# Coupon usage prediction on In-Vehicle Recommendation systems

Anushka Hegde. Yanming Liu. Krishna Hemant

# Problem Setting

- Coupons are seen almost everywhere these days, from online clothing websites to travel agencies, food delivery apps and third party websites or applications hosting them.
- Hosting coupons on another service can bring a lot of additional costs.
- It is imperative to accurately recommend the most likely to be accepted coupons for the target group.

## Research Objective:

Build a robust machine learning model to classify if the user will accept the given coupon or not.

# Blueprint at a Glimpse

- **Coupon Recommendation dataset** — Exploratory data analysis
- **Feature Engineering** — Feature selection and transformation
- **Model Exploration and Implementation** — Logistic Regression, Neural Networks, SVM(Hard and soft), Gaussian Naïve Bayes
- **Model Selection and Evaluation** — Choosing the best model according to the performance metrics

# Data Source & Explanation

- https://archive.ics.uci.edu/ml/datasets/in-vehicle+coupon+recommendation
- 12684 records, 26 attributes

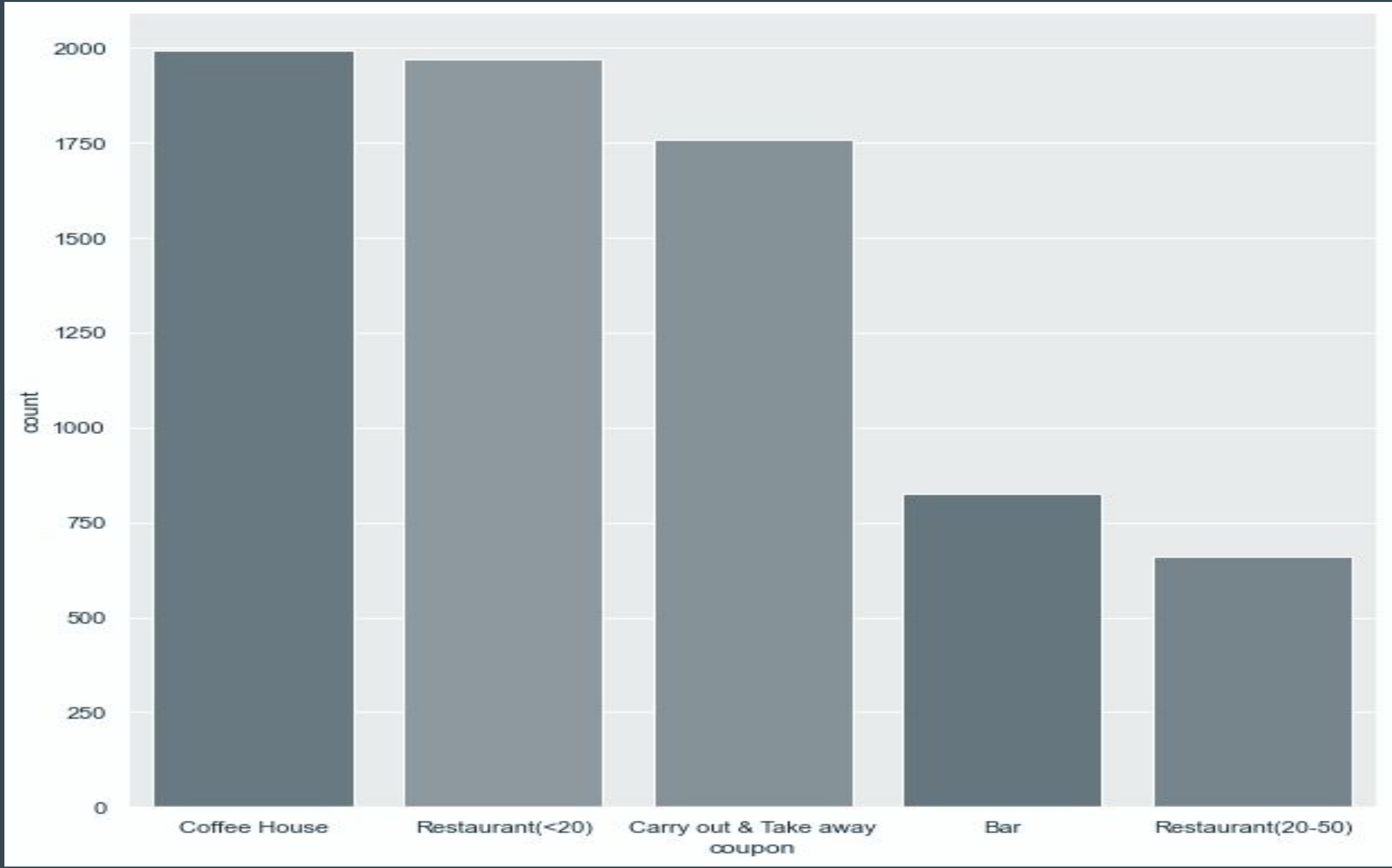
| | destination | passanger | weather | temperature | time | coupon | expiration | gender | age | maritalStatus | ... | CoffeeHouse | Ca |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | No Urgent Place | Alone | Sunny | 55 | 2PM | Restaurant(<20) | 1d | Female | 21 | Unmarried partner | ... | never | |
| 1 | No Urgent Place | Friend(s) | Sunny | 80 | 10AM | Coffee House | 2h | Female | 21 | Unmarried partner | ... | never | |
| 2 | No Urgent Place | Friend(s) | Sunny | 80 | 10AM | Carry out & Take away | 2h | Female | 21 | Unmarried partner | ... | never | |
| 3 | No Urgent Place | Friend(s) | Sunny | 80 | 2PM | Coffee House | 2h | Female | 21 | Unmarried partner | ... | never | |
| 4 | No Urgent Place | Friend(s) | Sunny | 80 | 2PM | Coffee House | 1d | Female | 21 | Unmarried partner | ... | never | |

5 rows × 26 columns

# EXPLORATORY DATA ANALYSIS

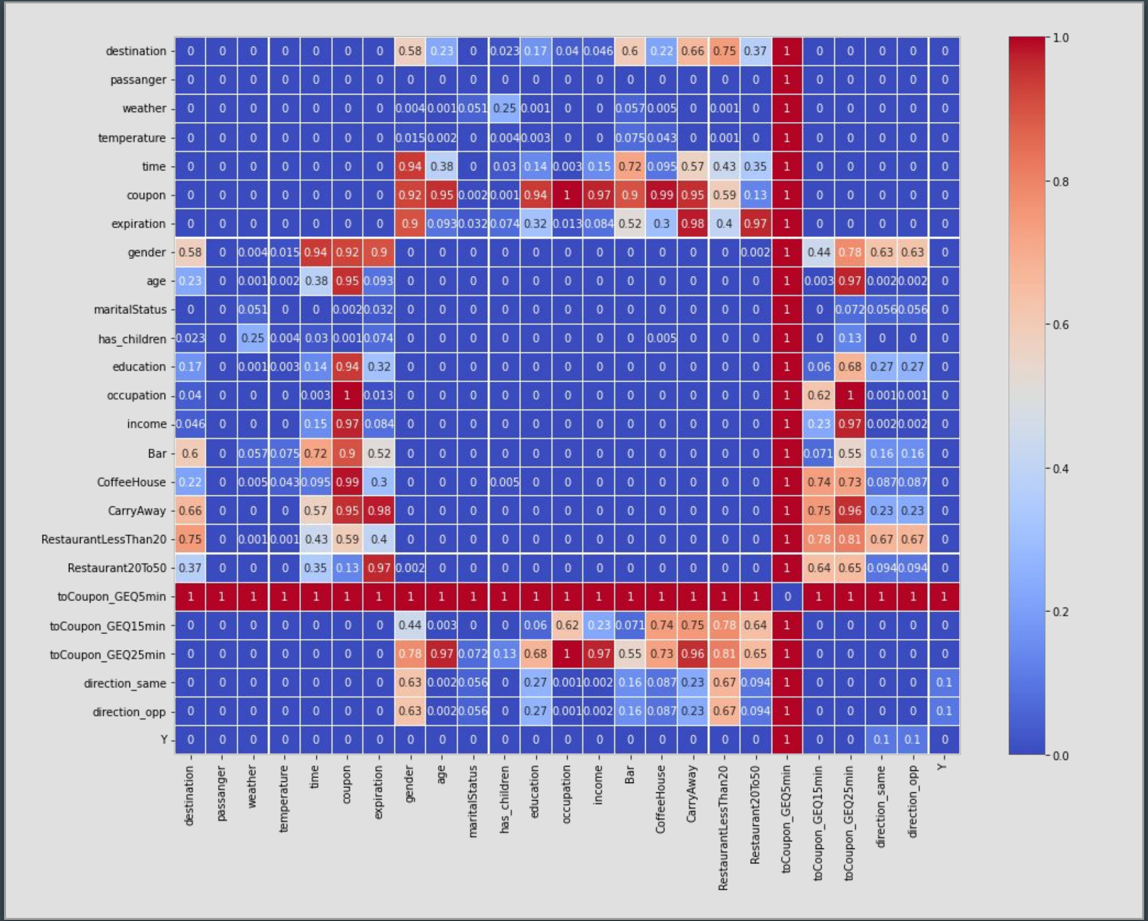
- Data Cleaning
  - Drop car column with 99% null values
  - Impute other nan value in categorical feature with the most frequent category occurrence
- Data Transformation
  - Set the data type for all features as category as expected
- Data Analysis
  - Coupons acceptance vs. rejection
  - Counts for coupons accept or not among various categories of features
  - Counts for different coupon types
  - Independence between features and coupon acceptance

# Coupons

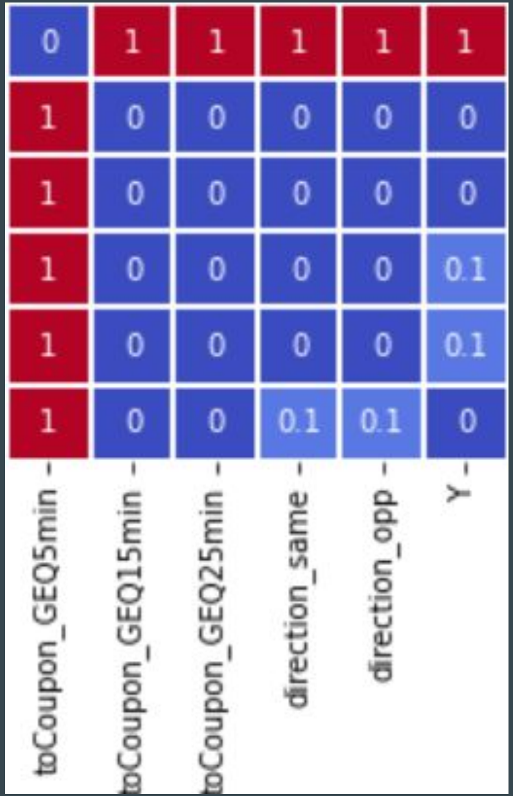


count
2000
1750
1500
1250
1000
750
500
250
0
Coffee House
Restaurant(<20)
Carry out & Take away
Bar
Restaurant(20-50)
coupon

# Correlation Analysis - Chi-square




Y



# Feature Engineering

- Feature Selection
  - Drop,

    toCoupon_GEQ5min, direction_same, direction_opp
- Feature Encoding
  - Five ways are conducted, and

    mixture 2 style is picked for a

    consideration of higher accuracy and robustness

    for neural networks.

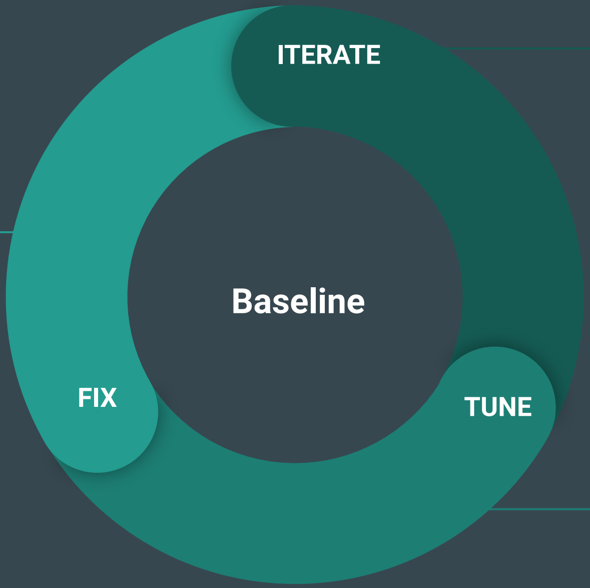
| | Encodings | Test Accuracy |
|---|---|---|
| 0 | OneHot | 0.672842 |
| 1 | Ordinal | 0.595980 |
| 2 | Target | 0.689791 |
| 3 | Mixture1 | 0.679543 |
| 4 | Mixture2 | 0.682696 |

# MACHINE LEARNING MODELS

- Logistic Regression
- Neural Networks
- Support Vector Machines
- Gaussian Naive Bayes

# HYPER-PARAMETER TUNING WORKFLOW

Fix all
but one parameter




Use the optimal value obtained from accuracy maximization for subsequent iterations

Select a new parameter to tune

Run the model over a given set of values of the flexible parameter

# Logistic Regression

- It is an extension of Linear Regression where the probability of class membership is analysed based on the sigmoid output.
- Logistic Regression was implemented with Stochastic Gradient Descent to accelerate convergence.
- Best parameters:
  - Learning rate = 0.001
  - Tolerance = 0.001
  - Maximum Iterations = 1000
  - Batch size = 128

```
learning_rate = [0.001, 0.005, 0.01, 0.1, 0.5]
Tolerance = [0.001, 0.005, 0.01, 0.1, 0.5]
max_Iterations = [100, 200, 500, 1000, 5000]
batch_sizes = [32, 64, 128, 512, 1028]
```

# Neural Network

- Neural Networks is implemented by Keras using optimizers to decrease the categorical cross entropy, and the model performance is obtained by test accuracy.
- Baseline test accuracy with no hidden layer, no dropout value, no regularization , SGD optimizer is obtained for comparison with the one after parameter tuning.
- Effect of hidden layer, and other parameters on validation accuracy is tested, and adopting the turning strategy as previous slide describe, we for loop the values on a model with 3 layers, in the graph and get the best parameters as below,
- dropout = 0.3, optimizer = ‘Adam’, epoch = 50,
  Batch size = 32, learning rate = 0.001,
  number of hidden neurons = 2048,

```
dropout = [0.1, 0.2, 0.3, 0.4, 0.5]
optimizers = ['RMSProp', 'Adam']
num_epochs = [25, 50, 100, 200, 300]
batch_size = [32, 64, 128, 256, 512]
learning_rate = [0.001, 0.005, 0.01, 0.1, 0.5]
n_hiddens = [128, 256, 512, 1024, 2048]
regularization_type = ['l1', 'l2']
```

# SVM (Hard Margin)

- It maximises the minimum distance from the decision boundary to the training points with no leeway for misclassification.
- Hard Margin SVM is implemented with l2 regularization.
- Best parameters for Hard Margin SVM:
  - Learning rate = 0.00001
  - Lambda  = -0.001
  - Maximum Iterations = 1000

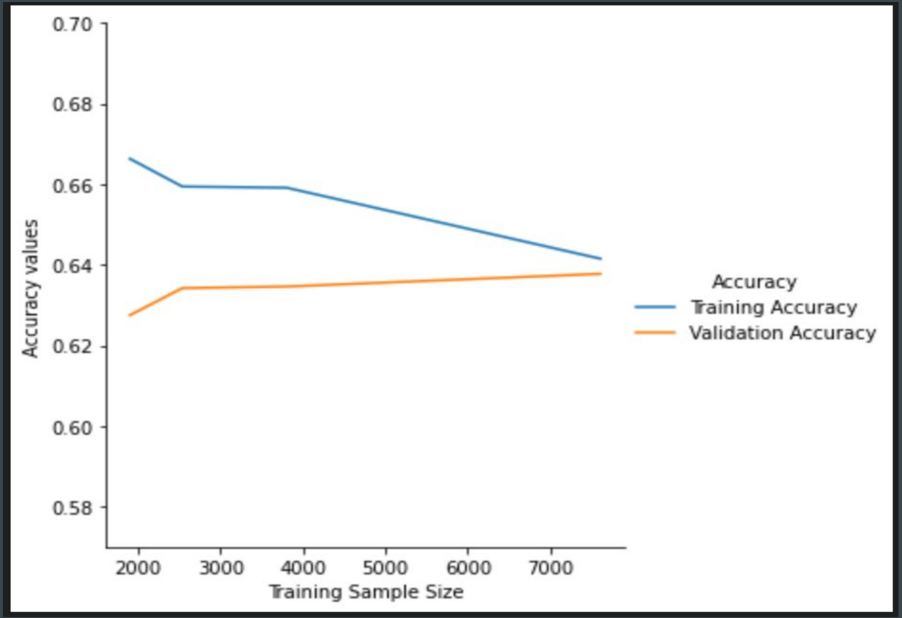
```
epochs = [25, 50, 100, 200, 300, 1000]
learning_rate = [0.00001, 0.0001, 0.001, 0.005, 0.01]
lamda = [0.0001, 0.001, 0.01, 0.1, 1]
```

# Gaussian Naive Bayes

- Gaussian NB supports continuous values or mixture encoded data and assumes an underlying Gaussian (normal) distribution with independent features.
- Naive Bayes was also attempted on one-hot encoded data with laplace smoothing
  - However, no significant improved over the baseline was observed
- Accuracy metrics on train, validation and test set:
  - Train = 0.6415
  - Validation = 0.6377
  - Test = 0.6357

# BIAS VARIANCE TRADEOFF (LEARNING CURVES)

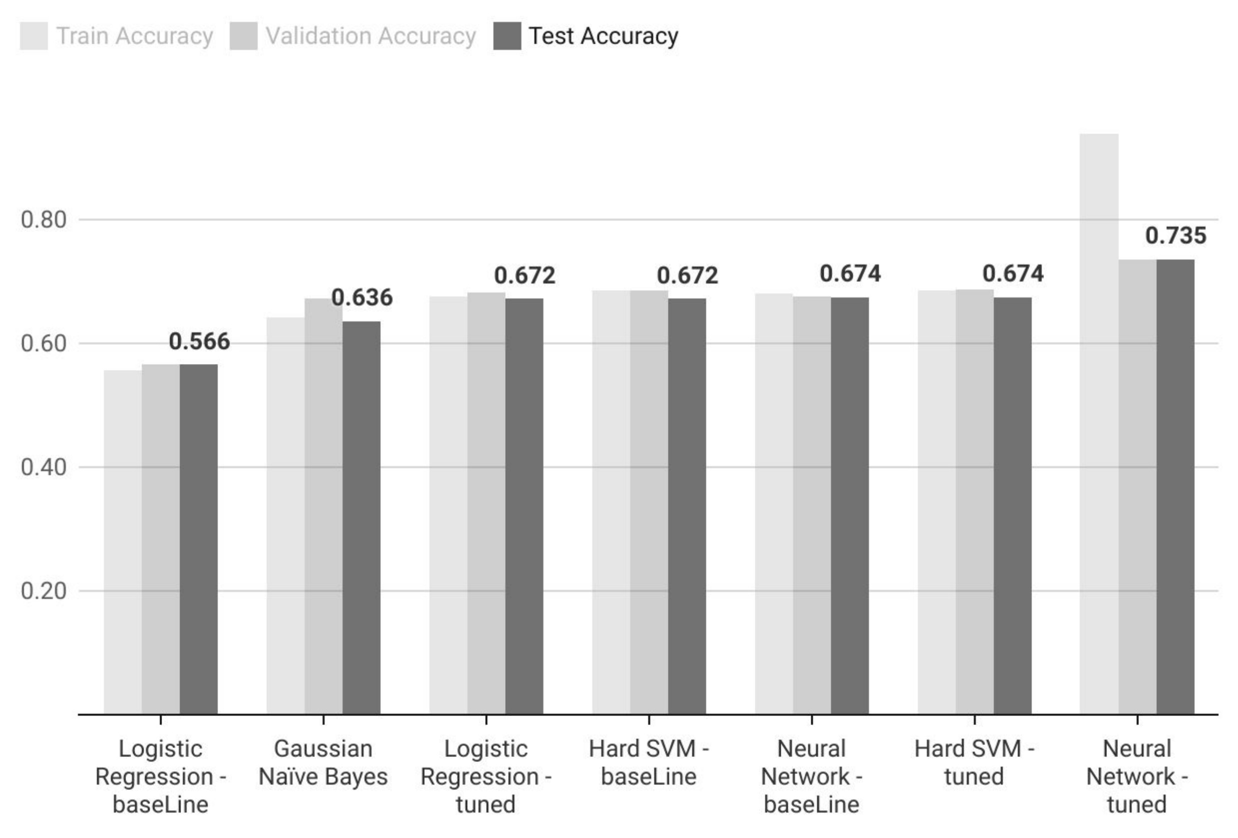
- A learning curve shows the validation and training accuracy of a model for varying numbers of training sample sizes.
- In a high bias learning curve:
  - Low overall performance
  - Difference in convergence of training and test accuracies is small
- Resolve: Increase the number of parameters or create new features.



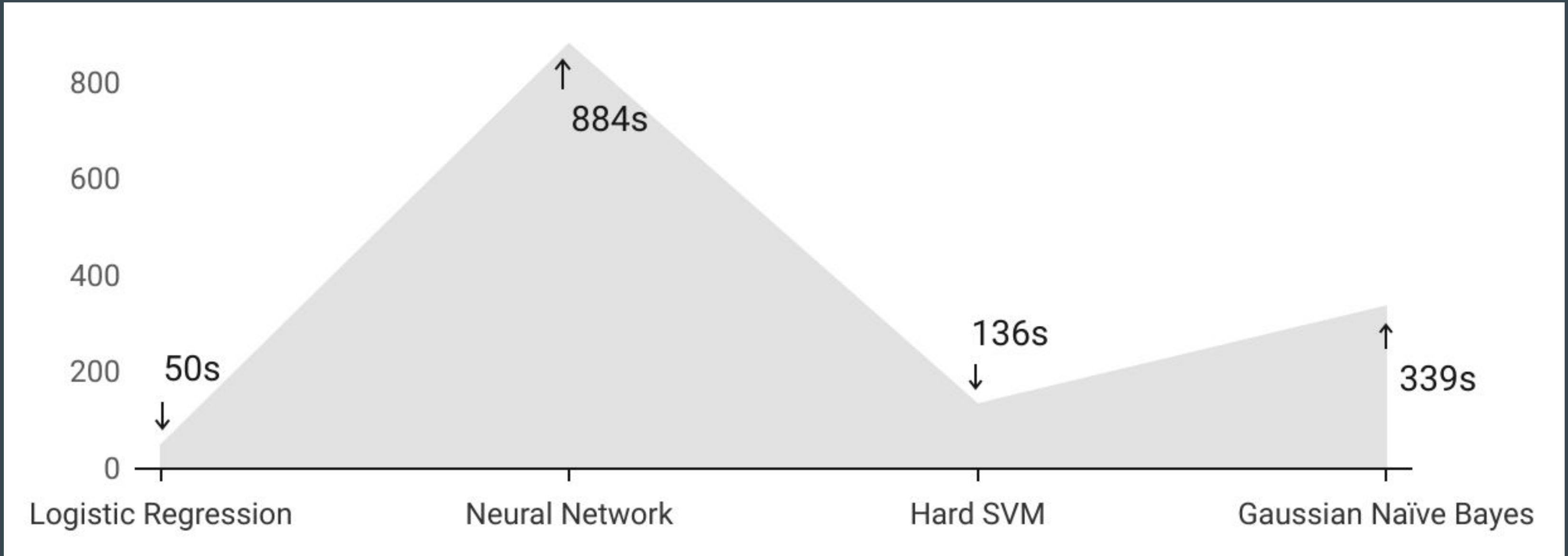
# Performance Evaluation

- Accuracy
- Time consumption
- Space Usage

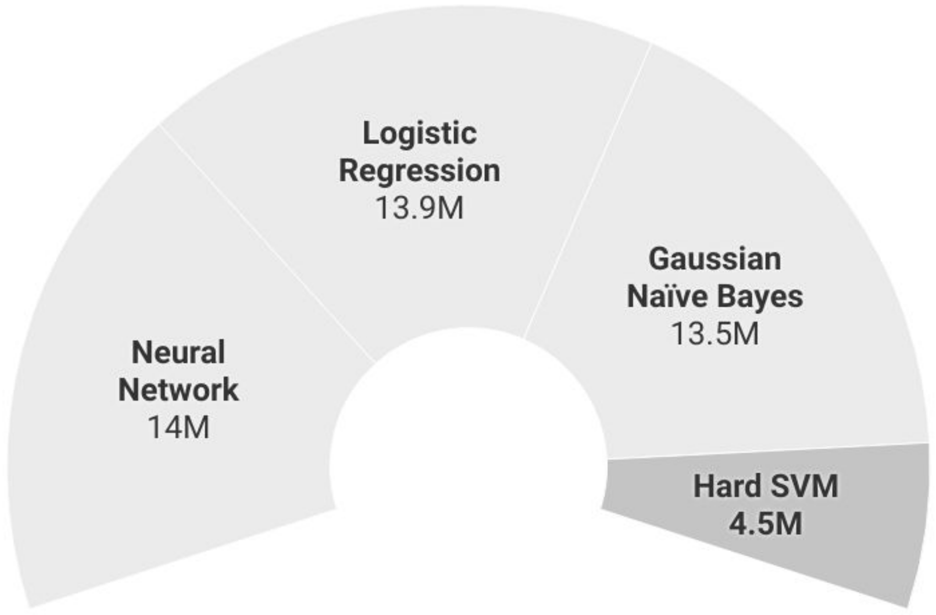
# Accuracy for Models



# Processing Time



# Peak Memory Usage (B)



# Summary

Based on the model selection criteria, we can choose the optimal models according to our preference for accuracy or efficiency. Since the overall performance of the tuned Neural Networks model is approximately 6% higher than the next best, when prioritizing high accuracy, this model will work well. When computational efficiency may be an issue, particularly in the case of large datasets, it may be wiser to go with the Hard Margin SVM classifier.

# Thank you!

- https://github.com/AnushkaHegde/In-Vehicle-Coupon-Recommendation-ML-Project