Capstone Project

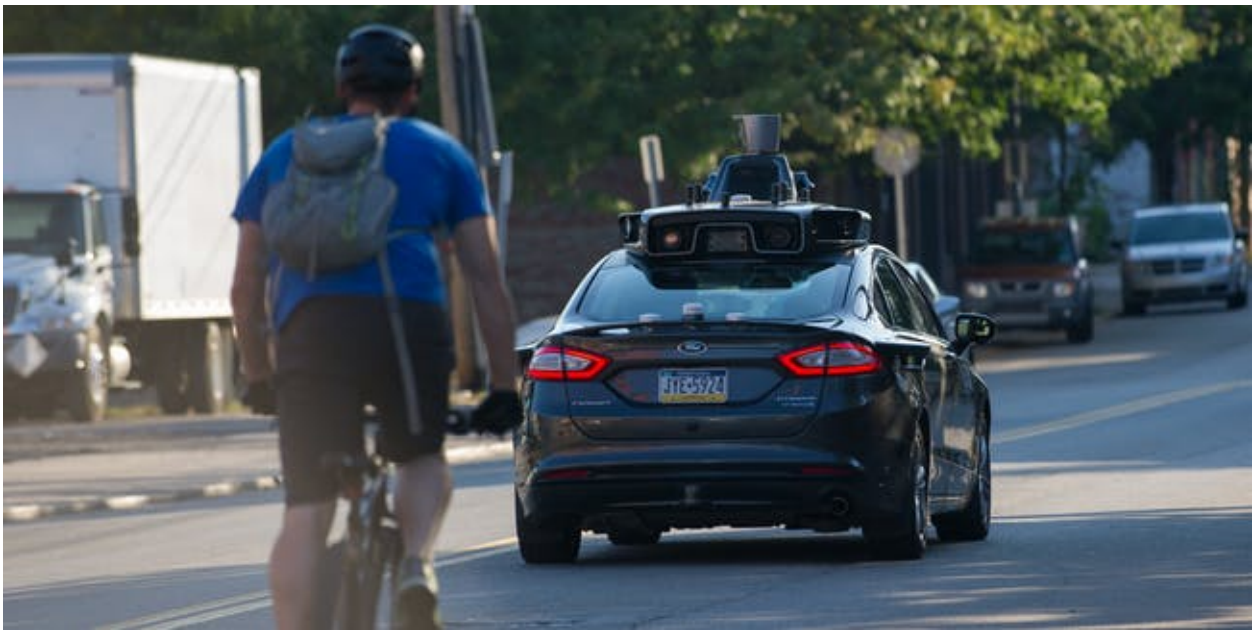
Udacity

Deployment of Autonomous Vehicles

Machine Learning Engineer Nanodegree

by Ramil Sharifsoy

June 18, 2017



Index

Definition

Project Overview

Topics like efficiency of marketing initiatives, revenue increase, and improvement in customer satisfaction were subject for many recent research and white papers. David Levis et al. proposed automation of customer interaction analysis utilizing machine learning on their paper "Concurrent Reinforcement Learning from Customer Interactions" [1]. White paper "Machine Learning for e-mail marketers" by Boomtrain [2] explains benefits of autonomous email marketing. Another paper, "How machine learning helps sales success" by Cognizant [3] explains how machine learning can be utilized to increase sales and enhance customer satisfaction. As an engineer in MBA program, I had a question to answer, how can I apply developments in technologies that I am excited about to promote that technology and make it successful? Especially to autonomous car industry. Survey conducted by BikePGH about interaction of pedestrians and bicyclists with autonomous cars in Pittsburgh, PA became handy and just in time for my project [6].

Understanding opinions of experienced citizens about AVs will assist AV companies in selecting their next cities and neighborhoods effectively for deployment and improve efficiency of sales and marketing strategies.

Problem Statement

Uber deployed autonomous vehicles fleet in Pittsburgh, PA in September 2016. First question to answer is, why Pittsburgh? There are many official explanations about how city was picked, but none of the stories involve public opinion of those who live in that city. People who live in the areas of deployment are end users of the autonomous cars, and they should be part of the decision making process.

Next question to answer is, which city is suitable for next deployment of AVs after Pittsburgh? By analyzing experience in Pittsburgh we can predict the best city for next business expansion. BikePGH`s survey results provide information about opinions of residents in certain zip codes around Pittsburgh. Publicly available information helps to describe type of people in those zip codes. Combined information from these two sources creates answer to question, who did approve AV deployment? By knowing approver's income level, value of his house and other demographics about his neighborhood we can predict best deployment areas around the country.

Supervised learning works with datasets where inputs, or features, and outputs, or targets, are known. Dataset for this problem is mixed, some features indicate category some indicate actual numbers. There are three classification algorithms which works with this kind of datasets, Decision Tree, Naïve Bays, and Support Vector Machines. All three algorithms were tested on dataset and SVM was chosen for final modeling.

Metrics

Dataset composed from survey results expected to be imbalanced. In this case one of the labels dominates almost half of the results and the rest of the labels share other half of the results. F1 score is a good performance metric for similar cases as it takes into account precision and recall in one formula. Precision is how many correct answers are among all answers, recall is how many of all correct answers was identified. The weighted harmonic mean of these two metrics makes F1 score.

$$F = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \qquad \mathrm{Precision} = \frac{tp}{tp + fp} \qquad \mathrm{Recall} = \frac{tp}{tp + fn}$$

Analysis

Data Exploration

There are 938 individual survey responses in dataset. Training and testing split is at 700/238, maintaining approximate 75% - 25% ratio.

Inputs data will be composed from following datasets:

1. [6] Autonomous Vehicle Survey of Bicyclists and Pedestrians in Pittsburgh, 2017
2. [7] Median Household Income in Pittsburgh, PA by Zip Code
3. [12] OASDI Beneficiaries by State and ZIP Code, 2014
4. [13] Median Home Value – Zillow Home Value Index (ZHVI)

Input dataset table head view and column descriptions shown below.

	PED	BIC	AVS	FTC	ZCD	POP	AHI	ZHV	ALB	APP
0	1	1	0	2	15201	14326	27031	129000	2920	0
1	1	0	0	1	15201	14326	27031	129000	2920	0
2	1	0	2	1	15201	14326	27031	129000	2920	1
3	1	2	2	4	15201	14326	27031	129000	2920	0
4	0	1	1	3	15201	14326	27031	129000	2920	3

In order to improve processing of data and shorted coding, data conversions were carried during dataset building process. Explanation of conversions shown under description lines below.

Features Columns (Demographics and Survey Questions):

1. PED: Pedestrians interacted with an AV using sidewalks and crosswalks

   `0 = No, 1 = Yes, 2 = Not Sure`

2. BIC: Bicyclists interacted with an AV while riding your bicycle

   `0 = No, 1 = Yes, 2 = Not Sure`

3. AVS: Safety of Autonomous Vehicles

   `0 = Very Unsafe to 4 = Very Safe`

4. FTC: Familiarity with AV technology

   `0 = Not familiar, 1 = Mostly unfamiliar,`
   `2 = Somewhat familiar, 3 = Mostly familiar, 4 = Extremely familiar`

5. ZCD: Zip code of residence

6. POP: Population of zip code
7. AHI: Average household income in the zip code
8. ZHI: Median home value in the zip code
9. ALB: Number of social security beneficiaries in the zip code

Target Column

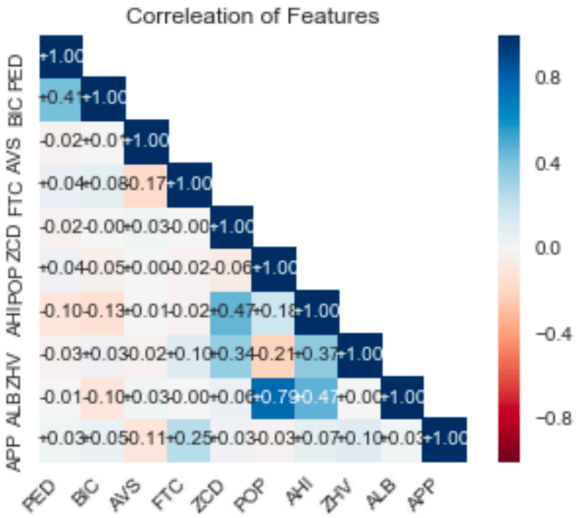
1. APP: Approval

   `0 = Disapproval, 1 = Somewhat disapproval,`

   `2 = Neutral, 3 = Somewhat approval, 4 = Approval`

Exploratory Visualization

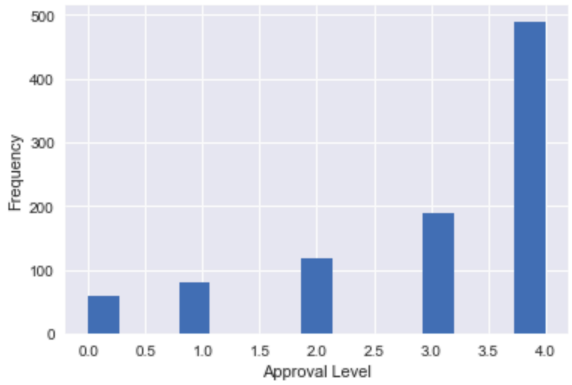
Correlation of features chart below shows that features are not significantly correlated with each other and all can be kept, as they can contribute to decision making process. Population has 0.79 correlation value with All Beneficiaries, which seems to be normal, and it would be acceptable to remove one of these factors. It was decided to keep both in analysis because current data set refers to one metropolitan area, in the future this correlation may change as new cities included in analysis.



Approval level shows that 52% of participant approved AV presence on the streets, and 20% somewhat approved. In total 72% of participants are positive about AVs in the city. Most of the participant are not confident about their knowledge about AVs, which was expected. They make

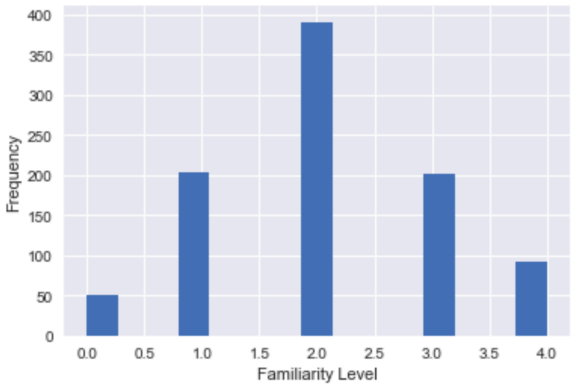
up about 83%. Slightly under 20% has confidence in their AV knowledge. Following histograms show more details about approval and familiarity levels.

Approval Level Histogram



58, %6 disapproving AV
81, %8 somewhat disapproving AV
119, %12 neutral with AV
190, %20 somewhat approving AV
490, %52 approving AV

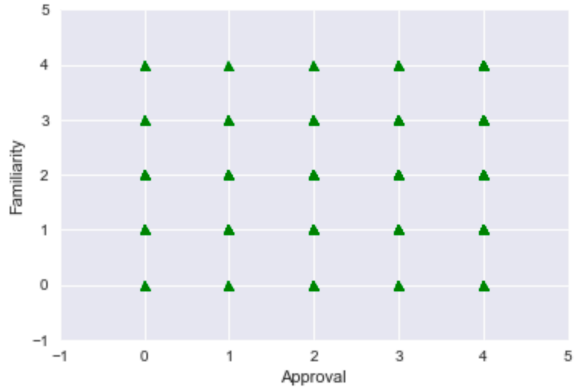
Familiarity level Histogram



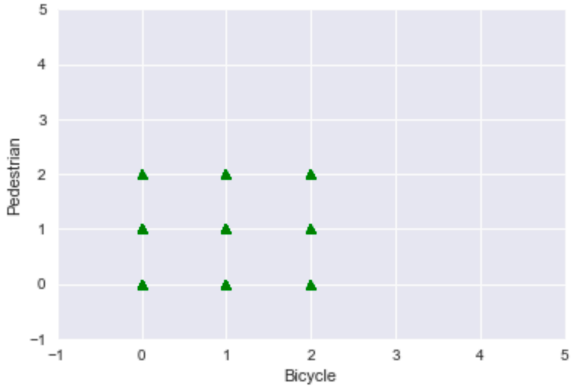
51, %5 not familiar with AV
203, %21 mostly unfamiliar with AV
391, %41 somewhat familiar with AV
201, %21 mostly familiar with AV
92, %9 extremely familiar with AV

Converted features were reviewed against each other to check if there are any significant anomalies to consider. Two of the comparisons shown below, Approval vs. Familiarities and Bicycle vs. Pedestrian. As it is seen, there are not any unusual behavior in data relations, and they are well diverse.

Approval vs Familiarity

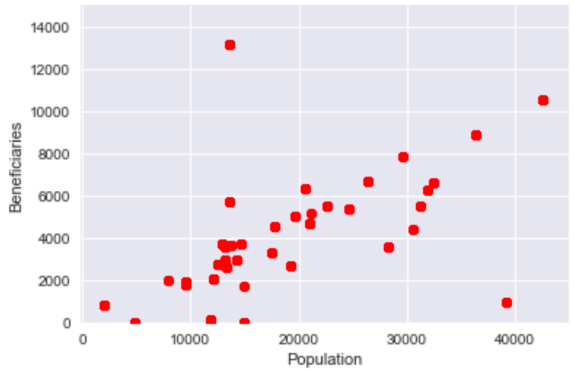


Bicycle and Pedestrian

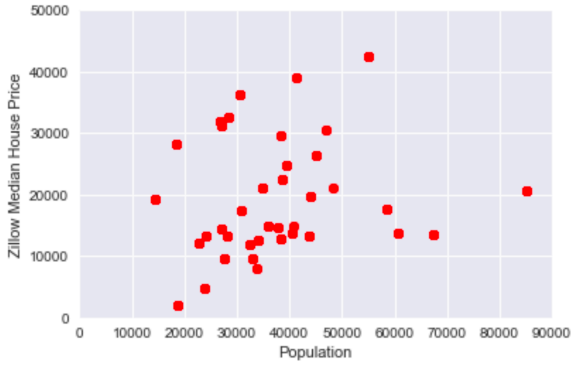


Strong linear correlation, with few outliers, between Population and Beneficiaries, and lack of strong correlation between Population and Median Household Income can be seen on the graph below, endorsing correlation of features chart.

Population & Beneficiaries

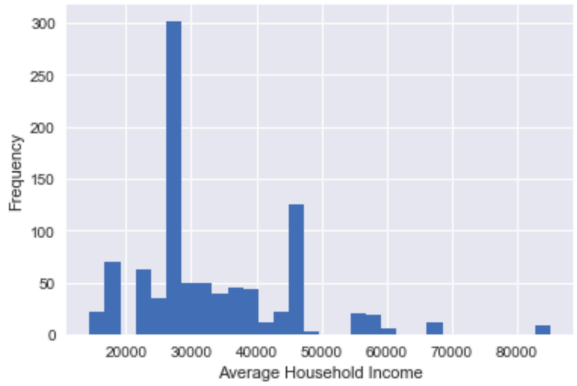


Population & Median Household Income

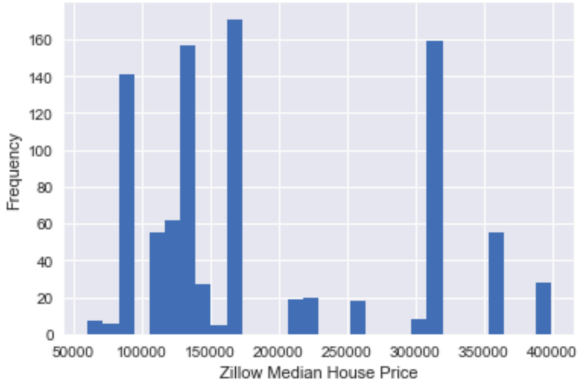


Average house hold income and median house price of participants was examined closely. As it is seen on histograms below, data is not perfect as it is limited in numbers, just 938 responses. Even with this size, at this point this data provides valuable information, with more surveys quality of data will increase and histograms below should fit normal distribution, or some other distribution.

Average Household Income



Median House Price



Algorithms and Techniques

It was important to find algorithm which will perform well with small data. Following three methods were used to identify which will perform better on given dataset:

Decision Trees. DTs are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Some advantages of DTs are: Trees can be visualized. Requires little data preparation. Able to handle both numerical and categorical data. Able to handle multi-output problems. Uses a white box model. Easily explained by Boolean logic. Possible to validate a model using statistical tests. Performs well even if its assumptions are somewhat violated by the true model from which the data were generated. Few disadvantages of DTs include: Does not support missing values. DT learners can create overfitting. Small variations in the data might result in a completely different tree being generated. DTs do not express all problems easily, such as XOR, parity or multiplexer problems. DT learners create biased trees if some classes dominate.

Gaussian Naive Bayes algorithm based on conditional independence (CI) assumption. It is the simplest form of Bayesian network, in which all attributes are independent given the value of the class variable. CI makes it one of the most efficient and effective learning algorithms but independence is rarely true in real world. In fact, there is a local dependency. With Gaussian Naive Bayes algorithm for classification, the likelihood of the features is assumed to be Gaussian. Some advantages of NB: NBs can handle missing data. NBs does not need a lot of data to perform well. Feature selection is the selection of those data attributes that best characterize a predicted variable. Calculate the probabilities for each attribute is very fast. Few disadvantages of NB: Strong independence assumption of features. Frequentist approach in case of data scarcity. The performance of NBs can degrade if the data contains highly correlated features. If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction. Support Vector Machines.

Support Vector Machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier's detection. SVMs are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. The advantages of support vector machines are: Effective in high dimensional spaces, even if the number of dimensions is greater than the number of samples. Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient. Different Kernel functions, including common and custom, can be specified for the decision function. The disadvantages of support vector machines include: If the number of features is much greater than the number of samples, the method is likely to give poor performances. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation. Perhaps the biggest limitation of the support vector approach lies in choice of the kernel. Another limitation is speed and size, both in training and testing. The most serious problem with SVMs is the high algorithmic complexity and extensive memory requirements of the required quadratic programming in large-scale tasks.

Benchmark

System Usability Score (SUS) is used for measuring perception of usability [10]. It is applicable to this project as we are measuring customer reaction to use of a new product. SUS score was used as a benchmark for this project. Average reported SUS score is 68 [11], which corresponds to 50th percentile, which has estimated comparison rate of 67%. Based on this number, F1 score of 0.67 was set as base limit to accept algorithm.

Methodology

Data Preprocessing

Data was split for training and testing at 700/238, maintaining approximate 75/25% ratio.

```
# Set the number of training points
num_train = 700

# Shuffle and split the dataset into the number of training and testing points
X_train, X_test, y_train, y_test = train_test_split(
    X_all, y_all, train_size=num_train)

# Show the results of the split
print "Training set has {} samples.".format(X_train.shape[0])
print "Testing set has {} samples.".format(X_test.shape[0])
```

```
Training set has 700 samples.
Testing set has 238 samples.
```

It is important to know how long it takes to complete the task for an algorithm with given dataset. Few time keeper functions around implemented algorithms will be helpful at this stage, and they will assist in choosing best algorithm to solve the problem. Following functions were initiated for this purpose:

```
# Create functions
def train_classifier(clf, X_train, y_train):
    # Fits a classifier to the training data

    # Start the clock, train the classifier, then stop the clock
    start = time.time()
    clf.fit(X_train, y_train)
    end = time.time()

    # Print the results
    print "Trained model in {:.4f} seconds".format(end - start)


def predict_labels(clf, features, target):
    # Makes predictions using a fit classifier based on F1 score

    # Start the clock, make predictions, then stop the clock
    start = time.time()
    y_pred = clf.predict(features)
    end = time.time()

    # Print and return results
    print "Made predictions in {:.4f} seconds.".format(end - start)
    return f1_score(target.values, y_pred, pos_label='1', average='weighted')


def train_predict(clf, X_train, y_train, X_test, y_test):
    # Train and predict using a classifer based on F1 score

    # Indicate the classifier and the training set size
    print "Training a {} using a training set size of {}. . .".format(clf.__class__.__name__, len(X_train))

    # Train the classifier
    train_classifier(clf, X_train, y_train)

    # Print the results of prediction for both training and testing
    print "F1 score for training set: {:.4f}.".format(predict_labels(clf, X_train, y_train))
    print "F1 score for test set: {:.4f}.".format(predict_labels(clf, X_test, y_test))
```

Key to improve chosen model was to set up classifier parameters in a way that model increases F1 score. Selecting parameters for SVM classifier consumed significant time compared to overall project execution time. Another challenge was to set training and testing sample sizes, it is

important to set numbers randomly and effectively so that results are meaningful and useful. Another challenge was in choosing classifier, for example design tree seems to be providing best results fast and easy, but it has some downsides which is risky to take.

Implementation

Three models were chosen and implemented on three different split points. It helped to visualize performance of algorithms. Three train and test splits points of data are 200/738, 500/438, and 700/238. Models are Decision Tree, Gaussian NB, and SVM.

```
# Initialize three models
clf_A = tree.DecisionTreeClassifier()
clf_B = GaussianNB()
clf_C = svm.SVC()

# Set up training set sizes
X_train_200 = 200
y_train_200 = 200

X_train_500 = 500
y_train_500 = 500

X_train_700 = 700
y_train_700 = 700
```

Results were compared based on implementation time and F1 score. These values influenced final method selection decision along with classifier pros and cons. SVM was chosen as classifier. One of the results shown below:

```
 Model 3 700: SVC
Training a SVC using a training set size of 700. . .
Trained model in 0.0273 seconds
Made predictions in 0.0079 seconds.
F1 score for training set: 0.4667.
Made predictions in 0.0029 seconds.
F1 score for test set: 0.4000.
```

Refinement

Grid search over SVM parameters improved F1 train score and passed initial base limit of 0.67. Following set of parameters were fed to classifier which resulted in train F1 score of 0.8204.

```
# Parameters list to tune
parameters = [{'C': [1, 10, 100, 200, 300, 400, 500, 600, 700, 1000, 20000],
               'gamma': [1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7],
               'kernel': ['rbf'], 'tol':[1e-3, 1e-4, 1e-5, 1e-6, 1e-7]
              }]

# Initialize the classifier
clf = svm.SVC()

# Make an f1 scoring function
f1_scorer = make_scorer(f1_score, pos_label='1', average='macro')

# Perform grid search on the classifier using the f1_scorer as the scoring method
grid_obj = GridSearchCV(clf, parameters, scoring = f1_scorer)
```

Test F1 score showed small improvement even with grid search. It only came to 0.4461. This is probably due to size of the dataset.

Results

Model Evaluation and Validation

Table below shows that Decision Tree is the best fit without any further tuning. Other two methods show low performance. But, decision tree is not best algorithm with missing data.

Classifier 1: Decision Tree

Training Set Size	Training Time	Prediction Time	F1 Score (train)	F1 Score (test)
200	0.0008	0.0002	0.9453	0.4412
500	0.0012	0.0002	0.8662	0.3766
700	0.0019	0.0004	0.8354	0.4247

Classifier 2: Gaussian NB

200	0.0008	0.0006	0.4028	0.3671
500	0.0007	0.0004	0.4049	0.3768
700	0.0009	0.0008	0.3837	0.3662

Classifier 3: SVM

200	0.0026	0.0008	0.5709	0.4134
500	0.0098	0.0046	0.5062	0.3996
700	0.0273	0.0079	0.4667	0.4000

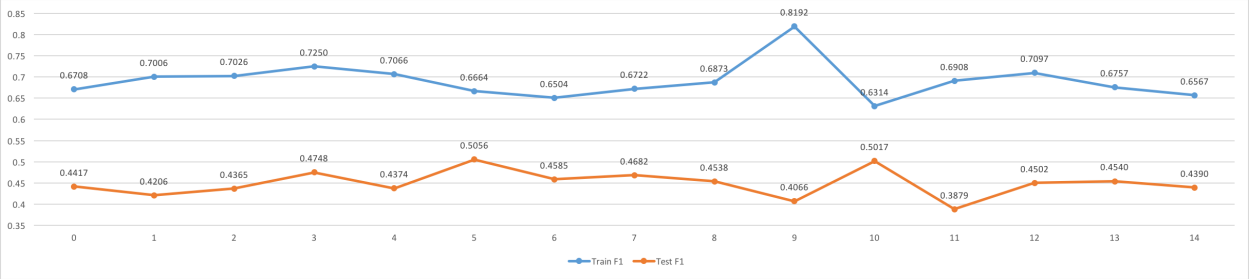
Future train and test data, which is incoming data from new survey results, is not expected to be complete, simply because it is a survey data and expected to have missing sections.

Therefore, one of other two algorithms will be appropriate for further tuning. SVM has better F1 train score than Gaussian NB, therefore SVM assumed to be more convenient to tune and reach target 0.67 F1 score. Tuned SVM brought train F1 score to 0.8204 but did not improve test F1, it was reduced. To improve test f1 score we need more data.

To validate tuned SVM model, we run same model 15 times with different train and test sets keeping split ratios at approximately 75/25%, 700/238 samples respectively. With average F1 score of 0.69, results stayed around benchmark for train sets, varying between 0.6314 and 0.8192. Section of code with tabulated results shown below:

```
# Validate model over different splits, run 15 times
for i in range(15):
    # Set the number of training points
    num_train_2 = 700

    # Shuffle and split the dataset into the number of training and testing points
    X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(
        X_all, y_all, train_size=num_train_2)
```



Justification

Final choice provides reliable solution based on training sets, but not on testing sets. Significant difference between testing score and training score indicates that more data required. This gap can be closed with further surveys and data collection. Developed model may need adjustment with addition of new data.

Conclusion

Reflection

Purpose of this project is to create a tool that can increase effectiveness and efficiency of AV deployment. One of the solutions is to conduct surveys and collect data from citizens of Pittsburgh and analyze demographics of participants. Create train and test datasets based on publically available demographics data of ZIP codes of participants combined with survey results. Further train an algorithm which can predict readiness of target city for AV deployment. Data flow illustrated on figure 1.

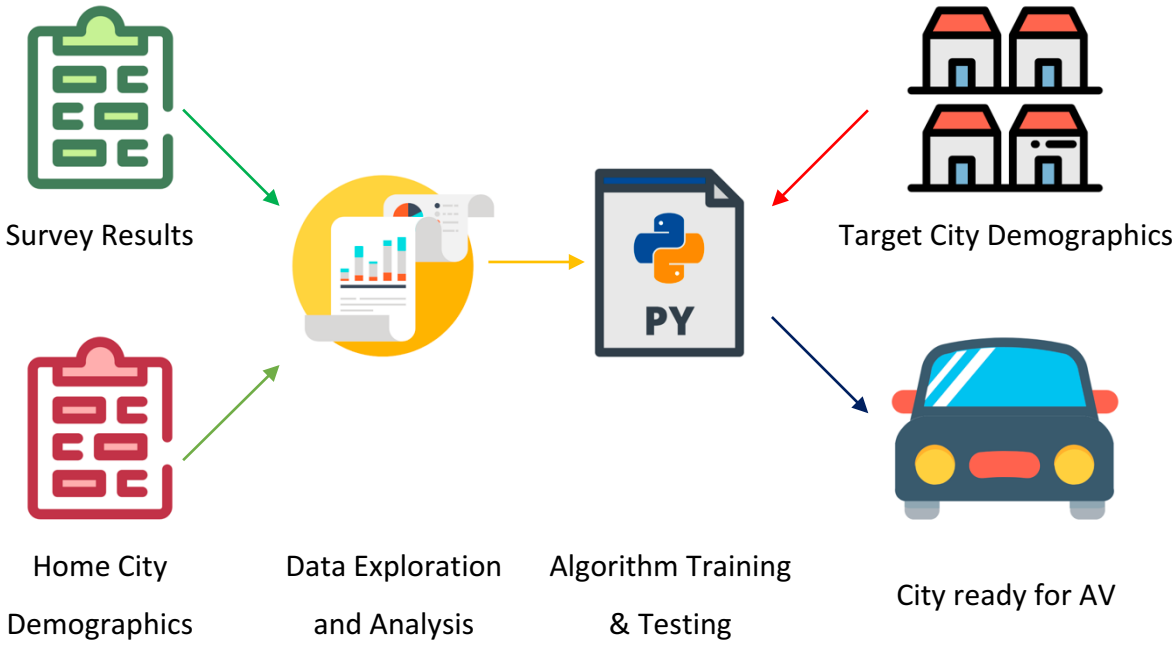


Figure 1: Data Flow

Trained and fine-tuned algorithm can be used to create heat map of entire country. Decision maker will need either that map, or an application where he can dial in demographics data of a target zip code and see what are the predicted chances of success in deploying AV in that area.

Improvement

There is a challenge with this solution, it depends on opinions of end users. Opinions of consumers influenced by factors that are not easy to formulate or predict. Opinions of end users will change with time and with expansion of AV deployment. Therefore, this process is iterative and should be conducted continuously until majority of users becomes familiar with AVs. This iteration may require changing set parameters or even replacing created model.

Testing accuracy of algorithms were in the rage of 0.40 which is lower than the benchmark. This result was expected with given dataset because of its small size. To improve performance of testing score it is necessary to collect more data either from Pittsburgh area or from other regions in the country where autonomous vehicles roam around the streets.

References

[1] David Silver, Leonard Newnham, David Barker, Suzanne Weller, Jason McFall; Proceedings of the 30th International Conference on Machine Learning, PMLR 28(3):924-932, 2013. http://proceedings.mlr.press/v28/silver13.html

[2] Machine Learning for Email Marketers, https://boomtrain.com/ebook-machine-learning-for-email-marketers/

[3] How Machine Learning Helps Sales Success, https://www.cognizant.com/de-de/pdf/Machine_Learning.pdf

[4] Icons: http://www.flaticon.com/

[5] Cover Photo: Autonomous Vehicles May Incite Reckless Human Driving, https://www.inverse.com/article/22374-self-driving-cars-reckless-humans

[6] Survey Data: Autonomous Vehicle Survey of Bicyclists and Pedestrians in Pittsburgh, 2017, https://catalog.data.gov/dataset/autonomous-vehicle-survey-of-bicyclists-and-pedestrians-in-pittsburgh-2017

[7] Pittsburg area income data: http://zipatlas.com/us/pa/pittsburgh/zip-code-comparison/median-household-income.htm

[8] Supervised Learning Workflow and Algorithms https://www.mathworks.com/help/stats/supervised-learning-machine-learning-workflow-and-algorithms.html#bswluh9

[9] Building a Student Intervention System https://github.com/ramilsharifsoy/Machine_Learning_ND/blob/master/P2_Academic_Intervention/student_intervention.ipynb

[10] A Method to Standardize Usability Metrics into a Single Score https://www.measuringu.com/papers/p482-sauro.pdf

[11] 10 Benchmarks for User Experience Metrics https://measuringu.com/ux-benchmarks/

[12] OASDI Beneficiaries by State and ZIP Code, 2014 https://catalog.data.gov/dataset/oasdi-beneficiaries-by-state-and-zip-code-2014

[13] Median Home Value – Zillow Home Value Index (ZHVI) https://www.zillow.com/research/data/#median-home-value