Machine Learning Engineer Nanodegree

Capstone Project : Predicting US High School Graduation Success

Author: jtmoogle @github.com Email: jtmoogle@gmail.com
Date: May 11, 2018

I. Definition

The purpose is to predict the success if US High School Graduates can meet 90% graduation rate in 2020[1]. In this proposal, we leveraged the publicly available data sources for AT&T 2015 'Data for Diplomas' Hakathon[2], and used machine learning/ML techniques listed in Table 1.

Table 1. Classification and Regression algorithms to evaluate problems

Problem Statement	Pattern ML	Predictive Model
1. **Prediction if achieving 90% graduation goal**	Supervised Learning	Classification model
2. **Prediction of graduation rate**	Supervised Learning	Regression model

Tasks were performed below

- Modeled and predicted individual performance of school district
- Compared predictive accuracy of different algorithms
- (Stretch target) If the feature is suitable for classification, a further comparative analysis was done that would be reusable/useful for regression.

Goal was to explore insights, and understand the factors related to success and failure, so we could propose actionable plans to increase the U.S. high school graduation rate reaching 90% by 2020[1]

Because we had very large number of features, we preprocessed and performed feature selections to extract those features that were really important rather than fitting a complete feature set of predictive models.

The inputs were chosen by performing various feature selection algorithms for classification and regression, respectively. As a result of feature selections, we obtained less features which were very important for fitting predictive models.

Through the feature selection, we allowed the ML algorithms to train faster, reduce training time and evaluation time, reduce the complexity of predictive models, improve the accuracy of models, and reduce overfitting.

Project Overview

Children are our future, higher education bring children better fortune. Our kids could benefit from the goal of "Increase high school graduation rate in 2020 to 90% with a "better future."

For decades, the high school graduation rate increased from 71.7% in 2001 to 81.4% in 2013 nationally [2]. People used a variety of machine learning techniques and predictive analytics, so they gain actionable insights and improve educational opportunities early. An example from Tacoma public schools in Washington State, which uses the ML models to predict the risk of dropping out from schools, so teachers could intervene early to help students at risks. In doing so, graduation rates increased from 55% to 78% by 2014[4].

By 2020, we need more 9% (about 310,000 more) graduates to meet our goal of 90% on-time graduation rate by 2020. Students in low-income, minority, and special education students, big cities/big districts, and big states seems struggle and challenge to reach graduation rate 90% on-time [2].

Problem Statement

This proposal used the supervised ML techniques of classification and regression models to evaluate problems below:

1. Prediction if high schools have a success or failure of 90% or more graduates on-time This was in supervised *classification* process.
2. Prediction of high school graduation rate This was in supervised *regression* process. Goals
3. Explore potential factors or insights: significant impact of features on the graduation rates, and seek potential linkages
4. Understand factors related to success and failure: evaluate the performance and predictive power of fitting models in the school district level

Experimental Tasks

- Feature selection: identify which predictor (feature/variables) really matter to establish a model that accurately predicts at or exceed 90% graduation rate
- Model and predict individual performance of school district
- Compare prediction accuracy of different algorithms

- (Stretch target) conduct a further comparative analysis if the feature is suitable for classification, that was reusable/useful for regression.

Analyzing the results may lead to the ideas of increasing graduation rates. In doing so, we can act on before problem becomes serious, and can propose workable solutions feasible to bring our goal closer to 90%

Metrics

Evaluation Metrics

For **Classification metrics**, we calculated average/mean, standard deviation of F-score (Precision, Recall) used for finding the correctness and accuracy of the model. The best value at 1 and worse score at 0

Table 2. Possible results of prediction

Confusion Matrix	Predicted as True	Predicted As False
Actually True	True Positive (TP)	False Negative (TN)
Actually False	False Positive (FP)	True Negative (TN)

Equations

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

For **Regression metrics**, we calculated Root Mean Squared Error (RMSE) used measure the difference between the predicted by a model and actual value observed, how accurately the model predicts the response.

Square root of the average of squared errors

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - p)^2} \quad where\, p_i\, is\, predicted\, value, p\, is\, modeled\, value$$

Lower values of RMSE indicate better fit.

II.Analysis

Data Exploration

We used the 2011-2012 high school graduates and 2010 census information per school district, state, and county level for our experiment.

1. Data Collection: the *GRADUATION_WITH_CENSUS.csv* was manually pre-downloaded from Data for Diplomas_Merged Data.zip located at https://challenges.s3.amazonaws.com/data_for_diplomas/Data for Diplomas_Merged Data.zip, and extracted CSV file and its definition ALL_DATA_SCHEMA_M.pdf
    - The dataset contain 2012 Four-year regulatory Adjusted Cohort Graduation Rates (ACGR) joined with the maximum overlapping 2010 Census data
    - The analysis was focus on the year of 2011/2012
2. Exploratory Analysis: our first data exploration found the money fields such as "Med_HHD_Inc_ACS_08_12", "Med_HHD_Inc_ACSMOE_08_12", "Aggregate_HH_INC_ACS_08_12", "Aggregate_HH_INC_ACSMOE_08_12", "Med_House_value_ACS_08_12", "Med_House_value_ACSMOE_08_12", "Aggr_House_Value_ACS_08_12", "Aggr_House_Value_ACSMOE_08_12" prefixed with a dollar sign ($) and embedded with commas (,). Cleaning step included removing the dollar sign and commas, and store the tidy dataset to a CSV file.

Inputs

The GRADUATION_WITH_CENSUS data contained various counts and percentages for each school district's population, ethnicity, gender, age, geographic location, and family educational background, and household etc. THe referenced fields were defined in ALL_DATA_SCHEMA_M.pdf

Because we had 576, very large number of features , we pre-processed and performed feature selections to extract those features that were really important rather than fitting a complete feature set of predictive models.

Actual inputs were the result of feature selections after performing various feature selection algorithms for classification and regression, and obtained less features that were very important for fitting predictive models

obtained less features that were very important for fitting predictive models.

Output Variable

The output variable represent the high school district performance on graduation

1. **Classification** target variable is "Success_Pass_90" indicator derived from "ALL_RATE_1112"
   - integer, value 1, PASSED successful if ALL_RATE_1112 is equal to or greater than 90.0
   - integer, value 0, NOT PASSED successful if ALL_RATE 1112 is less than 90.0
2. **Regression** target variable is "ALL_RATE_1112'
   - float number, in range of 0 to 100

Exploratory Statistics

We had 9,907 observations, and 576 features each observation in 2011-2012 graduation cohort. The input data contained the count and percentage for population, ethnicity, gender, age, geography, and family educational background, household by each school district.

- By Race total students in the 2011-2012 graduation cohort

```
  ALL_COHORT_1112             3,307,623   Total number of students in the 2011-2012 graduat
ion cohort
  MAM_COHORT_1112                30,965   Native Americans students
  MAS_COHORT_1112               168,589   Asian/Pacific Islander students
  MBL_COHORT_1112               553,003   Black/Africa American students
  MHI_COHORT_1112               658,507   Hispanic students
  MTR_COHORT_1112                46,676   Two/Three races students
  MWH_COHORT_1112             1,835,800   White students
  CWD_COHORT_1112               393,902   Children with disabilities
  ECD_COHORT_1112             1,431,050   Economically disadvantaged
```

- By Geography & Population total count

```
  Tot_Population_CEN_2010     41,000,304  U.S. resident population in 2010 Census
  RURAL_POP_CEN_2010          19,256,868  Population living outside of an Urbanized Area or
Urban Cluster
  URBANIZED_AREA_POP_CEN_2010 13,311,089  Population living in a densely settled area conta
ining 50,000 or more people
  URBAN_CLUSTER_POP_CEN_2010   8,432,347  Population living in a densely settled area conta
ining 2,500 to 49,999 people
```

- By Gender total count

```
  Males_CEN_2010              20,353,022  Males
  Females_CEN_2010            20,647,282  Females
```

- By Age total count

```
  Pop_under_5_CEN_2010         2,571,831  Persons less than age 5 in 2010 census
  Pop_5_17_CEN_2010            7,327,965  Persons aged 5 to 17 in 2010 census
  Pop_18_24_CEN_2010           3,608,431  Persons aged 18 to 24 in 2010 census
  Pop_25_44_CEN_2010           9,898,820  Persons aged 25 to 44 in 2010 census
  Pop_45_64_CEN_2010          11,477,346  Persons aged 45 to 64 in 2010 census
  Pop_65plus_CEN_2010          6,115,911  Persons aged 65 and over in 2010 census
```

- By Family Background total count

```
  Pov_Univ_ACS_08_12          39,675,431  People under 15 years old
  Prs_Blw_Pov_Lev_ACS_08_12    5,457,898  People classified as below the poverty level
  Civ_labor_16plus_ACS_08_12  20,216,951  Civilians aged 16 years and over and in the labor
force
  Civ_emp_16plus_ACS_08_12    18,543,130  Civilians aged 16 years and over and employed
  Civ_unemp_16plus_ACS_08_12   1,673,821  Civilians aged 16 years and over and unemployed
  Civ_labor_16_24_ACS_08_12    2,833,010  Civilians aged 16 to 24 in the labor force in
```

- By Family Education total

```
  Not_HS_Grad_ACS_08_12        3,730,293  People 25 years old and over who are not high sch
ool graduates
  College_ACS_08_12            6,266,894  People 25 years old and over with college degree
or higher
```

- By English Language Speaks (Households where NO One age 14 & over speaks English only/very well) Total for

```
  ENG_VW_ACS_08_12               337,275  Households where NO One age 14 & over speaks Engl
ish only/very well
```

```
  ENG_VW_SPAN_ACS_08_12          225,915  Spanish or Spanish Creole language
  ENG_VW_INDO_EURO_ACS_08_12      55,547  Indo-European language
  ENG_VW_API_ACS_08_12            43,778  Asian and Pacific Island language
  ENG_VW_OTHER_ACS_08_12          12,035  Language other than English, Spanish, Indo-Euro,
or API
```

- By Income Total

```
  PUB_ASST_INC_ACS_08_12               380,658  Households that receive public assistance
income
  Med_HHD_Inc_ACS_08_12            515,609,190  Median ACS household income
  Aggregate_HH_INC_ACS_08_12 1,035,306,469,500  Sum of all incomes in the household
```

- Other Total

```
  Born_US_ACS_08_12           38,404,354  People were citizen of US at birth
  Born_foreign_ACS_08_12       2,492,610  People were not citizen of US at birth
  US_Cit_Nat_ACS_08_12         1,062,900  People were citizen of US through naturalization
  NON_US_Cit_ACS_08_12         1,429,710  People were not citizen of US
  MrdCple_Fmly_HHD_CEN_2010    8,326,407  Married-couple family households
  Not_MrdCple_HHD_CEN_2010     7,361,715  Households with no Married Couple presented
  Female_No_HB_CEN_2010        1,728,876  Households with a female householder, no husband

  NonFamily_HHD_CEN_2010       4,890,585  Households lives alone or with nonrelatives only
```

> Reference: API implementation: jtmoogle.hsgraduation.load_gradcensus

Classification

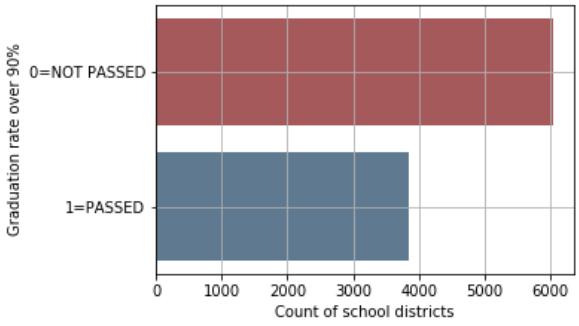
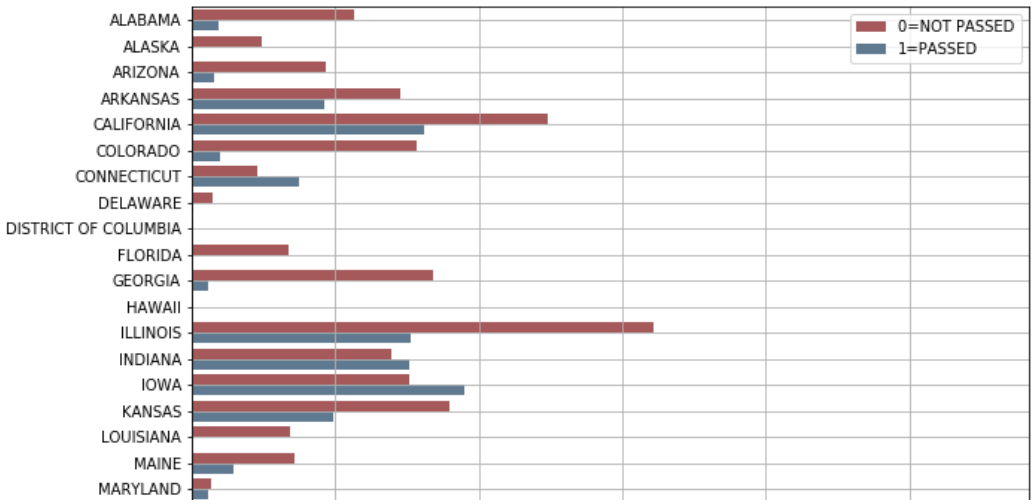


Figure 1. Graduation Census by Goal.

- 3,857 (39%) school districts have (1=PASSED) met the goal of 90% graduation rate
- 6,050 (61%) school districts has NOT met (0=NOT PASSED) the goal yet.



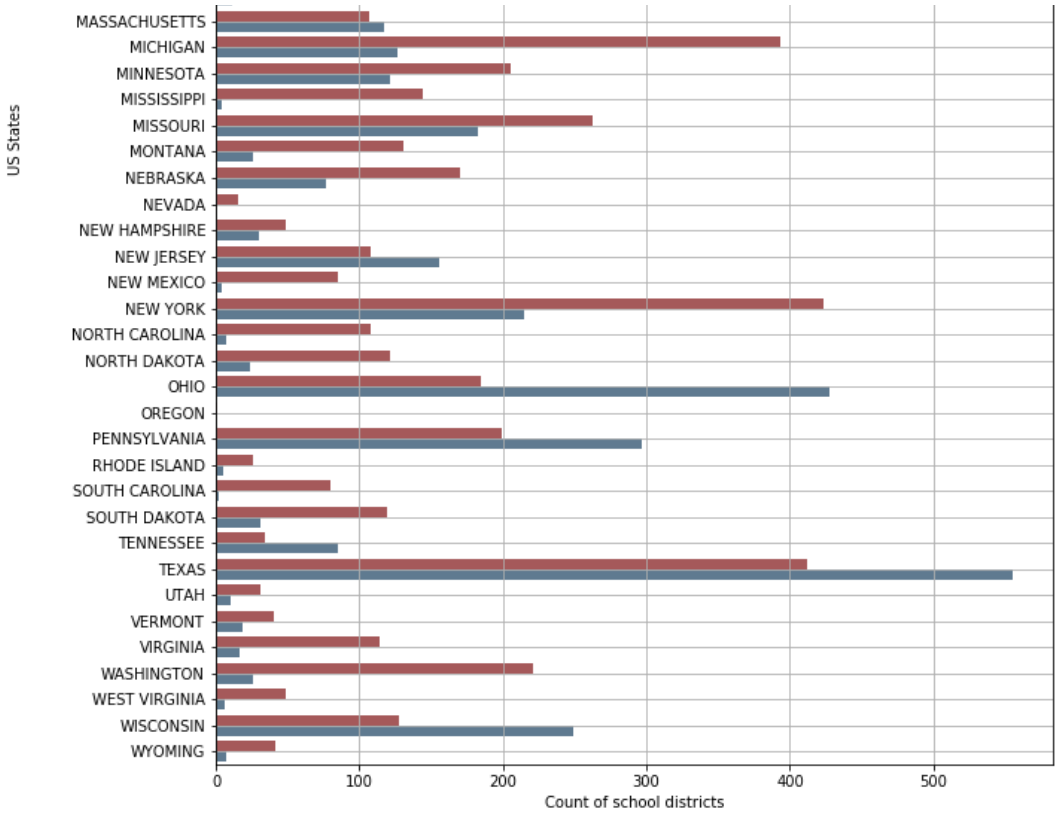


Figure 2. Graduation Census by State.

- 48 states, 320 counties, 3,307,632 high school students. (1=PASSED) The school districts have met the goal of 90% graduation rate; (0=NOT PASSED) school districts has NOT met yet.

> Reference: API implementation: jtmoogle.hsgraduation.plot_cls_gradcensus

Regression

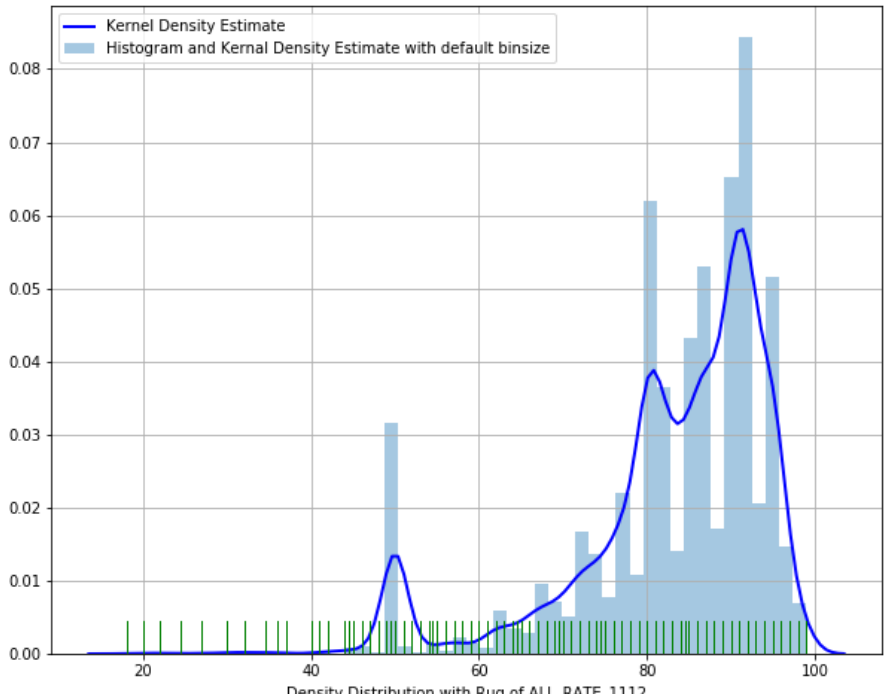


Figure 3. Visualize Regression distribution of ALL_RATE_1112.

- The Kernel Density Estimate/KDE Density plot showed all school districts on top of the corresponding histogram
- The rug on the x-axis showed actual data points. Observed overcrowded around 50%, 83-85% graduation rates
- y-axis represented the probability density per unit on the x-axis.

> Reference: API implementation: jtmoogle.hsgraduation.plot_rgs_gradcensus

Algorithms and Techniques

The two processes trained a set of predictive models respectively for classification and regression. The best candidate selection of supervised learning algorithms experimented. These models were evaluated using different performance metrics for classification and regression.

Table 3 Listed predictive models, feature selection methods, training and test sample counts by target variable

Predictive Model Target Variable	**Feature Selection Reduced to less variables**	**Samples**	**Evaluated Models**	**Metric Result Comparison**
Classification model Target:Success_Pass_90	**RFE model with Logistic Regression** Reduce to **20**	Training **6,920 / 70%** Testing **2,966 / 30%**	**Gaussian Naive Bayes (NB) Logistic Regression Random Forest(RF) K-Nearest Neighbors (KNN) Decision Tree(DT) Ridge Classifier(Ridge) Perceptron Gradient Boosting Classifier(GB)**	**F score**
Regression model Target:ALL_RATE_1112	**Stepwise Feature Selection** Reduce to **30**	Training **6,828 / 70** Testing **2,927 / 30%**	**Linear Regression(Linear) Random Forest(RF) k-Nearest Neighbors(KNN) Decision Tree(DT) Ridge Support Vector Regression(SVR)**	**Root Mean Square Error(RMSE)**

Classification

1. Feature Selection: selected good features through the feature selection, we used various ML algorithms to rank the feature importance, scaled the result in range of zero and one, and then calculated the mean/average of results.
   - Firstly used the Recursive Feature Elimination/RFE with the estimator of Random Forest Classifier to fit the regression models. This process iterated reserving the best performance features and removing the worse features until all features in database were exhausted.
     - Default to selected the 20th best ranked as input features for classification model
     - Used Random Forest Classifier to get the best feature importance and the best accuracy score for classification features
2. In addition, compared the result like coef or feature importance from the algorithms of Logistic Regression/Logit, Ridge Classifier, Decision Tree/DT, Random Forest Classifier, and average of mean of all ranking scores. We re-selected features if the mean of result was greater than 0.2 in range of 0 and 1.
3. Create Samples: the dataset was split into 70% training, 30% testing using the target variable. We used the training dataset to learn and evaluate predictive models, and used the testing dataset to measure the performance metric.
   - Training set was 6920 (70.00%) Testing 2966 (30.00%) in total 9886.
4. Predictive Models: used the following models to train/learn data, reduced training and evaluation time, reduced the complexity of predictive models, improve the accuracy of models and reduce overfitting.
   - Gaussian Naive Bayes (NB)
   - Logistic Regression
   - Random Forest (RF)
   - K-Nearest Neighbors (KNN)
   - Decision Tree (DT)
   - Ridge Classifier (Ridge)
   - Perceptron
   - Gradient Boosting Classifier (GB)
5. F score was used for metric result comparison

Regression

1. Feature Selection selected good features performed a stepwise feature selection based on p-value from statsmodels.api.OLS [9]. Iterate calculating, evaluating
   - if p-value was greater than threshold out (0.05), the feature which reduced the performance the least
   - if p-value was less than threshold in (0.01),the feature which improved performance the most

- Forward selected the best single feature, which improved performance the most.
- Backward elimination, a set of remaining features, repeatedly delete the feature that reduces performance the least
- Used Random Forest Regressor to get the best feature importance and the best R2 score for Regression features

2. In addition, compared the result like coef or feature importance from the algorithms of Linear Regression, Ridge Regressor, Decision Tree/DT, Random Forest/RF Regressor, and average of mean of all ranking scores. We re-selected features if the mean of result was greater than 0.2 in range of 0 and 1.
3. Create Samples: the dataset was split into 70% training, 30% testing using the target variable. We used the training dataset to learn and evaluate predictive models, and used the testing dataset to measure the performance metric.
   - Training set was 6,828 (69.99%) testing 2,927 (30.01%) in total 9,755
4. Predictive Models: used the following models to train/learn data faster, reduced training and evaluation time, reduced the complexity of predictive models, improve the accuracy of models and reduce overfitting.
   - Linear Regression (Linear)
   - Random Forest (RF)
   - k-Nearest Neighbors (KNN)
   - Decision Tree (DT)
   - Ridge
   - Support Vector Regression (SVR)
5. Root Mean Square Error/RMSE was used for metric result comparison

Benchmark

We used the naive technique to make a forecast and calculate the baseline performance. After we explored more options for naive prediction during capstone report, we decided to adopt different option. The model results were compared with the naive predictor benchmarks. We selected the best performance and the less training time for the final model.

1. Use Gaussian Naive Bayes (NB) for classification
2. Used Linear Regression for regression

III. Methodology

Data Processing

Pre-process Data

We applied the following logics on the 576 features, and reduced to 152 features.

1. Removed non-relevant variables toward target variable. Dropped the columns whose feature names contained the partial/full text of *MOE_, FRMS, Mail, Percentage, County, State, Tract, District, GIDTR, Tract, Flag, Response, Delete, Vacant, BILQ, Diff, Leave, Plumb*.
2. Included columns whose feature names contained partial/full text of *Inc, INC, COHORT, pct, avg, House, AREA, ALL_, Success*
3. Selected columns whose data types were float64 or int32/int64.
4. Dropped rows if column has NaN/nullable values
5. Extracted target data based on the target column name
6. Removed non-relevant variables such as the target column: (1) Success_Pass_90 for classification (2) ALL_RATE_1112 for regression
7. Identified missing value and imputed NaN with zero by filling with zero value
8. We did not do further data transformation because AT&T group has consolidated and applied necessary transformation for Hackathon in advanced.

> Reference: API implementation: jtmoogle.hsgraduation.preprocdata, preproc_cls_data, preproc_rgs_data

Implementation of APIs

We implemented various supervised learning algorithms respectively two process for classification and regression. The following APIs invoked in this report were implemented in Python, and source files were on GitHub jtmoogle

- helper.py contains classes as static methods
  - MyLogger - logging to a file or console output for purpose of debug, info, error
  - MyHelper - common functions to load dataset, print out statistic summary, save to result to a file
- hsgraduation.py contains functions and methods
  1. load_gradcensus - load jtmoogle/data/GRADUATION_WITH_CENSUS.csv
  2. Classification
     (1) plot_cls_gradcensus - illustrate classification graduation census data
     (2) preproc_cls_data - prepare classification data cleaning
     (3) cls_feature_sel - classification feature selection
     (4) compare_cls_featranking - compared classification feature ranking (5) cls_stats - classification statistic summary
     (6) cls_acc_featimportance - mean accuracy score and feature importance result for classification feature (7) create_cls_sample - create sample for Training and Testing datasets
     (8) cls_acc_featimportance - calculated F score, accuracy, feature importance
     (9) cls_visual_benchmark - benchmark result and visualization
     (10) handy functions: cls_pca - PCA result for classification

3. Regression
   (1) plot_rgs_gradcensus - illustrate regression graduation census data
   (2) preproc_rgs_data - prepare Regression data cleaning
   (3) rgs_feature_sel - regression feature selection
   (4) compare_rgs_featranking - compared regression feature ranking (5) rgs_stats - regression statistic summary
   (6) create_rgs_sample - create sample for Training and Testing datasets
   (7) rgs_r2_featimportanct - R2 score and feature importance for regression features
   (8) rgs_visual_benchmark - benchmark result and visualization

- runme.py which was capable to reproducible statistics summary results, and visualization seen in this report
- data/GRADUATION_WITH_CENSUS.csv (raw data) GRADUATION_WITH_CENSUS.csv.ds.csv (cleaner data, output data file via load_gradcensus method) ALL_DATA_SCHEMA_M.pdf (field definition)

Implementation for Classification

Feature Selection: Select Good Features

We implemented the wrapper method of *RFE/Recursive Feature Elimination using sklearn.feature_selection API. The task was repeated as follows

1. Constructed the Logistic Regression model estimator
2. Chose the best performance features
3. Removed the worse ones.

The process iterated until all features in dataset were exhausted. This was eliminated from 152 to 20 features, and value of 1 was ranked the 20th best features which were selected as input variables for classification model.

In addition, we used various models to evaluate these 20 features, obtained coefficient or feature importance values per feature, calculated the mean/average of coefficient or feature importance, and then compared side by side. At last, we selected features with better averages greater than 0.2. This reduced to 8 features with better important correlation

Table 4 Compared result of various feature selection methods. Chose features with better averages greater than 0.2

Features	DT	RF	Ridge	logit	Mean	Feature Category	Feature Description
MWH_COHORT_1112	1	1	0	0.02	**0.5**	HS Graduation	Number of White students in the graduation cohort
ECD_COHORT_1112	0.93	0.41	0.03	0.05	**0.36**	HS Graduation	Number of economically disadvantaged students in the graduation cohort
Med_HHD_Inc_ACS_08_12	0.23	0.18	0	1	**0.35**	Household	Median ACS household income
pct_Female_No_HB_CEN_2010	0	0.08	1	0	**0.27**	Household	Households with a female householder, no husband present
pct_College_ACS_08_12	0.03	0.09	0.91	0	**0.26**	Population	Persons 25 years and over with college degree or higher
pct_Not_MrdCple_HHD_CEN_2010	0.08	0.09	0.86	0.01	**0.26**	Household	Households with no Married Couple present
pct_Tot_Occp_Units_ACS_08_12	0.05	0.16	0.83	0.01	**0.26**	Housing Unit	Total Occupied Housing Units which has regular occupants in ACS
pct_Civ_emp_16p_ACS_08_12	0	0.07	0.75	0.01	**0.21**	Population	Civilians aged 16 years and over and employed
pct_NH_White_alone_CEN_2010	0.02	0.03	0.69	0	0.18	Population	no Hispanic origin and their only race as "White" as Irish, German, Italian, Lebanese, Arab, Moroccan, or Caucasian
pct_Prs_Blw_Pov_Lev_ACS_08_12	0.01	0.16	0.43	0	0.15	Population	Populateion classified as below the poverty level
pct_Hispanic_CEN_2010	0	0	0.45	0	0.11	Population	Mexican, Puerto Rican, Cuban, or another Hispanic, Latino, or Spanish origin
Aggregate_HH_INC_ACS_08_12	0	0.35	0	0	0.09	Household	Sum of all incomes in the Household
CWD_COHORT_1112	0	0.23	0.06	0.01	0.08	HS Graduation	Number of children with disabilities in the graduation cohort
MBL_COHORT_1112	0.06	0.19	0.02	0.02	0.07	HS Graduation	Number of Black students in the graduation cohort
MHI_COHORT_1112	0.12	0.07	0.04	0.02	0.06	HS Graduation	Number of Hispanic students in the graduation cohort
MAM_COHORT_1112	0.01	0.01	0.12	0	0.04	HS Graduation	Native American students in the graduation cohort
LEP_COHORT_1112	0.01	0.09	0.01	0.01	0.03	HS Graduation	Number of limited English proficient students in the graduation cohort
MAS_COHORT_1112	0.04	0.03	0.04	0	0.03	HS Graduation	Number of Asian/Pacific Islander students in the graduation cohort
pct_Not_HS_Grad_ACS_08_12	0.03	0.03	0.02	0	0.02		People 25 years old and over who are not high school graduates

Features	DT	RF	Ridge	logit	Mean	Feature Category	Feature Description
Aggr_House_Value_ACS_08_12	0	0.02	0	0	0	Housing Unit	Aggregate housing unit value

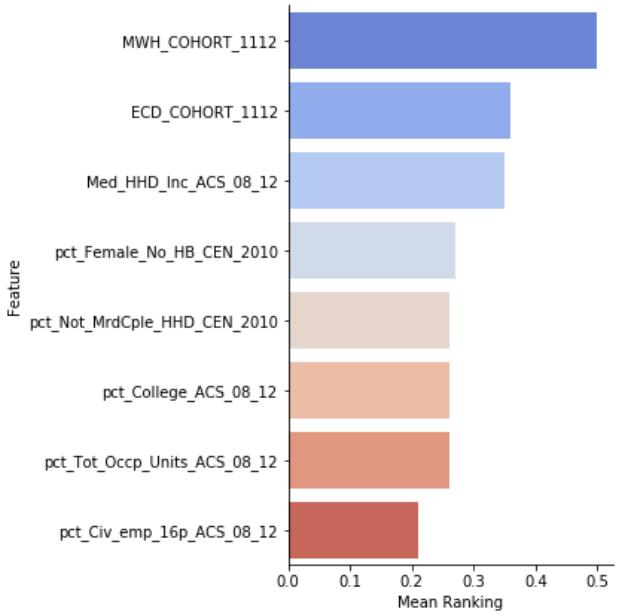


Figure 4: Classification Features and Mean Ranking Comparison Matrix. MWH_COHORT_1112 had the highest ranking mean value with 0.50; ECD_COHORT_1112 0.36; Med_HHD_Inc_ACS_08_12 0.35

> Reference: API implementation: jtmoogle.hsgraduation.cls_feature_sel, compare_cls_featranking

The Ordinary Least Square/OLS Result for classification in table 5 listed overall model fit 9,894 observation. Degree of freedom 9,874, and fitting model with 8 predictors

- If P>|t| is less than the confidence level, often 0.05, it indicates that there is a statistically significant relationship, correlation between the dependent (target) variable and the independent (predictor) variables.
- R-squared 0.467 overall measure of the strength of association in the dependent (target) variable 'Success_Pass_90' can be explianed by the independent variables (20 predictors)

Table 5. Classification: Summary of OLS Result for the dependent variable at the top (Success_Pass_90) with the 8 selected features as predictor variables below in Parameter Estimates.

OLS Regression Results

Dep. Variable:	Success_Pass_90	R-squared:	0.463
Model:	OLS	Adj. R-squared:	0.463
Method:	Least Squares	F-statistic:	1065.
Date:	Sat, 12 May 2018	Prob (F-statistic):	0.00
Time:	21:15:56	Log-Likelihood:	-6291.3
No. Observations:	9886	AIC:	1.260e+04
Df Residuals:	9878	BIC:	1.266e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
MWH_COHORT_1112	-3.173e-05	1.56e-05	-2.029	0.043	-6.24e-05	-1.07e-06
ECD_COHORT_1112	-5.833e-05	8.34e-06	-6.991	0.000	-7.47e-05	-4.2e-05
Med_HHD_Inc_ACS_08_12	1.685e-06	3.99e-07	4.227	0.000	9.03e-07	2.47e-06
pct_Female_No_HB_CEN_2010	-0.0035	0.001	-3.053	0.002	-0.006	-0.001
pct_Not_MrdCple_HHD_CEN_2010	-0.0046	0.001	-7.673	0.000	-0.006	-0.003
pct_College_ACS_08_12	0.0043	0.001	7.979	0.000	0.003	0.005
pct_Tot_Occp_Units_ACS_08_12	0.0058	0.000	14.098	0.000	0.005	0.007
pct_Civ_emp_16p_ACS_08_12	-0.0002	0.000	-0.426	0.670	-0.001	0.001

Omnibus:	276.301	Durbin-Watson:	1.696
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1093.084
Skew:	0.426	Prob(JB):	4.36e-238

When the following increase, Success_Pass_90 would (list first 3)

- pct_Tot_Occp_Units_ACS_08_12: increase 0.0058
- pct_College_ACS_08_12: increase 0.0043

- pct_College_ACS_08_12: increase 0.0043
- Med_HHD_Inc_ACS_08_12: increase 1.685e-06

When the following increase, Success_Pass_90 would (list first 3)

- pct_Not_MrdCple_HHD_CEN_2010: decrease 0.0046
- pct_Female_No_HB_CEN_2011: decrease 0.0035
- pct_Civ_emp_16p_ACS_08_12: decrease 0.0002

> Reference: API implementation: jtmoogle.hsgraduation.cls_stats, cls_acc_featimportance

Create Samples

We used the training dataset to learn and evaluate predictive models. The sklearn.model_selection.train_test_split API was used the target variable *Success_Pass_90* to split the dataset into 70% /6,920 observation for training, 30% / 2,966 observation for testing.

> Reference: API implementation: jtmoogle.hsgraduation.create_cls_sample

Detect Outliers

Using John Tukey's Method [35] to identfy outliers data points by calculating as 1.5 times the interquartile range (IQR). A data point with a feature that was beyond an outlier step outside of the IQR for that feature is considered abnormal. For each feature, steps were

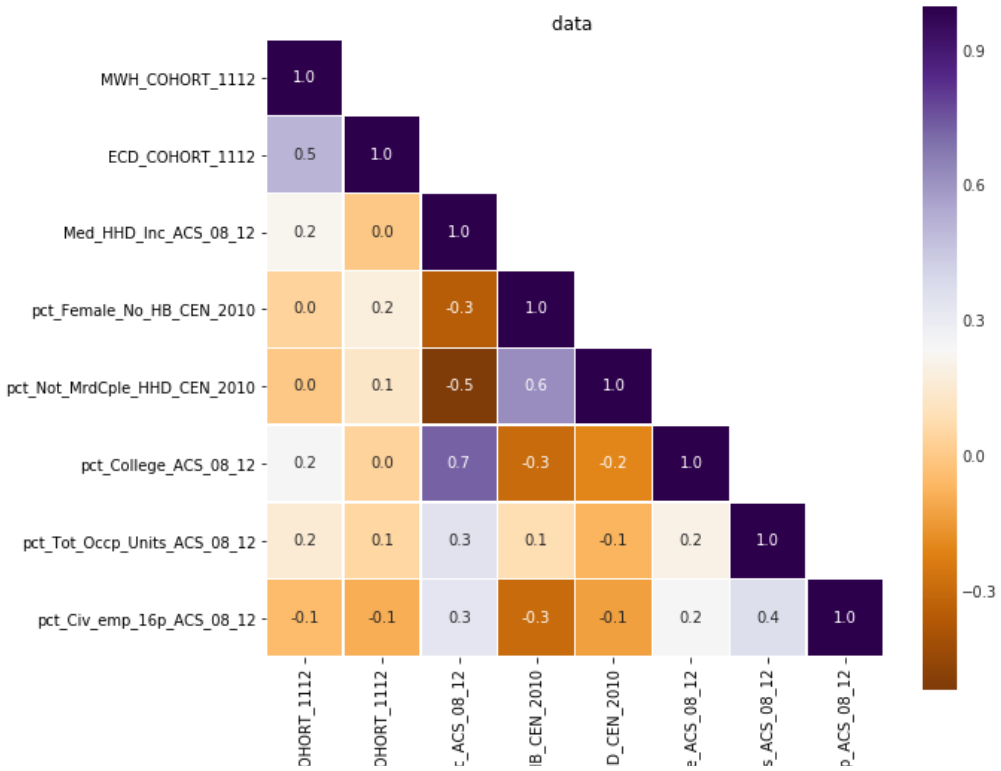
1. To find the data points with extreme high or low values
2. Calculated Q1 (25th percentile of the data) for the given feature
3. Calculated Q3 (75th percentile of the data) for the given feature
4. Computed Q3 - Q1 for IQR
5. Used the IQR to calculate an outlier step (1.5 times the IQR). Anything outside this range is an outlier

$$[Q_1 - k(Q_3 - Q_1), Q_3 + k(Q_3 - Q_1)] \quad where\, k\, is\, 1.5\, indicating\, an\, ' outlier'$$

All duplicated outlier data point would be removed from the sample, and stored data as 'good_data'

Feature Scaling

The natural logarithm was applied to scale the good_data, and we used heatmap to get further insight exhibited the feature's Correlation Matrix in Figure 5.



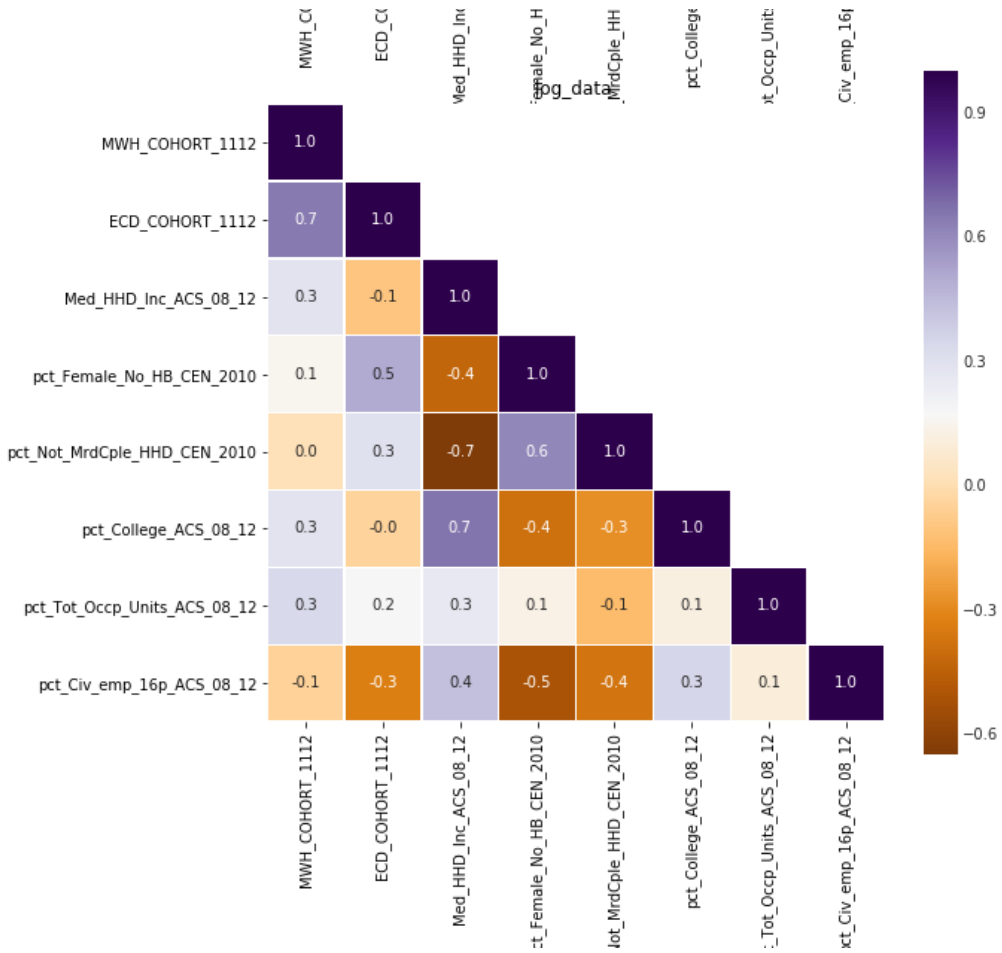


Figure 5. Visualized comparing Classification data and log_data in Correlation Matrix.

Table 6 Feature Scaling: Log Transform change if any value above 0.5

Features in correlation above 0.5	Value Change	Correlation Change
ECD_COHORT_1112 & MWH_COHORT_1112	0.5 -> 0.7	Stronger correlation
pct_Not_MrdCple_HHD_CEN_2010 & pct_Female_No_HB_CEN_2010	0.6 -> 0.6	no change
pct_College_ACS_08_12 & Med_HHD_Inc_ACS_08_12	0.7 -> 0.7	no change
pct_Female_No_HB_CEN_2010 & ECD_COHORT_1112	0.2 -> 0.5	Stronger

Fit Predictive Model for Supervised Learning Algorithms

We used the training dataset to learn and evaluate predictive models. The process trained a set of predictive models respectively, and experimented the sklearn.ensemble, sklearn.neighbors and sklearn.linear_model APIs as follows:

- Gaussian Naive Bayes (NB) used default value
- Logistic Regression (Logit) used default value
- Random Forest Classifier (RF) used 5 for max_depth, 10 for n_estimators
- K-Nearest Neighbors Classifier (KNN) used 3 for n_neighbors
- Decision Tree Classifier (DT) use 5 for max_depth
- Ridge Classifier (Ridge) used le-2 for tol, lsgr for solver
- Perceptron used 50 for n_iter, 0.1 for alpha, none for penalty
- Gradient Boosting Classifier(GB) used 100 for n_estimators, 4 for max_leaf_nodes, none for max_depth, 2 for random_state, 5 for min_samples_split

> Reference: API implementation: jtmoogle.hsgraduation.cls_acc_featimportance

Refinement

The Gaussian Naive Bayes (NB) was selected as Naive Prediction. We compared the results of performance metrics with the Naive Prediction. Time taken for training and testing were captured.

Implementation for Regression

Feature Selection: Select Good Features

We performed the *Stepwise feature selection based on p-value from statsmodels.api.OLS [9]* for regression model. The task was iterated calculating and evaluating the following logics

calculating and evaluating the following logics:

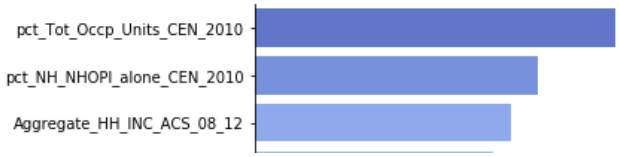
1. Repeated if p-value was greater than threshold-out value (default to 0.05), the feature had reduced performance to the least
2. Repeated if p-value was less than threshold-in value (default to 0.01),the feature had improved performance the most
3. The forward path selected the best single feature which improved performance the most.
4. The backward path eliminated a set of remaining features, repeatedly deleted the features that reduced performance the least

As result of the process completion, features were reduced from 152 to 30 features which were selected as input variables for regression model.

Furthermore, we used various models to evaluate these 30 features, obtained coefficient or feature importance values per feature, calculated the mean/average of coefficient or feature importance, and then compared side by side. At last, we selected features with better averages greater than 0.2. This reduced to 13 features with better important correlation.

Table 7 Compared result of various feature selection methods. Selected features with better averages greater than 0.2|

Feature	Corr	DT	Linear	RF	Ridge	Mean	Feature Category	Feature Description
pct_Tot_Occp_Units_CEN_2010	0.64	1	0.07	1	0.08	**0.56**	Housing Unit	Total Occupied Housing Units which has regular occupants in census
pct_NH_NHOPI_alone_CEN_2010	0	0.12	1	0.09	1	**0.44**	Population	Non-Hispanic Native Hawaiian and Other Pacific Islander only
Aggregate_HH_INC_ACS_08_12	1	0.54	0	0.44	0	**0.4**	Household	Sum of all incomes in the household
pct_NH_AIAN_alone_ACS_08_12	0.46	0.55	0.19	0.46	0.18	**0.37**	Population	Non-Hispanic American Indian and Alaska Native only
MBL_COHORT_1112	0.13	0.68	0	0.63	0	**0.29**	HS district	Number of Black students in the graduation cohort
pct_Pop_Under_5_CEN_2010	0.08	0.09	0.61	0.14	0.51	**0.29**	Population	Number of persons under age 5
pct_Prs_Blw_Pov_Lev_ACS_08_12	0.69	0.32	0.04	0.32	0.04	**0.28**	Population	Number of people classified as below the poverty level
pct_Age5p_WGerman_ACS_08_12	0.01	0	0.66	0	0.64	**0.26**	Population	People ages 5+ years who speak English less than "very well" and speak *another* West Germanic language at home
pct_PUB_ASST_INC_ACS_08_12	0.32	0.25	0.2	0.25	0.2	**0.24**	Household	households that receive public assistance income
Med_House_value_ACS_08_12	0.55	0.28	0	0.29	0	**0.22**	Housing Unit	Median of house value
pct_Age5p_German_ACS_08_12	0.01	0.05	0.48	0.06	0.48	**0.22**	Population	Persons 5+ years who speak English less than "very well" and speak German at home
pct_Females_CEN_2010	0.05	0.28	0.28	0.26	0.21	**0.22**	Population	Number of Females in total population
pct_NH_Blk_alone_CEN_2010	0.37	0.22	0.13	0.24	0.11	**0.21**	Population	Non-Hispanic Black/African American only
pct_Civ_emp_16p_ACS_08_12	0.1	0.33	0.05	0.36	0.06	0.18		
pct_Mobile_Homes_ACS_08_12	0.34	0.23	0.03	0.23	0.03	0.17		
pct_Pop_45_64_CEN_2010	0	0.27	0.17	0.2	0.18	0.16		
pct_Female_No_HB_ACS_08_12	0.19	0.21	0.08	0.22	0.06	0.15		
pct_HHD_PPL_Und_18_CEN_2010	0.06	0.16	0.17	0.16	0.12	0.13		
pct_NO_PH_SRVC_ACS_08_12	0.13	0.15	0.11	0.17	0.1	0.13		
pct_Pop_25_44_ACS_08_12	0.04	0.24	0.1	0.2	0.09	0.13		
pct_Rel_Under_6_CEN_2010	0.07	0.19	0.13	0.14	0.11	0.13		
pct_ENG_VW_INDOEURO_ACS_08_12	0	0.09	0.23	0.06	0.24	0.12		
pct_Hispanic_CEN_2010	0.03	0.21	0.05	0.25	0.04	0.12		
pct_NonFamily_HHD_CEN_2010	0.11	0.11	0.13	0.1	0.08	0.11		
pct_Census_UAA_CEN_2010	0	0.21	0.04	0.2	0.04	0.1		
pct_MLT_U10p_ACS_08_12	0.03	0.17	0.05	0.19	0.05	0.1		
pct_RURAL_POP_CEN_2010	0.38	0	0.02	0.1	0.02	0.1		
pct_Rel_Family_HHDS_CEN_2010	0.11	0.13	0.08	0.11	0.02	0.09		
URBANIZED_AREA_POP_CEN_2010	0.27	0.03	0	0.04	0	0.07		
pct_Inst_GQ_CEN_2010	0	0.06	0.09	0.1	0.07	0.06		



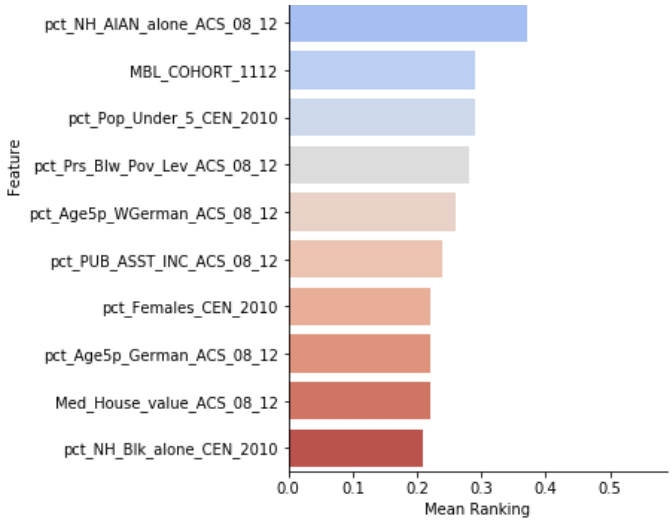


Figure 6: Regression Features and Mean Ranking Comparison Matrix. pct_Tot_Occp_Units_CEN_2010 had the highest ranking mean value with 0.56; pct_NH_NHOPI_alone_CEN_2010 0.44; Aggregate_HH_INC_ACS_08_12 0.4

> Reference: API implementation: jtmoogle.hsgraduation.rgs_feature_sel, compare_rgs_featranking

The OLS Result for regression model in table 8 listed overall model fit 9,755 observation. Degree of freedom/df 9,742, and fitting model with 13 predictors

- If P>|t| is less than the confidence level, often 0.05, it indicates that there is a statistically significant relationship, correlation between the dependent (target) variable and the independent (predictor) variables.
- R-squared 0.980 overall measure of the strength of association in the dependent (target) variable 'ALL_RATE_1112' can be explianed by the independent variables (34 predictors)

Table 8. Regression: Summary of Ordinary Least Squares/OLS Result for the dependent variable at the top (ALL_RATE_1112) with the 13 selected features as predictor variables below in Parameter Estimates.

OLS Regression Results

Dep. Variable:	ALL_RATE_1112	R-squared:	0.980
Model:	OLS	Adj. R-squared:	0.980
Method:	Least Squares	F-statistic:	3.647e+04
Date:	Sat, 12 May 2018	Prob (F-statistic):	0.00
Time:	21:19:10	Log-Likelihood:	-38004.
No. Observations:	9755	AIC:	7.603e+04
Df Residuals:	9742	BIC:	7.613e+04
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
pct_Tot_Occp_Units_CEN_2010	0.3504	0.011	31.446	0.000	0.329	0.372
pct_NH_NHOPI_alone_CEN_2010	-0.5520	0.495	-1.114	0.265	-1.523	0.419
Aggregate_HH_INC_ACS_08_12	1.664e-08	2.43e-09	6.843	0.000	1.19e-08	2.14e-08
pct_NH_AIAN_alone_ACS_08_12	-0.2153	0.017	-12.750	0.000	-0.248	-0.182
MBL_COHORT_1112	-0.0033	0.000	-8.491	0.000	-0.004	-0.003
pct_Pop_Under_5_CEN_2010	-0.3329	0.081	-4.130	0.000	-0.491	-0.175
pct_Prs_Blw_Pov_Lev_ACS_08_12	0.0165	0.016	1.019	0.308	-0.015	0.048
pct_Age5p_WGerman_ACS_08_12	0.9944	0.261	3.809	0.000	0.483	1.506
pct_PUB_ASST_INC_ACS_08_12	-0.2389	0.054	-4.409	0.000	-0.345	-0.133
pct_Females_CEN_2010	1.0530	0.020	53.583	0.000	1.015	1.092
pct_Age5p_German_ACS_08_12	-0.7203	0.265	-2.719	0.007	-1.240	-0.201
Med_House_value_ACS_08_12	8.88e-06	1.26e-06	7.039	0.000	6.41e-06	1.14e-05
pct_NH_Blk_alone_CEN_2010	-0.1202	0.010	-12.347	0.000	-0.139	-0.101

Omnibus:	1646.451	Durbin-Watson:	1.818
Prob(Omnibus):	0.000	Jarque-Bera (JB):	27211.480
Skew:	0.294	Prob(JB):	0.00
Kurtosis:	11.161	Cond. No.	5.18e+08

When the following increase, ALL_RATE_1112 would (list first 3)

- pct_Females_CEN_2010: increase 1.05
- pct_Age5p_WGerman_ACS_08_12: increase 0.99

- pct_Tot_Occp_Units_CEN_2010: increase 0.35

When the following increase, ALL_RATE_1112 would (list first 3)

- pct_Age5p_German_ACS_08_12: decrease 0.72
- pct_NH_NHOPI_alone_CEN_2010: decrease 0.55
- pct_Pop_Under_5_CEN_2010: decrease 0.33

> Reference: API implementation: jtmoogle.hsgraduation.rgs_stats, rgs_r2_featimportance

Create Samples

We used the training dataset to learn and evaluate predictive models. The sklearn.model_selection.train_test_split API was used the target variable *ALL_RATE_1112* to split the dataset into 70% /6,828 observation for training, 30% / 2,927 observation for testing.

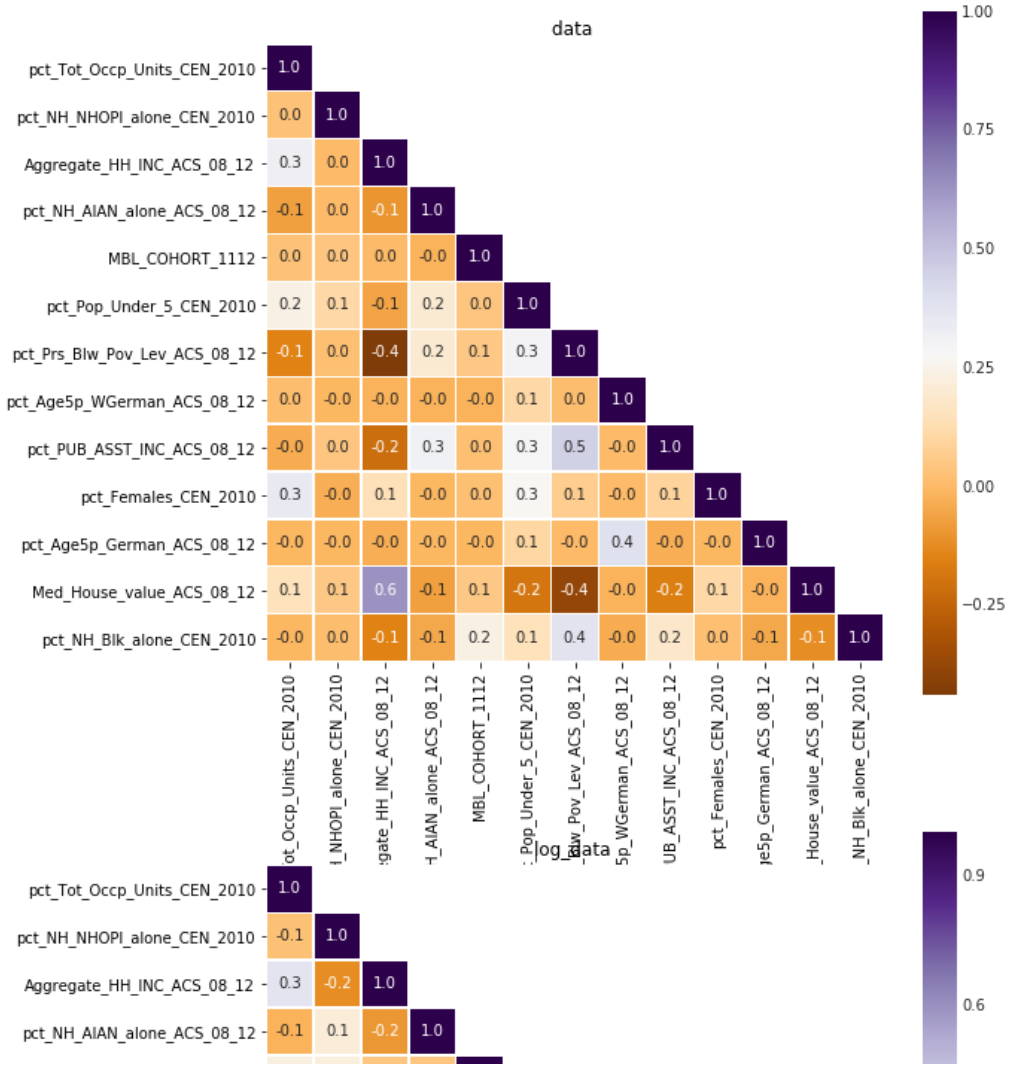
> Reference: API implementation: jtmoogle.hsgraduation.create_rgs_sample

Detect Outliers

Using John Tukey's Method [35] to identfy outliers data points by calculating as 1.5 times the interquartile range (IQR). A data point with a feature that was beyond an outlier step outside of the IQR for that feature is considered abnormal.

Feature Scaling

The natural logarithm was applied to scale the good_data, and we used heatmap to get further insight exhibited the feature's Correlation Matrix in Figure 7.



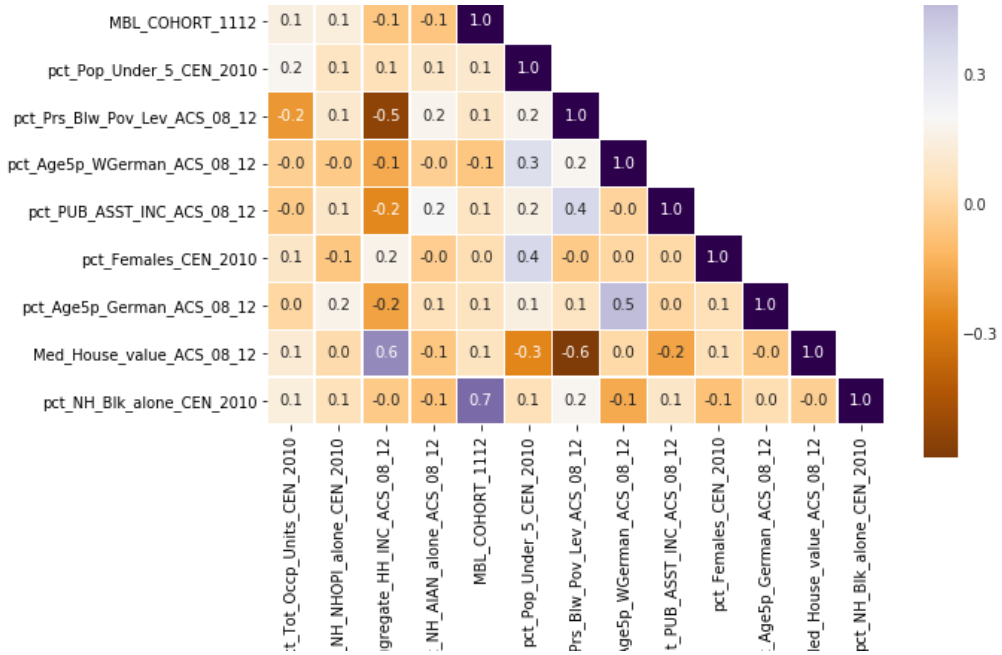


Figure 7. Visualize comparing Regression data and log_data in Correlation Matrix.

Table 9 Feature Scaling: Log Transform change if any value above 0.5

Features in correlation above 0.5	Value Change	Correlation Change
Med_House_value_ACS_0812 & Aggregate_HH_INC_ACS_08_12	0.6 -> 0.6	no change
pct_PUB_ASST_INC_ACS_08_12 & pct_Prs_Blw_Pov_Lev_ACS_08_12	0.5 -> 0.4	weaker correlation
pct_NH_Blk_alone_CEN_2010 & MBL_COHORT_1112	0.2 -> 0.7	Stronger
pct_Age5p_German_ACE_08_12 & pct_Age5p_WGerman_ACS_08_12	0.0 -> .5	Stronger

Fit Predictive Model for Supervised Learning Algorithms

We used the training dataset to learn and evaluate predictive models. The process trained a set of predictive models respectively, and experimented the sklearn.linear_model, sklearn.naive_bay, sklearn.neighbors, sklearn.svm APIs as follows:

- Linear Regression (Linear) used default value
- Random Forest Regressor (RF) used 30 for max_depth, 2 for random_state
- K-Nearest Neighbors Regressor (KNN) used 10 for n_neighbors=10
- Decision Tree Regressor (DT) used default value
- Ridge used 0.05 for alpha with 0.05
- Support Vector Regression (SVR) used rbf for Kernel

> Reference: API implementation: jtmoogle.hsgraduation.rgs_r2_featimportance

Refinement

The Linear Regression (Linear) was selected as Naive Prediction. We compared the results of performance metrics with the Naive Prediction. Time taken for training and testing were captured

IV. Results

Model Evaluation and Validation

We used Testing datase and the 10-fold cross-validation to test harness, compared the performance metrics exprimented various supervised learning algorithms. The API sklearn.model_selection.KFold used 10 for n_splits, 99 for random_state.

Benchmark for Classification

- The output of API sklearn.model_selection.KFold showed average of classification accuracy, precision, recall, **f1-score**
- The report of Confusion Matrix displayed the accuracy of a model with two or more classes. The table presents predictions on the x-axis and accuracy outcomes on the y-axis. Predictions for 0 that were actually 0 appear in the cell for prediction=0 and actual=0, whereas predictions for 0 that were actually 1 appear in the cell for prediction = 0 and actual=1.
- Classification Report of API sklearn.metric listed the precision, recall, f1-score and support for each class.
- If the model result had coef parameter, the output was in ascending order, the largest score the best score.

We used the Gaussian NB as Naive Prediction that predict all school district would have PASSED the goal of passing '900%' graduation rate. The accuracy was 0.55, and f1 score was 0.60.

Table 10 Classification Benchmark with 10 KFold cross validation, f1-score and Accurancy sorted by f1-score.

- Gaussian NB was Naive Prediction, accuracy 0.55, f1 score 0.60. But it had the longest time taken to learn.
- Gradient Boosting Classifier has the best performance, accuracy 0.75, f1 score 0.67

cls_names	train_time	test_time	accuracy_score	roc_auc	precision	avg_precision	recall	f1_score
Gradient Boosting Classifier (Best Performance)	1.00	0.00	**0.75**	0.83	0.69	0.76	0.64	**0.67**
Decision Tree Classifier	0.05	0.00	0.73	0.80	0.66	0.69	0.63	0.64
Random Forest Classifier	0.09	1.00	0.73	0.81	0.69	0.73	0.57	0.62
GaussianNB (Naive Prediction)	0.00	0.00	**0.55**	0.72	0.46	0.64	0.88	**0.60**
KNeighborsClassifier	0.09	1.00	0.64	0.65	0.54	0.50	0.53	0.53
LogisticRegression	0.09	0.00	0.70	0.73	0.69	0.66	0.43	0.53
RidgeClassifier	0.00	0.00	0.68	0.70	0.67	0.62	0.35	0.46
Perceptron	0.09	0.00	0.50	0.59	0.44	0.51	0.59	0.34

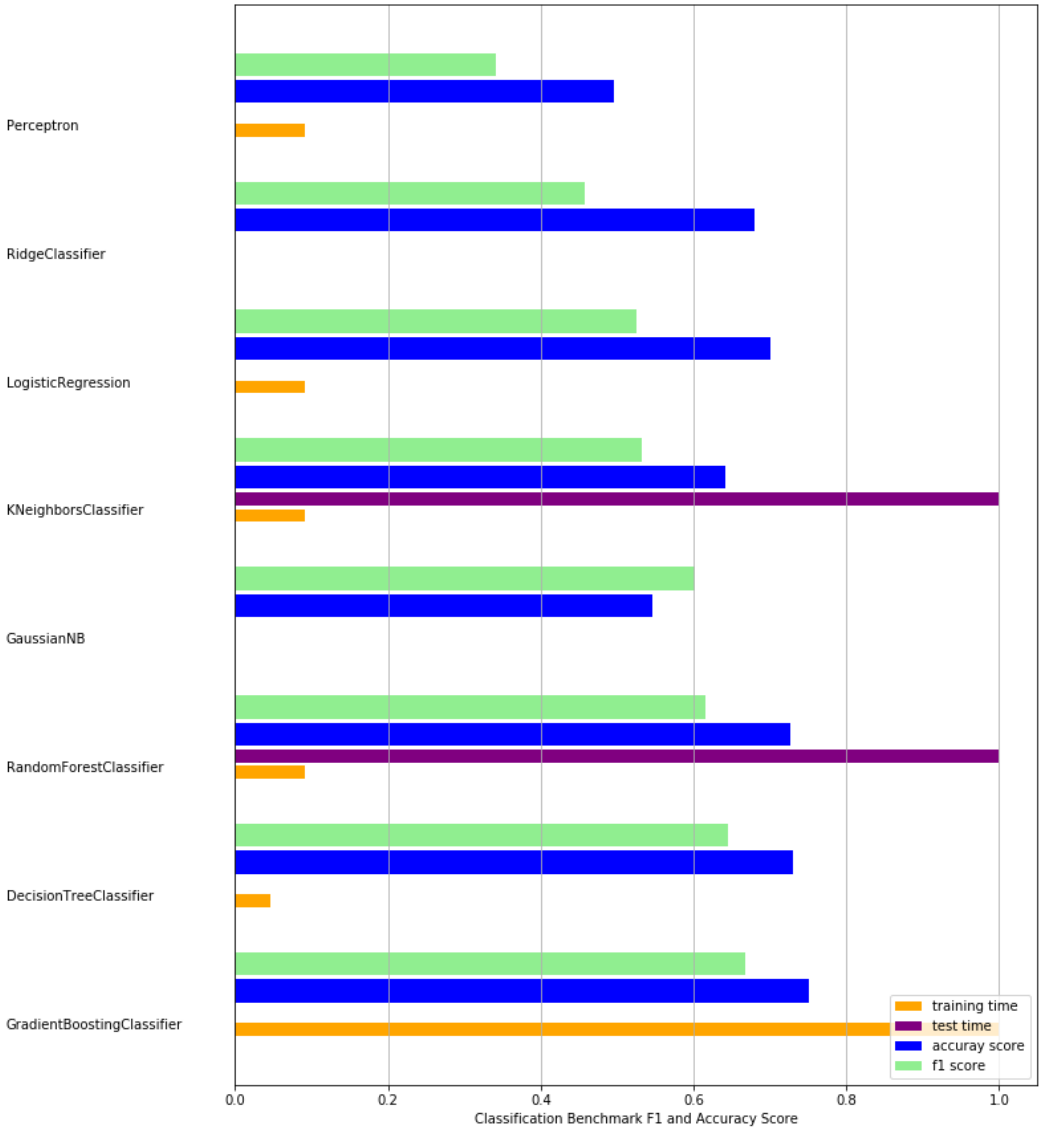


Figure 7 Visualize classification benchmark with F1-score and Accurancy sorted by f1-score.

- Gradient Boosting Classifer had best f2 score and accuracy score, but its training time taken was the longest
- GaussianNB has 4th beste of accuracy score and f1 score.

> Reference: API implementation: jtmoogle.hsgraduation.cls_visual_benchmark

Benchmark for Regression

- The output of API sklearn.model_selection.cross_val_score showed average of explained_variance_score, mean_absolute_error/MAE, mean_squared_error/MSE, root_mean_squared_error/RMSE, median_absolute_error/Median Abs Err, r2_score
- If the model result had calculated root sqaure of MSE, the output was in ascending order, the largest score the best score. If the success, the output would be 100, if school districts meet the goal of 90% graduation rate.

We used the Linear Regression as Naive Prediction that predict all school district would have '100%' graduation rate. The RMSE was 10.71 and r2 score was 0.19.

Table 11 Regression Benchmark with 10 KFold cross validation, RMSE and r2 score sorted by RMSE score.

- Linear Regression was Naive Prediction, RMSE 10.71, r2 score 0.19
- Random Forest Regressor has the best fit, RMSE 10.44, r2 score, 0.22

Regressor	Train Time	Test Time	Explained Variance	MAE	MSE	RMSE	Median Abs Err	R2 Score
Random Forest Regressor (Best Fit)	0.11	0.00	0.22	7.67	111.04	**10.54**	5.67	0.22
Ridge	0.00	0.00	0.19	7.72	114.66	10.71	5.97	0.19
Linear Regression (Naive Prediction)	0.00	0.00	0.19	7.72	114.71	**10.71**	5.96	0.19
SVR	1.00	1.00	0.01	8.50	151.31	12.30	5.81	0.07
K-Neighbors Regressor	0.01	0.01	0.17	9.53	166.07	12.89	7.07	0.17
Decision Tree Regressor	0.02	0.00	0.37	9.74	198.05	14.07	6.80	0.39

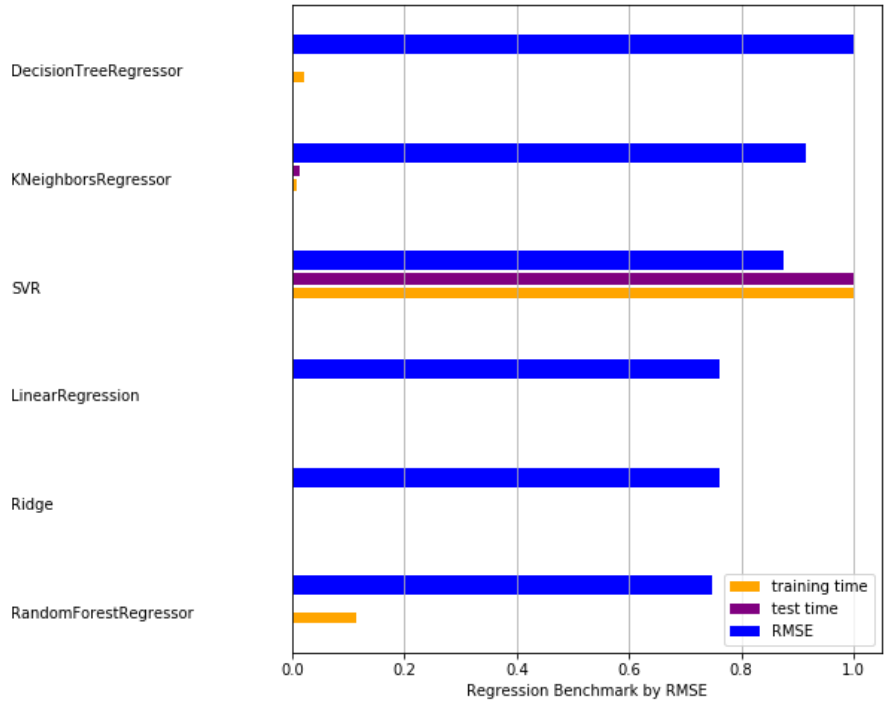


Figure 8 Visualize Regression benchmark with RMSE

- Random Forest Regressor has the best RMSE 10.44
- Linear Regression has 4th best of RMSE 10.71

> Reference: API implementation: jtmoogle.hsgraduation.rgs_visual_benchmark

Justification

During capstone development, we experienced the following which cause us to reinstall software platform, or OS drivers updates

- Changed to use new GPU card which was different manufactory from prior GPU, we need to uninstall and reinstall GPU drivers, as well

as tensor flow too.
- Anaconda major update, we need to re-install all dependent API packages, and tensor flow too

In summary, the application is useful for large number of data elements (i.e. 500 data elements), capable to distinguish and keep the higher feature importance within acceptable timeframes (i.e. 15 minutes for processing 9k observations) It allows to reduce to have 5% features (i.e. 25 features) which have higher feature importance.|

Conclusion

In this report, we demoed preliminary results for predicting High School Graduation from a large number of dataset including high school graduation and census records. We experimented various feature selection algorithms, detected outliers, applied feature scaling, and compared for the best performance. We found the Classification has better accuracy score than Regression result, and less features to fit and predict.

- Classification: 'Gradient Boosting Classifier' has the best performance, as it has the higher f1 score 0.67 than naive prediction Gaussian NB 0.60. The accuracy was 0.75 better than 0.55.
- Regression model: 'Random Forest Regressor' has the best fit because it has the less RMSE 10.54 value than naive prediction Linear Model 10.71. R2 0.22 better than 0.19

We explored the following insights from the experimented result of supervised learning algorithms, and understood the factors related to increase percentage achieving the goal of 90% graduation rate.

Regression
The 'Success_Pass_90' target would increase, suggest if population and households within the school districts or neighborhoods,

- More households increasing median household income if our job opportunities increase and more high paid jobs
- More family own the houses increasing percentage house owners live in the housing unit
- More family persons 25 years and over with college degree
- Decreasing not married couple in the household (reduce single parent percentage)
- Decreasing percentage of female no husband (reduce divorce percentage)
- Decreasing civilians aged 16 years and over and employed (reduce teenagers dropping out from high schools)

Classification
If graduation rate would increase percentage, suggest if population and households within the school districts or neighborhoods,

- Increasing females population
- Increasing people ages 5+ years who speak English less than "very well" and speak another West Germanic language at home
- More family own the houses increasing percentage house owners live in the housing unit
- Decreasing persons 5+ years who speak English less than "very well" and speak German at home
- Decreasing non-Hispanic Native Hawaiian and Other Pacific Islander only
- Decreasing number of persons under age 5

Deployment

All analysis performed in this study can be reproducible, and source files were on GitHub jtmoogle

- Anaconda 3 (64-bit)
- Program language: python 3 for data manipulation.
- tensorflow on CUDA GPU on Windows 10
- Tool: jupyter notebook for report and capstone analysis
- Machine learning libraries: sckikt-learn/sklearn
- Capstone Report in PDF format

Improvement

We would consider to make our jtmoogle APIs available to public, and benefit to broader audiences

- load_gradcensus - load high school graduation data (jtmoogle/data/GRADUATION_WITH_CENSUS.csv)

Using high school graduation data, we would expose

- For Classfication APIs
  - plot plot_cls_gradcensus - illustrate classification graduation census data
  - preproc_cls_data - prepare classification data cleaning
  - cls_feature_sel - classification feature selection
  - compare_cls_featranking - compared classification feature ranking
  - cls_stats - classification statistic summary
  - cls_acc_featimportance - mean accuracy score and feature importance result for classification feature (7) create_cls_sample - create sample for Training and Testing datasets
  - cls_acc_featimportance - calculated F score, accuracy, feature importance
  - cls_visual_benchmark - benchmark result and visualization
  - cls_pca - PCA result for classification
- For Regression APIs using high school g
  - plot_rgs_gradcensus - illustrate regression graduation census data
  - preproc_rgs_data - prepare Regression data cleaning

- rgs_feature_sel - regression feature selection
- compare_rgs_featranking - compared regression feature ranking
- rgs_stats - regression statistic summary
- create_rgs_sample - create sample for Training and Testing datasets
- rgs_r2_featimportanct - R2 score and feature importance for regression features
- rgs_visual_benchmark - benchmark result and visualization

In addition, we could generated web solution

- Allow school adminstratives, faculties and parents to see history the particular school district graduation rates
- Allow people to change the selected features to predict school graduation rates. This could help upcoming planning for school improvement.

Reflection

We spent over 3 months to implement python APIs for this capstone project.

Our thump up experience

- The most time taken was to explore options for feature selections for classification and regression. Experimented and troubleshoot several of supervising learning APIs and algorithms in python 3 because not of all of APIs are compatible to python 3.
- The flexibility we developed was to the configuration list, and to dynamically invoke reusable methods.
- The most handy was MyHelper.stats method skimpily generated common statistic output and plots.
- The automation we'd love was runme method passing scenario number to execute classification model or/and regression model.

Our thump down experience

- The most frustration was to re-install Anaconda and enforce updating API packages several times due to incompatible API version updates or OS driver GPU hardware change and hard drive boot disk failure etc.
- The unsolve task which we still could NOT make this work smoothly was to use jupyter converter to PDF file. The output of PDF file was poorly generated and not acceptable.

References

[1] The GradNation Campaign: Our Goal: Increase the nation's on-time high school graduation rate to 90% for the class of 2020

[2] 2015 Building a Grad Nation Report: Progress and Challenge in Ending the High School Dropout Epidemic
Number of Additional Graduates Needed to Reach a 90 Percent Graduation Rate by State and Subgroup

[3] An Intro to Data for Diploma by devpost.com
Data for Diploma by devpost.com - Resources
Data for Diploma by devpost.com - Submissions

[4] ML Predicts School Dropout Risk & Boosts Graduation Rates
Predicting student dropout risks, increasing graduation rates with cloud analytics.

[5] GeneLesinskiaStevenCornsbCihanDagli, "Application of an Artificial Neural Network to Predict Graduation Success at the United States Military Academy", Oct 2016

[6] Stamos T. Karamouzis and Andreas Vrettos, "n Artificial Neural Network for Predicting Student Graduation Outcomes ", Oct 2008

[7] Suchita Borkar, K.Rajeswari, "Attributes Selection for Predicting Students' Academic Performance using Education Data Mining and Artificial Neural Network"

[8] Making better use of graduation rates to assess school success

[9] Cited: Seabold, Skipper, and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. Proceedings of the 9th Python in Science Conference. 2010.

[10] Cited: Does scikit-learn have forward selection/stepwise regression algorithm?

[11] Sebastian Raschka, Aug 2014 Predictive modeling, supervised machine learning, and pattern classification — the big picture

[12] Cited: Comparison of kernel ridge regression and SVR

1. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
2. API design for machine learning software: experiences from the scikit-learn project, Buitinck et al., 2013.

[13] Jason Brownlee, Ph.D. How To Implement Baseline Machine Learning Algorithms From Scratch With Python

[14] A Machine Learning Approach to Prioritizing Students at Risk of not Graduating High School on Time

[15] Predicting STEM attrition using student transcript data

[16] Application of an Artificial Neural Network to Predict Graduation Success at the United States Military Academy, 2016, by Gene Lesinskia, Steven Cornsb, Cihan Dagli

[17] 7 tools in every data scientist's toolbox

[18] Big Data in Education 1.1 Introduction

[19] Data Science by Dr. Saed Sayad

[20] Education in the United States

[21] GradNation leaders sound the alarm as U.S. remains off-track to reaching 90 percent

[22] Predictors of Postsecondary Success

[23] Progress and Challenge in Ending the High School Dropout Epidemic in 2015 Building a Grad Nation report

[24] Public High School Graduation Rates

[25] Stability Selection by Nicolai Meinshausen and Peter Buhlmann, in 2009

[26] Supervised and Unsupervised Machine Learning Algorithms

[27] Supervised learning

[28] U.S. Department of Education

[29] Udacity MLND Capstone Project Description - Education

[30] Udacity Machine Learning Enginner Nanodegree Program

[31] Unsupervised learning

[32] Prediction of Graduation Delay Based on Student Characteristics and Performance

[33] Selecting good features – Part IV: stability selection, RFE and everything side by side, Ando Saabas

[34] Feature Selection methods with an example (or how to select the right variables?)

[35] (1) Highlighting Outliers in your Data with the Tukey Method (2) Outlier: Tukey's fences

Appendix 1: Python Source and Execution Result

The jtmoogle APIs were developed for Predicting US High School Graduation Success capstone report.

- The sources files for APIs were available in GitHub at jtmoogle
- The runme.pdf listed execution steps and output result

Appendix 2: Fields from feature selection in the proposal phase proposal data analysis

Features	Classifier process	Regression process	Description
County.1	include	include	County ID Number
CWD_COHORT_1112	include		Number of children with disabilities in the graduation cohort
leaid11	include		Local Education Agency (district) NCES ID
MAS_COHORT_1112	include		Number of Asian/Pacific Islander students in the graduation cohort
MBL_COHORT_1112	include	include	Number of Black students in the graduation cohort
MHI_COHORT_1112	include		Number of Hispanic students in the graduation cohort
pct_Age5p_German_ACS_08_12		include	percentage of population ages 5 years and over who speak English less than "very well" and speak German at home.
pct_Age5p_Navajo_ACS_08_12	include	include	percentage of population ages 5 years and over who speak English less than "very well" and speak Navajo at home
pct_Age5p_OthPacIsl_ACS_08_12	include		percentage of population ages 5 years and over who speak English less than "very well" and speak some other Pacific Island language at home
pct_Age5p_Scandinav_ACS_08_12		include	percentage of population ages 5 years and over who speak English less than "very well" and speak a Scandinavian language at home
pct_Age5p_WGerman_ACS_08_12		include	percentage of population ages 5 years and over who speak English less than "very well" and speak another West Germanic language at home
pct_Census_UAA_CEN_2010	include	include	percentage of addresses was returned to the Census with the postal code "Undeliverable as Addressed"
pct_Civ_unemp_16p_ACS_08_12		include	percentage of civilians ages 16 years and over in the labor force that are unemployed Persons
pct_College_ACS_08_12	include	include	percentage of population aged 25 years and over that have a college degree or higher
			The percentage of all ACS occupied housing units with a female

Features	Classifier process	Regression process	Description
pct_Female_No_HB_ACS_08_12	include	include	householder and no husband of householder present
pct_Females_CEN_2010		include	percentage of population that is female
pct_HHD_PPL_Und_18_CEN_2010	include	include	percentage of all census occupied housing units where one or more people are ages 18 years or under
pct_Hispanic_CEN_2010	include	include	percentage of total population that identify as "Mexican", "Puerto Rican", "Cuban", or "another Hispanic, Latino, or Spanish origin"
pct_Inst_GQ_CEN_2010	include	include	percentage of Census population who live in group quarters and are primarily ineligible, unable, or unlikely to participate in labor force while residents
pct_Males_CEN_2010	include	include	percentage of Census total population that is male Persons 2010
pct_MLT_U10p_ACS_08_12	include	include	percentage of all ACS housing units that are in a structure that contains 10 or more housing units
pct_Mobile_Homes_ACS_08_12	include		percentage of all ACS housing units that are considered mobile homes
pct_MrdCple_HHD_CEN_2010	include		percentage of all Census occupied housing units where the householder and his or her spouse are listed as members of the same household;
pct_NH_AIAN_alone_ACS_08_12	include	include	percentage of total population indicate no Hispanic origin and their only race as "Asian Indian", "Chinese", "Filipino", "Korean", "Japanese", "Vietnamese", or "Other Asian"
pct_NH_AIAN_alone_CEN_2010		include	percentage of population indicate no Hispanic origin and their only race as "American Indian or Alaska Native" or report entries such as Navajo, Blackfeet, Inupiat, Yup'ik, or Central/South American Indian
pct_NH_Blk_alone_CEN_2010	include	include	percentage of total population indicate no Hispanic origin and their only race as "Black, African Am., or Negro" or report entries such as African American, Kenyan, Nigerian, or Haitian
pct_NH_NHOPI_alone_ACS_08_12	include		percentage of the population that indicate no Hispanic origin and their only race as "Native Hawaiian", "Guamanian or Chamorro", "Samoan", or "Other Pacific Islander"
pct_NH_NHOPI_alone_CEN_2010		include	percentage of the 2010 Census total population that indicate no Hispanic origin and their only race as "Native Hawaiian", "Guamanian or Chamorro", "Samoan", or "Other Pacific Islander"
pct_NO_PH_SRVC_ACS_08_12		include	ercentage of ACS occupied housing units that do not have a working telephone and available service
pct_No_Plumb_ACS_08_12		include	percentage of all ACS housing units that do not have complete plumbing facilities
pct_Pop_25_44_CEN_2010		include	percentage of the 2010 Census total population that is between 25 and 44 years old
pct_Pop_45_64_ACS_08_12		include	percentage of the ACS population that is between 45 and 64 years old
pct_Pop_45_64_CEN_2010	include		percentage of the Census population that is between 45 and 64 years old
pct_Pop_5_17_CEN_2010	include		percentage of the Census population that is between 5 and 17 years old
pct_Pop_Under_5_CEN_2010	include	include	percentage of Census population that is under under 5 years old
pct_Prs_Blw_Pov_Lev_ACS_08_12	include	include	percentage of the ACS eligible population that are classified as below the poverty level given their total family or household income
pct_PUB_ASST_INC_ACS_08_12		include	percentage of all ACS occupied housing units that receive public assistance income
pct_Rel_Under_6_CEN_2010		include	percentage of 2010 Census family-occupied housing units with a related child under 6 years old
pct_Sngl_Prns_HHD_CEN_2010	include		percentage of all 2010 Census occupied housing units where a householder lives alone
pct_TEA_Update_Leave_CEN_2010	include	include	percentage of addresses from which a Census form was expected to be delivered for mail return that were in an Update/Leave type of enumeration area in the 2010 Census
pct_URBAN_CLUSTER_POP_CEN_2010		include	percentage of the 2010 Census total population that lives in a densely settled area containing 2,500 to 49,999 people
pct_Vacant_CEN_2010		include	percentage of addresses in a 2010 Census mailback area that were confirmed as vacant housing units
pct_Vacant_Units_ACS_08_12	include		percentage of all ACS housing units where no one is living regularly at the time of interview; units occupied at the time of interview entirely by persons who are staying two months or less and who have a more permanent residence elsewhere are classified as vacant
PUB_ASST_INC_ACS_08_12		include	Number of ACS households that receive public assistance income
State 1		include	State ID Number

	Classifier	Regression	

> For any questions or questions, please contact jtmoogle@gmail.com