Capstone Project

Humana-Mays Healthcare Analytics Case Competition 2020

Under the guidance of Dr. Murali Shanker

Srushti Padade
Harika Penjerla

Table of Contents

Introduction

The social determinants of health (SDoH) are the conditions in which people are born, grow, live, work and age that affect a wide range of health risks and outcomes. These circumstances are shaped by the distribution of money, power, and resources at global, national, and local levels. Some of the key concepts in the social determinants are Employment conditions, Social exclusion, Public health programs, Women and gender equity, early child development, Globalization, Health system and Urbanization [3].
Social determinants of health are a key component of Humana Insurance's company integrated value-based health ecosystem. 60% of what creates health has to do with the interplay between our socio-economic and community environments and lifestyle behaviors. Humana is seeking that "broader view "of our members to better understand the whole person and to assist them in new ways towards achieving their best health. In the absence of regular, universal screening for SDoH, Humana needs to utilize data science to understand which members are struggling with SDoH mainly focused on Transportation challenges.

- Transportation screening question is coming from the Accountable Health Communities –Health Related Social Needs Screening Tool.
- The question reads: "In the past 12 months, has a lack of reliable transportation kept you from medical appointments, meetings, work or from getting things needed for daily living?" Yes / No

The goal is to identify members most likely experiencing Transportation Challenges and solutions to overcome this barrier to access care and achieving their best health.
To predict the members with transportation challenges we have used supervised modeling strategies like Ensemble model- Random Forest, XGBoost, Support Vector Machine.

Data Overview

HUMANA has collected it's MAPD (Medicare Advantage Plan) members data. The data is synthetic data, which is used for more robust and demographics details, having a 1-year lookback for a member before event collection. The target variable is transportation challenge which has a binary flag to indicate (like '1' means facing transportation issues and '0' indicates no transportation issues). The main features of the dataset are Medical Claim Features, Pharmacy Claims Features, Lab Claims Features, Demographics, Credit data, Condition Related features, CMS Features, and others.

Important Data Types

- **Medical Claims Features,** includes CCS Procedure Code Categories, BETOS Procedure Code Categories, Utilization by Category (IP admits/ER visits/Outpatient/Ambulance etc.)
- **Pharmacy Claims Features,** consists of Prescription Days Covered, Brand/Generic Prescription, Mailed/Non-mailed Prescription, Maintenance Prescription, GPI2 Level Prescription Utilization
- **Lab Claims Features,** stores Abnormal Lab Results Indicator, Abnormal Lab Results Indicator by Category (Cholesterol/ EGFR/HbA1c/Hemoglobin etc.)
- **Demographics/Consumer Data,** includes Age, Geography, Census Education Level, Household Composition, Homeowner Status, Census Percent Motor Vehicle Ownership
- **Credit data,** including Balance All Mortgage Accounts Past Due, % HH Bank Card Accts - Severe Derogatory Accounts, Number All Mortgage Accts -120 Days Past Due or Collections, % Balance to High Mortgage Credit
- **Condition Related Features** including Behavioral Health Condition Indicator, Charlson Comorbidity Index, Functional Comorbidity Index, Diabetes Complication and Severity Index, CMS Diagnosis Code Categories, MCC Diagnosis Code Categories
- **CMS Features** including Disability, Dual Eligibility, Low Income Subsidy, CMS Risk Score, CMS Total Payment Amount
- **Other features** including Health Program Participation/Status, HEDIS-like Features, Provider Specialty Features, Revenue Code Features, Behavioral Segmentation

Data Exploration

The exploratory data analysis of the data is performed with the help of Google Colaboratory (Colab) using Python programming language. Initially we have imported libraries related to pandas, numpy, sklearn, seaborn and matplotlib in the environment.
Here, we can see the basic information of the data and its memory usage while loading the dataset.

```
<class 'pandas. core. frame.DataFrame'>
RangeIndex: 69572 entries, 0 to 69571
Columns: 826 entries, person_id_syn to submcc_rsk_chol_ind
dtypes: float64(443), int64(361), object (22)
memory usage: 438.4+ MB
```

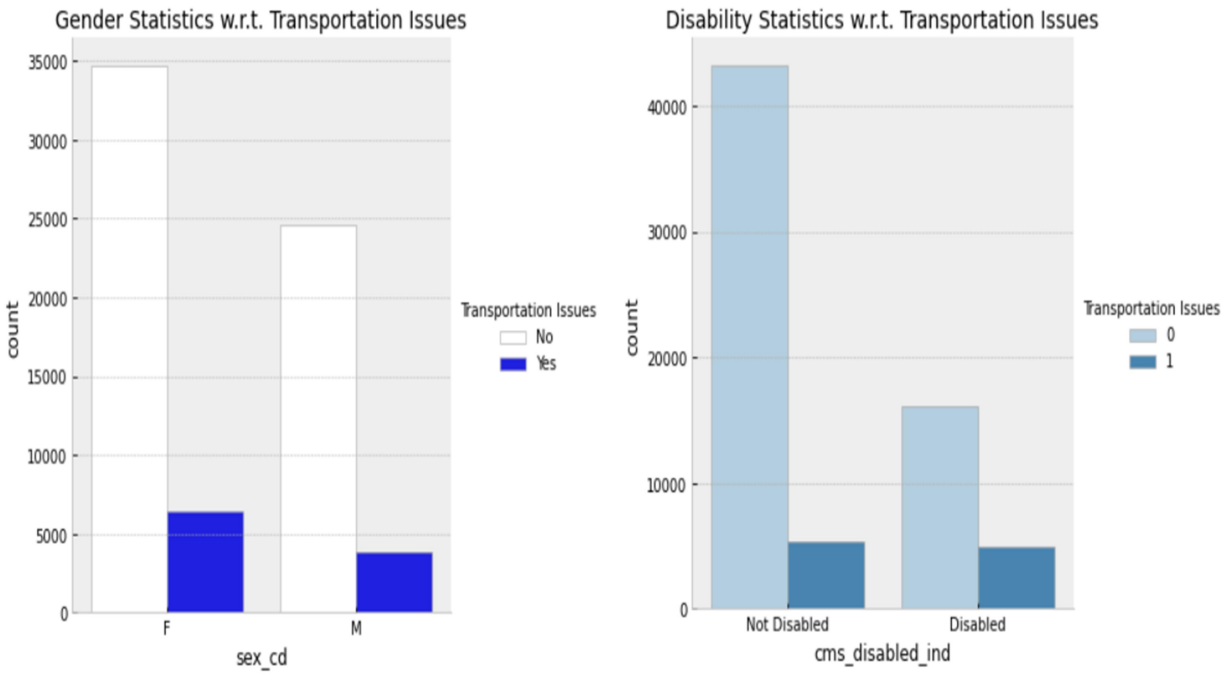

Figure 1: Gender Statistics and Disability Statistics w.r.t. Transportation Issues

Based on the data provided, the comparatively higher transportation issues are faced by the female population than that of male. The possibility of members having transportation issue cannot be determined by their disability metrics. The transportation challenge is faced by **14.66%** members of HUMANA organization.

The dataset contains **69572 observations and 826 features**. The data is divided into the object, integer and float data types. We have observed there are no duplicate entries in the dataset, although there are about **11411 blocks of missing values among the member information which is about 0.96%**. As the null values are randomly distributed throughout the rows and columns, we are not eliminating those data entries.

To gain a better understanding of data for prediction models the necessity is to have a meaningful and complete data. For this purpose, the missing data of numeric data variables are imputed with the median of the respective columns. However, the categorical data variable needs to be handled in a different way. These missing data entries of categorical features are imputed using the most common element method from respective columns. At the end, the categorical data are being factored into dummy variables.

Approach

The modelling portion for predicting the future outcome of transportation challenge is provided in this section. The model suitable to predict or identify who is struggling with transportation issues are classification models.

A machine learning pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative. The purpose of the pipeline is to assemble several steps that can be cross validated together while setting different parameters. We have started with constructing the pipeline for the Logistic Regression model, Random Forest, and Simple Vector machine XGBoost.

Before we start with the modeling there are certain things to be considered such as the important variables, or the factors that could implicate the performance of the model - Feature selection, data balancing.

Feature Selection

Variance Threshold is a simple baseline approach to feature selection. It removes all features whose variance does not meet some threshold. By default, it removes all zero-variance features, i.e., features that have the same value in all samples.
As an example, suppose that we have a dataset with Boolean features, and we want to remove all features that are either one or zero (on or off) in more than 80% of the samples. Boolean features are Bernoulli random variables, and the variance of such variables is given by

$$\mathbf{Var|X| = p(1-p)}$$

Applying the Variance threshold method of feature selection, the new variable set consists of 730 features. Once the important features are selected the very next step is to divide the dataset into the training data and validation data. We have split the dataset into 80% training and 20% validation dataset using randomization or stratification.

```
X Training Dataset Shape:  (55657, 729)
Y Training Dataset Shape:  (55657,)
X Validation Dataset Shape: (13915, 729)
Y Validation Dataset Shape: (13915,)
```

Data Balancing

As the statistics about the data demonstrates the uneven distribution of transportation issues faced by the members, it is important to have a blanched dataset to exhibit the information and avoid a biased decision we could make. One approach to addressing imbalanced datasets is to oversample the minority class.
The simplest approach involves duplicating examples in the minority class, although these examples do not add any new information to the model. Instead, new examples can be synthesized from the existing

examples. This is a type of data augmentation and is referred **as Synthetic Minority Oversampling Technique-SMOTE** [2] for short.

Our method of synthetic oversampling works to cause the classifier to build larger decision regions that contain nearby minority class points. Hence, after applying the SMOTE balancing technique we end up having the dataset with equal amounts of data with members facing and not facing transportation issues of 47499 members each.

Experimental Design Models

With all considerations in place, the goal is simple: Build a model to identify Medicare members most at risk for a Transportation Challenge with least cost and time.

Foundation

The code for this experiment was developed with Python in Jupyter Notebook and Google Colaboratory, along with the libraries such as pandas, numpy, sklearn, seaborn and matplotlib.

Modeling

To begin with any modeling strategies, the very first model built is simple Logistic Regression for classification problems. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

The pipeline model of Logistic Regression came up with the estimator value as shown in appendix.
Next, we tried the Random Forest Ensemble model. A random forest is a Meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

“Support Vector Machine” (SVM) is a supervised algorithm, where it plots each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, perform classification by finding the hyper-plane that differentiates the two classes very well.

The final model we used is XGBoost Classifier. XGBoost [1] is an implementation of a gradient boosted decision tree for speed and performance. Here we have used the XGBoost Classifier coupled with SMOTE- balancing method to predict the outcome of transportation issues.

```
XGBClassifier (base_score=0.5, booster='gbtree', colsample_bylevel=1,
     colsample_bynode=1,    colsample_bytree=1,    gamma=0,    learning_rate=0.1,
     max_delta_step=0,     max_depth=3,     min_child_weight=1,     missing=None,
     n_estimators=100,  n_jobs=1,  nthread=None,  objective='binary:  logistic',
     random_state=0,reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
     silent=None, subsample=1, verbosity=1) array ([[249,2],[ 35,  4]])precision
recall  f1-score   support
```

Results

As working through the modeling, we have made certain observations with each model and with the data preprocessing. Also, the time consumptions with each model are significantly variable. The below table defines findings with each modeling:

Models	**Training Accuracy**	**Validation Accuracy**
Logistic Regression	85.6	84
Random Forest	86.5	86.6
Support Vector Machine	86.4	86.6
XGBoost	70	85
XGBoost with SMOTE	95.4	91

Table 1: Model Accuracies

Looking at the accuracy of each model we could see that Random forest, SVM and Logistic Regression are having almost the same accuracy with approximately 85%. However, The XGBoost have shown a greater deflection with the training accuracy than the others with 95% accuracy. Although the validation accuracy is 10% lesser than training data, it displays the overfitting of the data. However, we have solved this problem with balancing the dataset.
Using XGBoost as our final model we can see the below results with and without balancing the dataset.

```
array([[11690,   186],
       [ 1840,   199]])
```

	precision	recall	f1-score	support
0	0.86	0.98	0.92	11876
1	0.52	0.10	0.16	2039
accuracy			0.85	13915
macro avg	0.69	0.54	0.54	13915
weighted avg	0.81	0.85	0.81	13915

The validation accuracy without balancing the data is as above. However, the validation accuracy with balancing the data is as displayed below with 91% accuracy.

```
array([[11690,   186],
       [ 1860, 10016]])
```

	precision	recall	f1-score	support
0	0.86	0.98	0.92	11876
1	0.98	0.84	0.91	11876
accuracy			0.91	23752
macro avg	0.92	0.91	0.91	23752
weighted avg	0.92	0.91	0.91	23752

Thus, using the XGBoost with SMOTE balancing as our final model we have predicted the test dataset outcome i.e., the members having transportation issues among them.

Conclusion

We have a list of 17681 members of HUMANA. We could predict that about **430** members are facing the transportation challenge. Among these members the **60% population are female**. Also, the members facing the transportation challenge are mostly of the **age above 55 years**. Considering the medicare segmentation of the members, the people with **Self-Engaged Optimists** and **Auto-Pilot Participators** are mostly prone to having transportation issues.

Future Scope & Recommendations

Looking at the predicted statistical parameters there are few recommendations that could be made. Firstly, as the female members are more prone to face transportation challenges, we recommend having online appointments where possible so the travelling could be avoided. Also, similar could be suggested for the members with the age above 50 years. This is possible when the insurance could provide the coverage for the online appointments. The patients categorized as Self engaged optimists and auto pilot participants could be helped by further breaking them down based on their absolute necessity for the House based visits by the Hospital persona.

Acknowledgement

Special thanks owed to Dr. Murali Shanker for his unequivocal guidance and student-focused instruction across the wide field of Machine Learning and Visualization. We are thankful to Dr. Razavi, Dr.Wu, and other professors for all the teachings in the Business Analytics course and my fellow batch mates for all the help and guidance and support.

References

[1] https://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/

[2] https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[3] https://www.who.int/social_determinants/sdh_definition/en/

[4] https://press.humana.com/news/news-details/2020/Texas-AM-University-Humana

Competition Feedback

Judgement based on the following criteria:

- Quantitative analysis identifying key business insights
- Professionalism, data visualization, and presentation skills
- Ability to provide meaningful implications and recommendations based on results/insights

Send us an email mentioning our team is not selected for the second round and Thank you for participation.

Appendix

Model Specifications in Coding:

1. Logistic Regression

Below are the logistic regression estimator values

```
Performing model optimizations...

Estimator: Logistic Regression
GridSearchCV (cv=5, error_score=nan, estimator=Pipeline(memory=None,
    steps=[('scl', StandardScaler(copy=True, with_mean=True, with_std=True)),
('clf', LogisticRegression(C=1.0, class_weight=None, dual=False,
fit_intercept=True, intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2', random_state=42,
solver='lbfgs', tol=0.0001, verbose=0, warm_start=False))], verbose=False),
iid='deprecated', n_jobs=None,
param_grid= [{'clf__C': [1.0, 0.5, 0.1],
             'clf__penalty': ['l1', 'l2'],
             'clf__solver': ['liblinear']}],
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='accuracy', verbose=0)

Best params: {'clf__C': 0.1, 'clf__penalty': 'l1', 'clf__solver':
'liblinear'}
```

2. Random forest

Below are the Random forest model estimator values calculated for the modeling.

Estimator: Random Forest

```
GridSearchCV (cv=5, error_score=nan, estimator=Pipeline(memory=None,
      steps=[('scl', StandardScaler(copy=True, with_mean=True,
with_std=True)), ('clf', RandomForestClassifier(bootstrap=True,
ccp_alpha=0.0, class_weight=None, criterion='gini', max_depth=None,
max_features='auto', max_leaf_nodes=None, max_samples=None,
min_impurity_decrease=0.0, min_impurity_split=None… random_state=42,
verbose=0, warm_start=False))], verbose=False), iid='deprecated', n_jobs=-1,
      param_grid= [{'clf__criterion': ['gini', 'entropy'],
      'clf__max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
      'clf__min_samples_leaf': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
      'clf__min_samples_split': [2, 3, 4, 5, 6, 7, 8, 9,10]}],
       pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
       scoring='accuracy', verbose=0)
```

```
Best params: {'clf__criterion': 'gini', 'clf__max_depth': 9,
'clf__min_samples_leaf': 1, 'clf__min_samples_split': 4}
```

3. Support Vector Machine

Implemented the SVM for our prediction and estimator values as per the pipeline is as below.

```
Estimator: Support Vector Machine

GridSearchCV (cv=5, error_score=nan, estimator=Pipeline(memory=None,
      steps=[('scl', StandardScaler(copy=True, with_mean=True,
with_std=True)), ('clf', SVC(C=1.0, break_ties=False, cache_size=200,
class_weight=None, coef0=0.0, decision_function_shape='ovr', degree=3,
gamma='scale', kernel='rbf', max_iter=-1, probability=False, random_state=42,
shrinking=True, tol=0.001, verbose=False))], verbose=False),
iid='deprecated', n_jobs=-1,
      param_grid= [{'clf__C': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'clf__kernel':
['linear', 'rbf']}],  pre_dispatch='2*n_jobs', refit=True,
return_train_score=False, scoring='accuracy', verbose=0)

Best params: {'clf__C': 1, 'clf__kernel': 'rbf'}
```