of the data
- Uncertainty regarding data completeness.

Lesson Learned

- Importance of balancing data
- Importance of data quality and lineage

Opportunity for Enhancement:

Gathering more data for model building and tuning:

- Increasing the number of observations of drivers that filed a claim in the dataset
- Increasing the number of features
- Further feature engineering

Q&A – Ask us anything!

BOOZ ALLEN HAMILTON INTERNAL

MARCH 2021

Appendix : Demo Backup

Send test data to best model built by AutoML AutoPilot

```
[146]: # xgb_predictor = tuner_linear.deploy(initial_instance_count=1, instance_type='ml.m4.xlarge')
xgb_tuned_predictor = sagemaker.predictor.Predictor(
    endpoint_name="SageMakerEndpoint-04132023",
    sagemaker_session=sagemaker.Session(),
    serializer=sagemaker.serializers.CSVSerializer()
)
```

```
[149]: s3_input_validate_df = pd.read_csv('s3://techexcellence.ml.project.team3/CarInsuranceClaim/data/test.csv')
s3_input_validate_df.head()
```

[149]:

	policy_id	policy_tenure	age_of_car	age_of_policyholder	area_cluster	population_density	make	segment	model	fuel_type	max_torque	max_power	engine_type	airbags	is_esc	is_adjustable_steering	is_tpms	is_parking_sensors	is_parking_camera	rear_bra
0	ID58593	0.341732	0.00	0.586538	C3	4076	1	A	M1	CNG	60Nm@3500rpm	40.36bhp@6000rpm	F8D Petrol Engine	2	No	No	No	Yes	No	
1	ID58594	0.307241	0.13	0.442308	C8	8794	1	B2	M6	Petrol	113Nm@4400rpm	88.50bhp@6000rpm	K Series Dual jet	2	No	Yes	No	Yes	No	
2	ID58595	0.327924	0.12	0.451923	C8	8794	2	A	M3	Petrol	91Nm@4250rpm	67.06bhp@5500rpm	1.0 SCe	2	No	No	No	No	Yes	
3	ID58596	0.782654	0.01	0.461538	C5	34738	1	A	M1	CNG	60Nm@3500rpm	40.36bhp@6000rpm	F8D Petrol Engine	2	No	No	No	Yes	No	
4	ID58597	1.233404	0.02	0.634615	C5	34738	1	A	M1	CNG	60Nm@3500rpm	40.36bhp@6000rpm	F8D Petrol Engine	2	No	No	No	Yes	No	

```
[147]: predictions_list = []

for i in chunker(s3_input_validate_df, 2500):
    test_data_array = i.values
    xgb_tuned_predictor.serializer = csv_serializer

    predictions = xgb_tuned_predictor.predict(test_data_array).decode('utf-8')
    predictions_array = np.fromstring(predictions[1:], sep='\n')
    predictions_list.append(predictions_array)
    print(predictions_array.shape)

y_hat_2 = np.concatenate(predictions_list, axis=None)
y_hat_2
```

```
...
```

```
[148]: output_df = pd.DataFrame(data=[s3_input_validate_df['policy_id'], y_hat_2]).T
output_df.columns = ['policy_id', 'is_claim_hat']
output_df.head(10)
```

[148]:

	policy_id	is_claim_hat
0	ID58593	0.0
1	ID58594	0.0
2	ID58595	0.0
3	ID58596	0.0
4	ID58597	0.0
5	ID58598	0.0
6	ID58599	1.0
7	ID58600	0.0
8	ID58601	0.0
9	ID58602	0.0