Project: In-Vehicle Coupon Recommendation

"We have some catching up to do in the area of machine learning and artificial intelligence." ~Klaus Froehlich

Introduction

To successfully grows a business, understanding customer behavior is essential [1]. In recent years machine learning researchers have worked on customer behavior analysis [1]. Many big tycoon companies such as Amazon, Google, and Facebook invest time, effort, and money in machine learning only to understand the behavior of users so they can make business decisions[2].

In this project we have worked on "In-Vehicle Coupon Recommendation", training a machine learning model that predicts customer behavior that it will accept a coupon of a nearby location(restaurant or coffee house, or bar) based on user, contextual and coupon information.

There is a research paper that uses the following technique.

Table 1. Building 4 sub-datasets.

Sub-datasets	Mode imputing	Random forest imputing
No scale	d1	d3
Scale	d2	d4

Table 2. Performance results.

		Dataset			
Model	**Measure**	**d1**	**d2**	**d3**	**d4**
Decision	acc	0.69	0.69	0.69	0.69

tree	f1	0.73	0.73	0.73	0.73
	auc	0.68	0.68	0.68	0.68
Random forest	acc	**0.76**	**0.76**	0.75	0.75
	f1	0.79	0.79	0.79	0.79
	auc	**0.83**	**0.83**	**0.83**	**0.83**
Logistic regression	acc	0.69	0.69	0.69	0.68
	f1	0.74	0.73	0.74	0.73
	auc	0.74	0.74	0.74	0.74
SVC	acc	0.69	0.69	0.69	0.69
	f1	0.74	0.74	0.74	0.74
	auc	0.74	0.74	0.74	0.74
MLP	acc	0.74	0.74	0.74	0.75
	f1	0.78	0.78	0.78	0.78
	auc	0.81	0.81	0.81	0.81
Bagging	acc	**0.76**	**0.76**	**0.76**	**0.76**
	f1	**0.80**	**0.80**	**0.80**	**0.80**
	auc	**0.83**	**0.83**	**0.83**	**0.83**
Adaboost	acc	0.68	0.68	0.68	0.68
	f1	0.73	0.73	0.73	0.73
	auc	0.74	0.74	0.74	0.74
XGBoost	acc	0.72	0.72	0.72	0.72
	f1	0.77	0.77	0.77	0.77
	auc	0.79	0.79	0.79	0.79

In this we applied the following techniques:

Model Name	Accuracy	AUC	F1
KNN	0.70	0.74	0.70
Naive Bayes	0.68	073	0.68

Logistic Regression	0.71	0.76	0.71
SVM	0.73	0.778	0.73
Random Forest	0.68	0.73	0.68
CatBoost	0.72	0.777	0.72
ANN	0.71	0.76	0.71

To keep in view above we apply hyperparameter tuning on SVM model and deploy it on a Flask.

Problem statement

Understanding customer behavior is essential for business decision-making. Business decision-making is a very critical task because it involves the money and effort of companies, and any misinformation can lead to a big problem.

The dataset that we use in this project has problems such as duplicates, missing values, and many unnecessary data(noise). There is a need for pre-processing before moving toward modeling.

Proposed Methodology

- Explored the dataset and found problems in it.
  - Features and shape:

```
destination  :  ['No Urgent Place' 'Home' 'Work']
passanger  :  ['Alone' 'Friend(s)' 'Partner' 'Kid(s)']
weather  :  ['Sunny' 'Snowy' 'Rainy']
temperature  :  [55 80 30]
time  :  ['6PM' '7AM' '10PM' '2PM' '10AM']
coupon  :  ['Coffee House' 'Restaurant(<20)' 'Carry out & Take away' 'Bar'
 'Restaurant(20-50)']
expiration  :  ['2h' '1d']
gender  :  ['Female' 'Male']
age  :  ['26' '21' '41' '50plus' '46' '36' '31' 'below21']
maritalStatus  :  ['Married partner' 'Unmarried partner' 'Single' 'Widowed' 'Divorced']
has_children  :  [0 1]
education  :  ['Associates degree' 'Some college - no degree'
 'Graduate degree (Masters or Doctorate)' 'Bachelors degree'
 'High School Graduate' 'Some High School']
occupation  :  ['Unemployed' 'Education&Training&Library' 'Business & Financial'
 'Student' 'Arts Design Entertainment Sports & Media' 'Sales & Related'
 'Personal Care & Service' 'Office & Administrative Support' 'Legal'
 'Management' 'Farming Fishing & Forestry' 'Architecture & Engineering'
 'Computer & Mathematical' 'Retired' 'Food Preparation & Serving Related'
 'Healthcare Support' 'Production Occupations' 'Protective Service'
 'Life Physical Social Science' 'Community & Social Services'
 'Transportation & Material Moving' 'Healthcare Practitioners & Technical'
 'Installation Maintenance & Repair'
 'Building & Grounds Cleaning & Maintenance' 'Construction & Extraction']
income  :  ['$37500 - $49999' '$25000 - $37499' '$87500 - $99999' '$100000 or More'
 '$12500 - $24999' '$62500 - $74999' 'Less than $12500' '$50000 - $62499'
 '$75000 - $87499']
Bar  :  ['less1' '4~8' 'never' '1~3' 'gt8']
CoffeeHouse  :  ['1~3' 'less1' '4~8' 'never' 'gt8']
CarryAway  :  ['1~3' '4~8' 'less1' 'gt8' 'never']
RestaurantLessThan20  :  ['1~3' 'less1' '4~8' 'gt8' 'never']
Restaurant20To50  :  ['1~3' 'never' 'less1' '4~8' 'gt8']
toCoupon_GEQ5min  :  [1]
toCoupon_GEQ15min  :  [0 1]
toCoupon_GEQ25min  :  [0 1]
direction_same  :  [0 1]
Y  :  [1 0]
```

```
df.shape
```

```
(12684, 26)
```

- There is a feature named toCoupon_GEQ5min that has a constant value.

```
toCoupon_GEQ5min   :   [1]
```

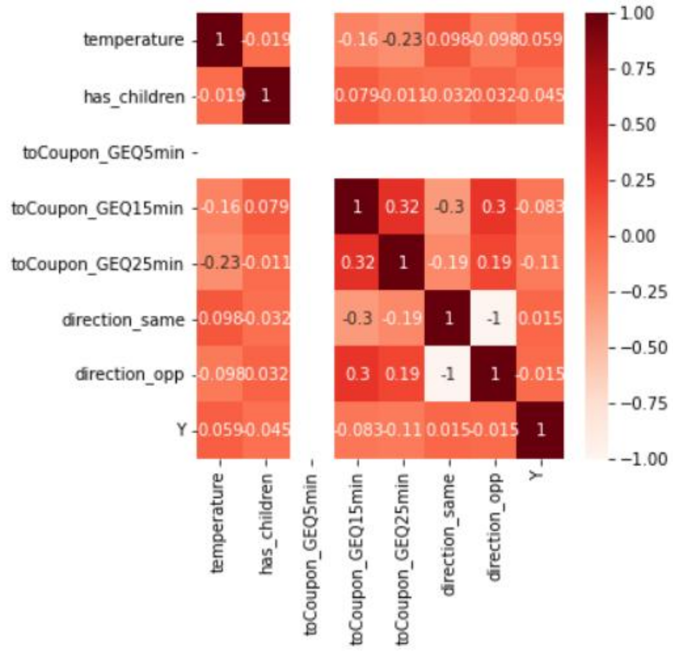
- Missing values:

```
car                     12576
Bar                       107
CoffeeHouse               217
CarryAway                 151
RestaurantLessThan20      130
Restaurant20To50          189
dtype: int64
```

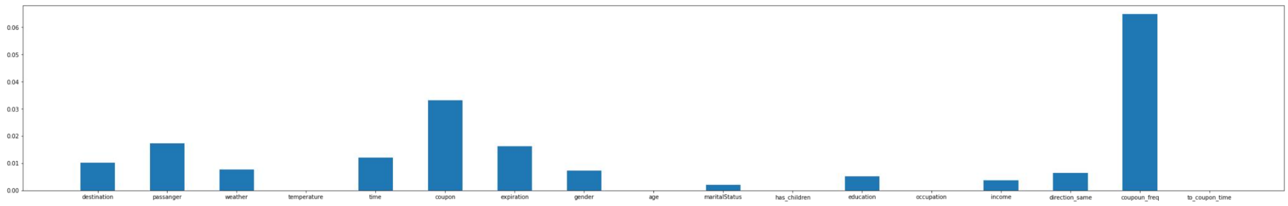
- Duplicate data:

Duplicate rows	74

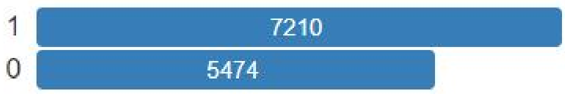
- Correlation between features:



- Most of the columns in the dataset are categorical and some of them are unnecessary that contribute nothing to prediction:



- Class Imbalance:

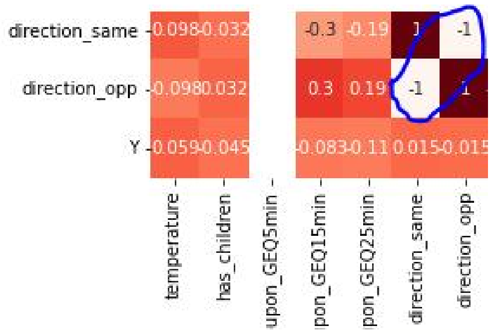


- Applied pre-processing to solve the above-mentioned problems.
  - Drop duplicate data
  - Drop feature named car, because it is 99% null

Missing	12576
Missing (%)	99.1%

  - Fill mode of features for filling missing values, because all the features we have are categorical.

- Drop the feature named direction_opp, because it is highly correlated with the feature named direction_same



- Extract a new feature named coupon_freq from Bar, CoffeeHouse, CarryAway, RestaurantLessThan20, and Restaurant20To50 because we want only the feature that name is provided in the coupon column.

```
couponDictionary = {
    "Coffee House":"CoffeeHouse",
    "Restaurant(<20)":"RestaurantLessThan20",
    "Carry out & Take away":"CarryAway",
    "Bar":"Bar",
    "Restaurant(20-50)":"Restaurant20To50"
}
freqList = list()
couponFreqIndex = list(df.columns).index("coupon")
for i in range(df.shape[0]):
  fte = couponDictionary[df.iloc[i, couponFreqIndex]]
  freq = df[fte].iloc[i]
  freqList.append(freq)
df["coupoun_freq"] = freqList
```

```
#drop Bar, CoffeeHouse, CarryAway, RestaurantLessThan20, and Restaurant20To50
df.drop(["Bar", "CoffeeHouse", "CarryAway", "RestaurantLessThan20", "Restaurant20To50"], axis=1, inplace=True)
df.head()
```

- Extract a new feature named to_coupon_time from toCoupon_GEQ15min and toCoupon_GEQ25min, because these both are represented time so can merge them. 0 represents time is less than 15 minutes, 1 represents time is greater than or equal to 15 minutes but less than 25 minutes, and 2 represents time is greater than or equal to 25 minutes.

```
df["to_coupon_time"] = df["toCoupon_GEQ15min"] + df["toCoupon_GEQ25min"]
df.drop(["toCoupon_GEQ15min", "toCoupon_GEQ25min"], axis=1, inplace=True)
df.head()
```

- Find mutual information of features and select those features that have mutual information scores greater than 0.005.

```
threshold = 0.005
selected_features = ["Y"]
columns = df.drop("Y", axis=1).columns
for i in range(len(mi_score)):
  s = mi_score[i]
  f = columns[i]
  if s>threshold:
    selected_features.append(f)
print(selected_features)
```

```
['Y', 'destination', 'passanger', 'weather', 'time', 'coupon', 'expiration', 'gender', 'education', 'direction_same', 'coupoun_freq']
```

```
df = df[selected_features]
df.head()
```

- Convert categorical columns into dummy variables.

```
['Y', 'direction_same', 'destination_No Urgent Place',
 'destination_Work', 'passanger_Friend(s)', 'passanger_Kid(s)',
 'passanger_Partner', 'weather_Snowy', 'weather_Sunny', 'time_10PM',
 'time_2PM', 'time_6PM', 'time_7AM', 'coupon_Carry out & Take away',
 'coupon_Coffee House', 'coupon_Restaurant(20-50)',
 'coupon_Restaurant(<20)', 'expiration_2h', 'gender_Male',
 'education_Bachelors degree',
 'education_Graduate degree (Masters or Doctorate)',
 'education_High School Graduate', 'education_Some High School',
 'education_Some college - no degree', 'coupoun_freq_4~8',
 'coupoun_freq_gt8', 'coupoun_freq_less1', 'coupoun_freq_never'],
```

- Apply smote to solve the class imbalance problem.

- Split data into three parts (train, test, and validation)

```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.2, random_state=42)
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, stratify=y_train, test_size=0.3, random_state=42)
```

- Apply different classifiers and find the AUCROC score.
  - KNN

```
              precision    recall  f1-score   support

           0       0.64      0.68      0.66      1091
           1       0.74      0.71      0.73      1431

    accuracy                           0.70      2522
   macro avg       0.69      0.69      0.69      2522
weighted avg       0.70      0.70      0.70      2522

Cross Validation accuracy  0.6858151378051713
RocAuc Score :  0.741120571655134
```

  - Naive Bayes

	precision	recall	f1-score	support
0	0.61	0.70	0.65	1091
1	0.74	0.66	0.70	1431
accuracy			0.68	2522
macro avg	0.68	0.68	0.67	2522
weighted avg	0.68	0.68	0.68	2522

```
Cross Validation accuracy  0.6755884860009179
RocAuc Score :  0.7339867321794928
```

- Logistic regression

	precision	recall	f1-score	support
0	0.66	0.66	0.66	1091
1	0.74	0.74	0.74	1431
accuracy			0.71	2522
macro avg	0.70	0.70	0.70	2522
weighted avg	0.71	0.71	0.71	2522

```
Cross Validation accuracy  0.6973794068148537
RocAuc Score :  0.7629188948906017
```

- SVM

	precision	recall	f1-score	support
0	0.69	0.69	0.69	1091
1	0.76	0.76	0.76	1431
accuracy			0.73	2522
macro avg	0.72	0.72	0.72	2522
weighted avg	0.73	0.73	0.73	2522

```
Cross Validation accuracy  0.7043396061460452
RocAuc Score :  0.7782261447930818
```

- Random Forest

	precision	recall	f1-score	support
0	0.63	0.65	0.64	1091
1	0.73	0.71	0.72	1431
accuracy			0.68	2522
macro avg	0.68	0.68	0.68	2522
weighted avg	0.68	0.68	0.68	2522

```
Cross Validation accuracy  0.6742596113916027
RocAuc Score :  0.7252182746709147
```

- CatBoost

```
              precision    recall  f1-score   support

           0       0.68      0.68      0.68      1091
           1       0.76      0.75      0.75      1431

    accuracy                           0.72      2522
   macro avg       0.72      0.72      0.72      2522
weighted avg       0.72      0.72      0.72      2522

Cross Validation accuracy  0.7000305990863988
RocAuc Score :  0.7773550317347769
```

- ANN

```
              precision    recall  f1-score   support

           0       0.66      0.68      0.67      1091
           1       0.75      0.73      0.74      1431

    accuracy                           0.71      2522
   macro avg       0.71      0.71      0.71      2522
weighted avg       0.71      0.71      0.71      2522

95/95 [==============================] - 0s 2ms/step - loss: 0.6133 - accuracy: 0.6928
cross validation accuracy :  0.6927651166915894
AucRoc Score :  0.7659424898845199
```

- Select the SVM classifier for the next steps because it has a better AUCROC score and accuracy.
- Tune hyperparameters of SVM using RandmizedSearchCV

```
param_dictionary = {'C': [0.1, 1, 10],
              'gamma': [1, 0.1, 0.01],
              'kernel': ['rbf']
              }
bestParam = RandomizedSearchCV(SVC(random_state=42, probability=True), param_dictionary, refit = True, n_iter=5)
bestParam.fit(X_train, y_train)
print(bestParam.best_estimator_)
y_pred = bestParam.predict(X_test)
scores = cross_val_score(bestParam, X_val, y_val, cv = 10, scoring='accuracy')
print(classification_report(y_test, y_pred))
print("Cross Validation accuracy ", scores.mean())
print("RocAuc Score : ", roc_auc_score(y_test, bestParam.predict_proba(X_test)[:, 1]))
```

```
SVC(C=10, gamma=0.1, probability=True, random_state=42)
              precision    recall  f1-score   support

           0       0.68      0.69      0.68      1091
           1       0.76      0.75      0.76      1431

    accuracy                           0.72      2522
   macro avg       0.72      0.72      0.72      2522
weighted avg       0.73      0.72      0.73      2522

Cross Validation accuracy  0.6934419600900488
RocAuc Score :  0.7724319619067384
```

- Save the model using Joblib and deploy it on the flask.

```
import joblib
joblib.dump(bestParam.best_estimator_,
            'InVehicleCouponRecommendationPredictor.pkl')
```

```
['InVehicleCouponRecommendationPredictor.pkl']
```

Dataset Discussion

This data was collected via a survey on Amazon Mechanical Turk. The survey describes different driving scenarios including the destination, current time, weather, passenger, etc., and then asks the person whether he will accept the coupon.

Basic information about Dataset:

Name	In-Vehicle Coupon Recommendation
Problem	Binary Classification
no: instances	12684
no: features	26
duplicates	74
missing	yes
Classes frequency	1: 7210, 0: 5474
Year	2017

There are 26 attributes.

destination

No Urgent Place	6283
Home	3237

Work 3164

passenger

Alone 7305

Friend(s) 3298

Partner 1075

Kid(s) 1006

weather

Sunny 10069

Snowy 1405

Rainy 1210

temperature

80 6528

55 3840

30 2316

time

6PM 3230

7AM 3164

10AM 2275

2PM 2009

10PM 2006

coupon

Coffee House 3996

Restaurant(<20) 2786

Carry out & Take away 2393

Bar	2017
Restaurant(20-50)	1492

expiration

1d	7091
2h	5593

gender

Female	6511
Male	6173

age

21	2653
26	2559
31	2039
50plus	1788
36	1319
41	1093
46	686
below21	547

maritalStatus

Married partner	5100
Single	4752
Unmarried partner	2186
Divorced	516
Widowed	130

has_children

0	7431
1	5253

occupation

Some college - no degree	4351
Bachelors degree	4335
Graduate degree (Masters or Doctorate)	1852
Associates degree	1153
High School Graduate	905
Some High School	88
Name: education, dtype: int64	
Unemployed	1870
Student	1584
Computer & Mathematical	1408
Sales & Related	1093
Education&Training&Library	943
Management	838
Office & Administrative Support	639
Arts Design Entertainment Sports & Media	629
Business & Financial	544
Retired	495
Food Preparation & Serving Related	298
Healthcare Practitioners & Technical	244
Healthcare Support	242
Community & Social Services	241
Legal	219
Transportation & Material Moving	218
Architecture & Engineering	175
Personal Care & Service	175

Protective Service	175
Life Physical Social Science	170
Construction & Extraction	154
Installation Maintenance & Repair	133
Production Occupations	110
Building & Grounds Cleaning & Maintenance	44
Farming Fishing & Forestry	43

income

\$25000 - \$37499	2013
\$12500 - \$24999	1831
\$37500 - \$49999	1805
\$100000 or More	1736
\$50000 - \$62499	1659
Less than \$12500	1042
\$87500 - \$99999	895
\$75000 - \$87499	857
\$62500 - \$74999	846

car

Scooter and motorcycle	22
Mazda5	22
do not drive	22
crossover	21
Car that is too old to install Onstar :D	21

Bar

never	5197
less1	3482

1~3	2473
4~8	1076
gt8	349

CoffeeHouse

less1	3385
1~3	3225
never	2962
4~8	1784
gt8	1111

CarryAway

1~3	4672
4~8	4258
less1	1856
gt8	1594
never	153

RestaurantLessThan20

1~3	5376
4~8	3580
less1	2093
gt8	1285
never	220

Restaurant20To50

less1	6077
1~3	3290
never	2136
4~8	728

gt8 264

toCoupon_GEQ5min

1 12684

toCoupon_GEQ15min

1 7122

0 5562

toCoupon_GEQ25min

0 11173

1 1511

direction_same

0 9960

1 2724

direction_opp

1 9960

0 2724

Y

1 7210

0 5474

Major Outcomes

- Solve problems of the dataset, apply the feature extraction, and selection.

- Applying different classifiers and selecting the classifier based on the AUCROC score.
- Saving the model and deploying it on the flask.

Project Timeline

Week	Task
1	Data Exploration
2	Preprocessing and Modeling
3	Hyperparameter Tuning, Deployment
4	Testing

Conclusion

For taking business decisions understanding customer behavior is very important. Taking business decisions is a very critical task. In recent years machine learning emerge as a tool for understanding customer behavior and taking business decisions. In this project, we work on a recommendation system that predicts customer behavior that it will accept a coupon of a nearby location such as a restaurant or coffee house or bar, etc based on its context and user and coupon information.

References

1. Quynh, T. D., & Dung, H. T. T. Prediction of Customer Behavior using Machine Learning: A Case Study. In *Proceedings of the 2nd International Conference on Human-centered Artificial Intelligence (Computing4Human 2021). CEUR Workshop Proceedings, Da Nang, Vietnam (Oct 2021).*

2. https://spd.group/artificial-intelligence/ai-for-customer-behavior-analysis/
3. Çelik, E., & Omurca, S. İ. Comparative Analysis of Offline Recommendation Systems with Machine Learning Algorithms. *PROCEEDINGS BOOK*.
4. Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., & MacNeille, P. (2017). A bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, *18*(1), 2357-2393.
5. https://medium.com/@niralidedaniya/in-vehicle-coupon-recommendation-a-machine-learning-classification-case-study-df67e7835703

Abbreviations

SVM: Support Vector Machine
ANN: Artificial Neural Network
KNN: KNearest Neighbour
AUCROC: Area under the ROC Curve
ROC: receiver operating characteristic curve