BAH Machine Learning Engineering Presentation
Team #2
Team Name: NMV
Members: Sebhat Gezehey, Abel Kebekabe, Alexander Bradley

April 2023

TEAM 2 (NMV)
Project: Home Loan Approval

BAH TechX Machine Learning Engineering Program

INTRODUCTIONS

SEBHAT GEZEHEY
Associate
Lead Engineer

ABEL KEBEKABE
Senior Consultant
ML Engineer

ALEXANDER BRADLEY
Senior Consultant
ML Engineer

Booz | Allen | Hamilton

Project

Home Loan Approval

1. Introductions
2. Project Overview
3. Project Process and Schedule
4. Design/Architecture Diagram
5. Tools/Environments Used
6. EDA
7. Models and Results
8. Challenges/Issues/Lessons
9. Future Plans
10. Demo
11. Questions and Answers

Project Overview

Booz | Allen | Hamilton

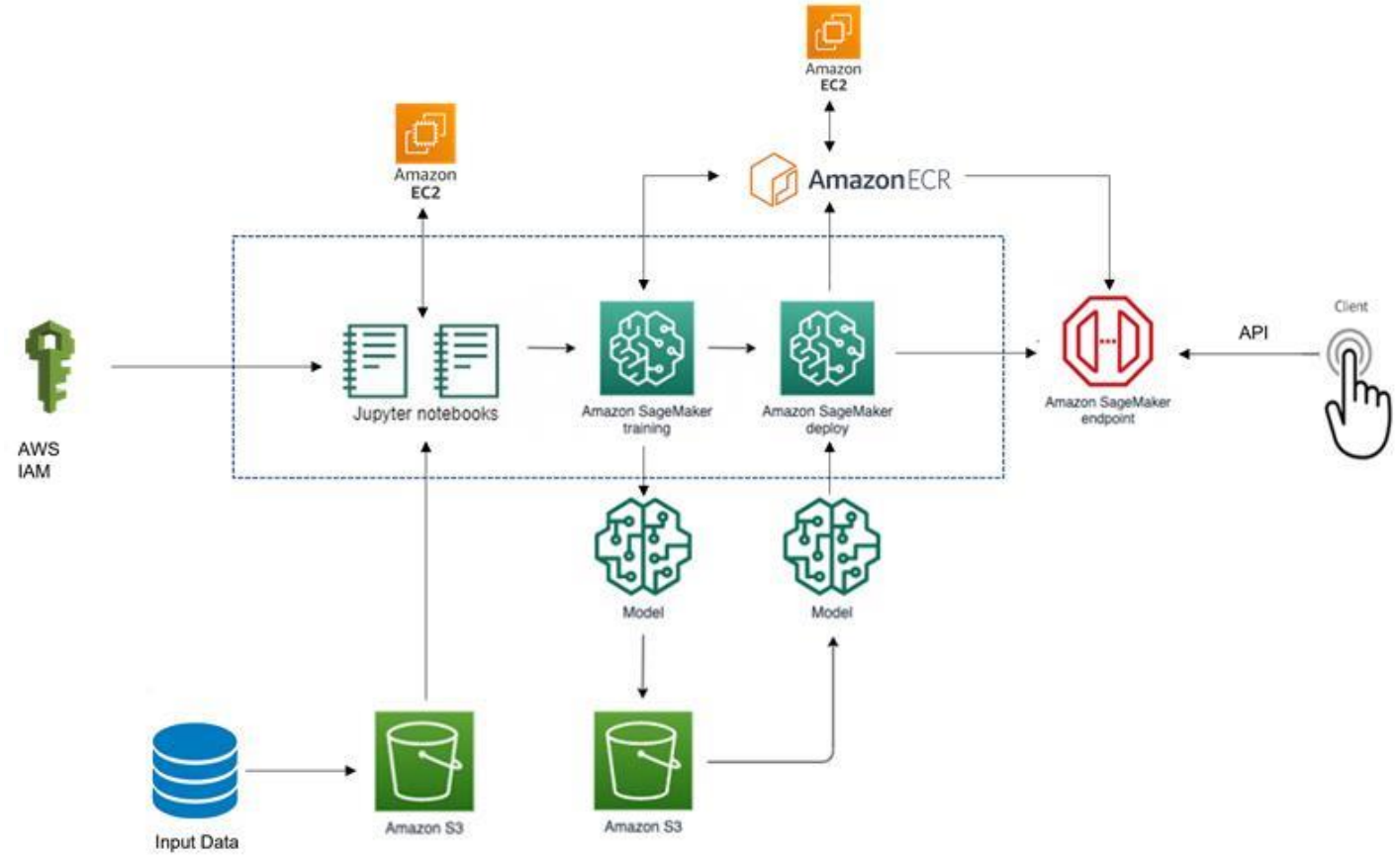
- **Summary:** A bank has a list of past applicants who previously applied for home loans. Data is provided for each applicant and whether the loan was approved or denied. A model to predict whether a loan is approved based off applicant information requested.
- **Business Case**:
  - Reduce loan process time
  - Increase consistency among all home loan approvals across the company
- **Problem Statement:** Identify the best way to approve home loans based off available applicant data
- **Goal(s):** Deploy an ML model that determines approval of home loans consistent with past approvals
- **Scope:**
  - Report any significant findings in the data
  - Create and deploy a predictive model that determines home loan approval
  - Out-of-scope: Determining reason for loan approval
- **Current State Metrics:**
  - F-Score, Accuracy, Precision, Recall, AUC

Project Process and Schedule

- Day 1-2
  - Perform EDA to understand dataset
  - Discuss metrics for scoring models to fit business case
  - Establish Project Overview
- Day 3
  - Create multiple models and compare results
- Day 4
  - Continue model evaluation
  - Choose a final model to deploy based off metrics and ease of deployment
- Day 5
  - Finalize documentation and presentation

System Design and Architecture

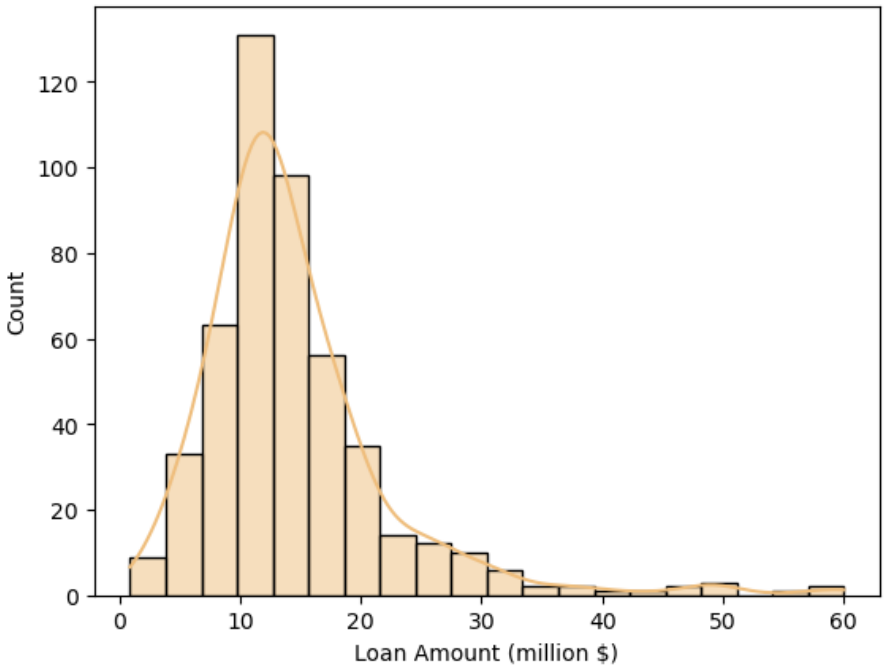
Booz | Allen | Hamilton



Technologies and Environment used

- **AWS SageMaker**
- **AWS EC2**
- **AWS S3**
- **AWS ECR**
- **GitHub**

Exploratory Data Results



Loan Amounts of $0 value removed for graph and statistics

Minimum Loan: $900k

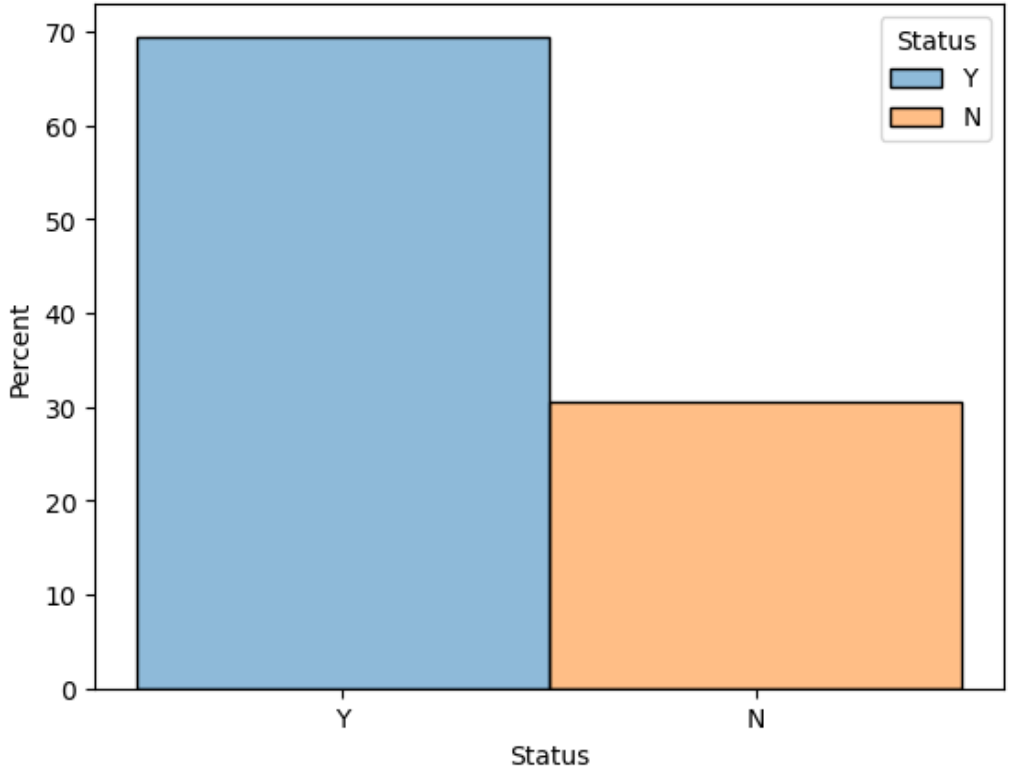
Maximum Loan: $60M

Median Loan: 12.8M

Exploratory Data Results

	468	500
Gender	Female	Female
Married	Yes	No
Dependents	2	0
Education	Not Graduate	Graduate
Self_Employed	NaN	No
Applicant_Income	21000	64500
Coapplicant_Income	291700.0	368300.0
Loan_Amount	9800000	11300000
Term	360.0	480.0
Credit_History	1.0	1.0
Area	Semiurban	Rural
Status	Y	Y

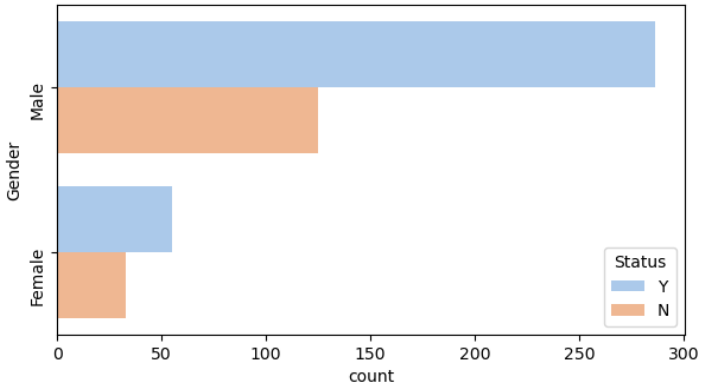
Exploratory Data Results – Continued…

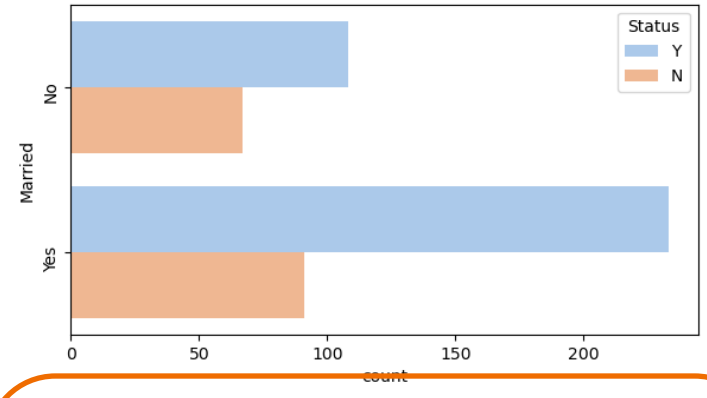


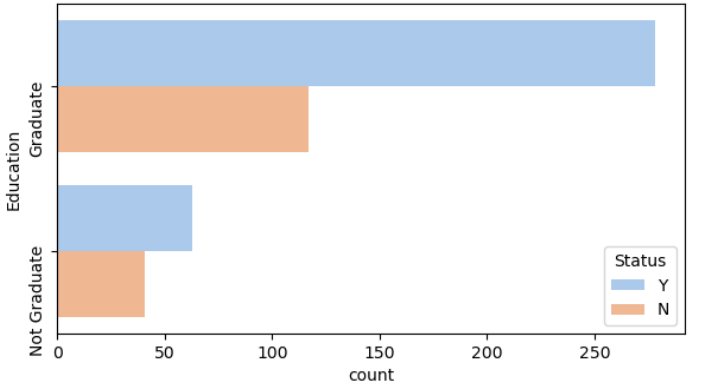
Observation:
Target Variable is Imbalanced

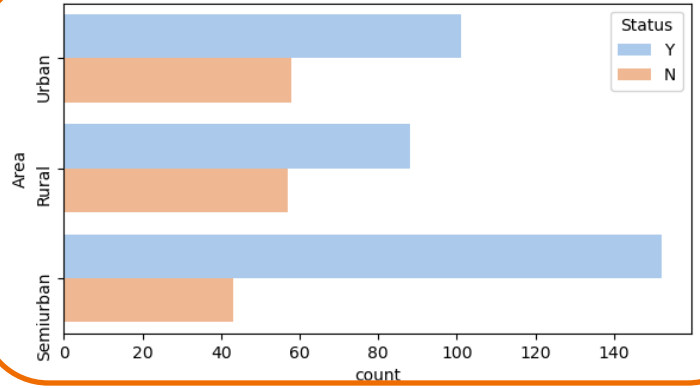
Exploratory Data Results – Continued…

Booz | Allen | Hamilton



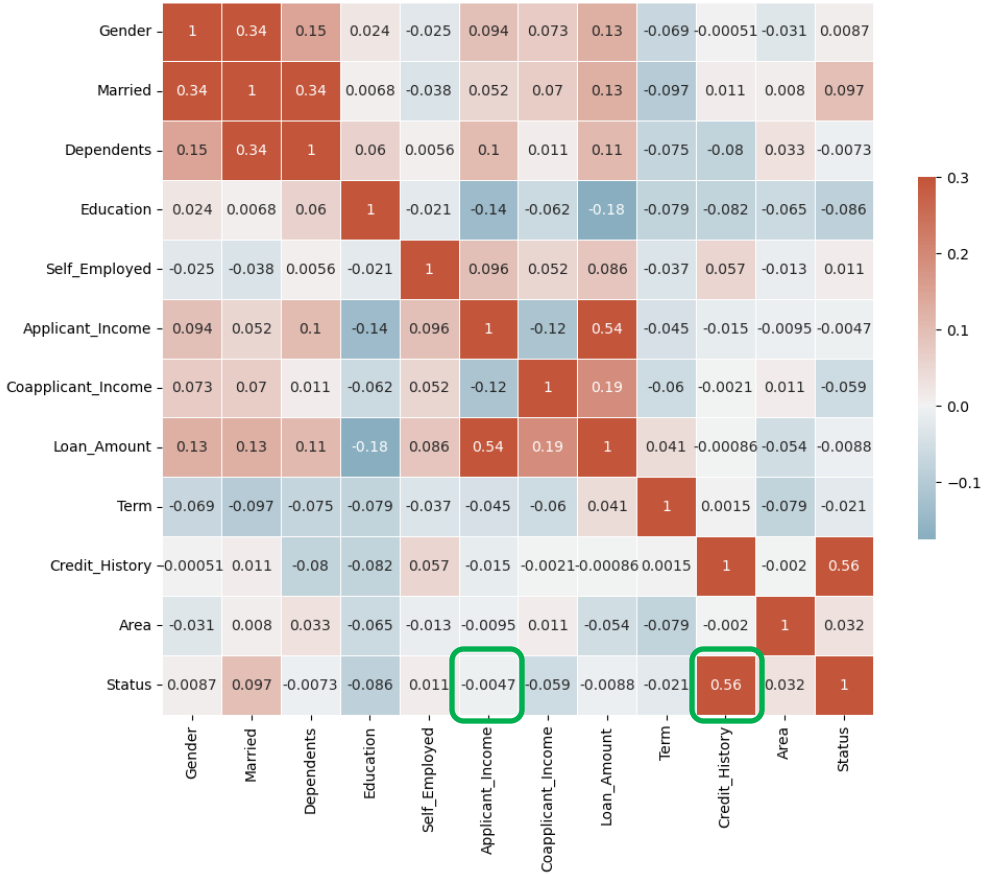






Exploratory Data Results – Continued…

Booz | Allen | Hamilton



Exploratory Data Results – ...Continued

Other Observations

6 of the 12 of the columns contained missing values.
The table to the right shows the 6 columns with the highest percentage of missing values.

	% of Missing Values
Credit_History	8.14%
Self_Employed	5.21%
Dependents	2.44%
Term	2.28%
Gender	2.12%
Married	0.49%

Variable "Loan_Amount" contained loans of $0 value

Dataset Properties

Rows	Columns	Duplicate rows	Target column	Missing target values	Invalid target values	Detected problem type
614	12	0.00%	Status	1.88%	1.88%	BinaryClassification

Detected Column Types

	Numeric	Categorical	Text	Datetime	Sequence
Column Count	4	7	0	0	0
Percentage	36.36%	63.64%	0.00%	0.00%	0.00%

Model Versions

Version	Model	Description
Baseline	 AutoML	AutoML was used to produce a quick analysis of multiple models. WeightedEnsemble was the best model.
Version 1.0	Random Forest	Model was chosen for being a viable solution to a binary classification problem.
Version 1.1	XGBoost	Chosen as it was one of the better performing algorithms from the AutoML.
Version 1.2	XGBoost	Additional tuning of XGBoost was done.

Model Versions

Version	Model	Features
Baseline	Amazon SageMaker AutoML	Raw Data
Version 1.0	Random Forest	1) Removed Missing Data 2) Label Encoded Categorical Variables
Version 1.1	XGBoost	1) Removed Missing Data 2) Label Encoded Categorical Variables
Version 1.2	XGBoost	1) Removed Missing Data 2) Label Encoded Categorical Variables 3) Removed $0 Loan Amount Value

Model Results

Version	Model	F1	Accuracy	Precision	Recall	AUC
Baseline	Amazon SageMaker AutoML	0.677	0.839	0.913	0.538	0.247
Version 1.0	Random Forest	0.85	0.77	0.77	0.88	0.67
Version 1.1	XGBoost	0.96	0.96	0.96 (1.0 on Approval)	0.96	0.942
Production Version 1.2	XGBoost	0.82	0.84	0.85	0.84	0.748

Challenges, Issues, Lessons

- More data desired for more testing
- Learned it is harder to deploy a Bring Your Own Model on AWS SageMaker than a native SageMaker algorithm
- On testing it was found that a large income, but no credit history resulted in a high likelihood of the loan being denied

Future Plans/Recommendations

- Additional common-sense checks of model inputs like the issue mentioned above where no credit history resulted in a high likelihood loan being denied
- Testing of additional models
- Hyperparameter tuning
- Create AWS MLOps Pipeline to make model iterations faster

Demo


```
predictions_array
```

```
array([0.76343787, 0.46597332, 0.70553535, 0.83843571, 0.82133377,
       0.90497601, 0.72584224, 0.23788796, 0.78886515, 0.65653336,
       0.72160769, 0.76274806, 0.79909384, 0.74878019, 0.69548339,
       0.87321091, 0.87321091, 0.19923036, 0.78186738, 0.83322304,
       0.17902289, 0.19555779, 0.81155813, 0.23835607, 0.88869292,
       0.85579586, 0.87321091, 0.14648503, 0.69306409, 0.19666469,
       0.87843382, 0.81664765, 0.82935792, 0.71457011, 0.55616462,
       0.8456676 , 0.817568  , 0.80307311, 0.92997241, 0.71048701,
       0.58344918, 0.79251015, 0.82907891, 0.64178544, 0.90586245,
       0.13079849, 0.20029396, 0.66306961, 0.82758921])
```

```
from sklearn.metrics import classification_report
print(classification_report(test_data['Status'], np.round(predictions_array), target_names=['approve', 'deny']))
```

```
              precision    recall  f1-score   support

     approve       0.90      0.56      0.69        16
        deny       0.82      0.97      0.89        33

    accuracy                           0.84        49
   macro avg       0.86      0.77      0.79        49
weighted avg       0.85      0.84      0.82        49
```

Demo

Booz | Allen | Hamilton

Newly Generated Applicant Data

```
sample_record_csv = '''Gender,Married,Dependents,Education,Self_Employed,Applicant_Income,Coapplicant_Income,Loan_Amount,Term,Credit_History,Area
1,0,0,1,0,584900,0,15000000,360,1,2
'''
```

```
import pandas as pd
from io import StringIO
```

```
sample_record_io = StringIO(sample_record_csv)
sample_df = pd.read_csv(sample_record_io)
```

```
sample_record_io = StringIO(sample_record_csv)
sample_df = pd.read_csv(sample_record_io)
```

```
sample_record_io = StringIO(sample_record_csv)
sample_df = pd.read_csv(sample_record_io)

#if contains_target_column:
    # Drop the target column from the sample csv
    #print(f"Target column value of sample record: {sample_df.iloc[0][target_column_name]}")
    #sample_df = sample_df.drop(columns=[target_column_name])

sample_record_payload = sample_df.to_csv(header=False, index=False)
print(f"Sample record to predict: {sample_record_payload}")
```

```
Sample record to predict: 1,0,0,1,0,584900,0,15000000,360,1,2
```

```
prediction = predictor.predict(sample_record_payload, initial_args={"ContentType": "text/csv"})

print(f"The predicted target is: {prediction}")
if prediction > b'0.5':
  print("Loan has been approved")
elif prediction < b'0.5':
  print("Loan has been denied")
```

Result

```
The predicted target is: b'0.8253770470619202'
Loan has been approved
```

Q & A

April 2023