WalletHub Data Scientist Test Modeling Project Report

Aiden V. Johnson
December 17, 2018
Project Overview
Goal: Create a model to predict the target variable y
Data Sources:
A csv file containing 100,000 rows and 304 labeled features: x001 to x304
Classification or Regression Model: Regression

Modeling Assumptions

- The variables are unique and don't include any duplicates with unique names.
- Variable collinearity and potential synonymousness will be managed with modeling techniques implemented.
- There are no outliers in the dataset, that require mitigation prior to modeling via IQR or z-score evaluation.

Modeling Methodology Description

After reading in the dataset the null values were identified, reviewed for counts and distribution across the dataset. The data size allowed for column-wise dropping of null values. Imputation of null values is considered another reasonable approach, however, the number of features remaining after column-wise dropping of null values did not necessitate imputation.

The large number of features indicate an opportunity to apply dimension reduction techniques in the modeling process. The data was first scale standardized and then a Principle Component Analysis was applied to capture the majority of the variance in the data. Reviewing the plot of the cumulative explained variance against the number of components indicates 96 components are needed to capture 95% of the total variance in the data set.

The data was split into a 70/30 training and testing set, in order to allow for holdout model performance evaluation.
Based on the limited exploratory analysis completed and potential for variable correlation the

two modeling algorithms chosen for model development include; Lasso Regression and Random Forest Regression. Both of these techniques can effectively optimize in high feature dimensionality. Using grid search optimization with a testing holdout sample the best parameters for each model were identified. Each grid search optimization was tested first on the 96 components making up the features for training data and then as the combination of the 96 PC's and the original training set concatenated together, for a total of 263 features. The combination of the PC features + the original features provided the best model performance metrics for both the Lasso and the Random Forest regressor.

The final model chosen was the Random Forest regression. The final step in the modeling process was to read in the entire dataset and re-train the random forest model on the entire data set rather than the initial training data subset.

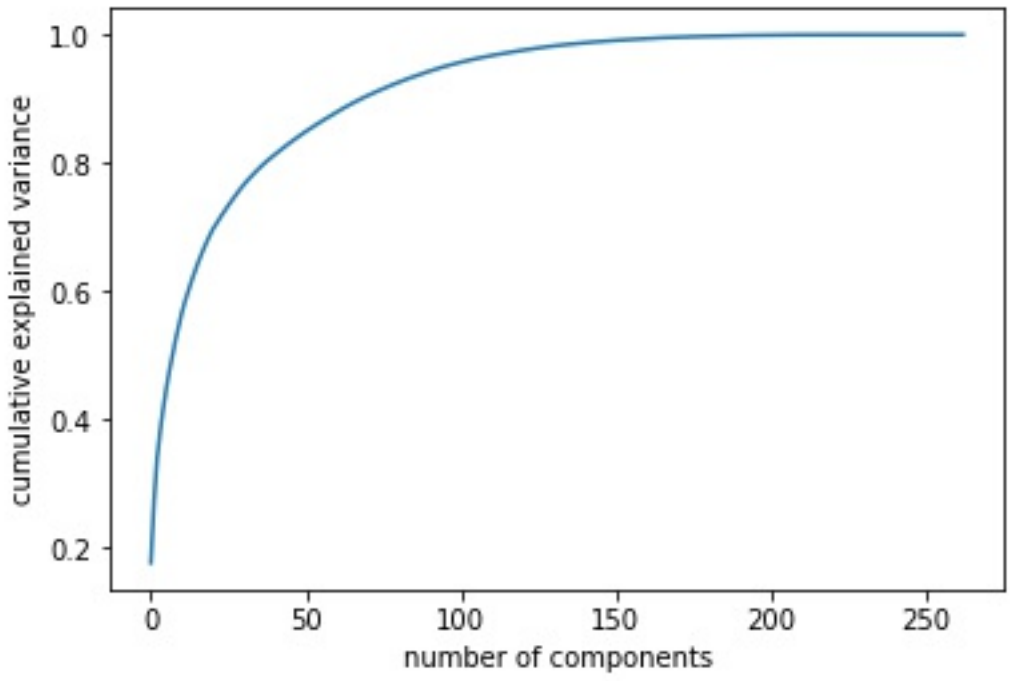
Modeling Methodology Overview

Data Processing and Model Development Steps

1. Drop null values

2. Apply a standard scaler to X values

3. Apply Principle Components Analysis - Plot below is Percent Explained Variance by Number of PC's



4. Split into Test and Train data subsets

5. Train Lasso Regression models
   5.1. Trained via alpha cross validation optimization on train transformed into the 96 principle components making up 95% of the variance
   5.2. Evaluate model RMSE with test set transformed into the 96 principle components
   5.3. Concatenate 96 PC's + train set -> Trained Lasso via cv approach
   5.4. Evaluate model RMSE with 96 PC's + test set

vi. Train RandomForestRegressor models
   6.1. Trained via grid search cross validation optimization on the training set transformed into the 96 principle components making up 95% of the variance
   6.2. Evaluate model RMSE with test set transformed into the 96 principle components
   6.3. Concatenate 96 PC's + train set -> Trained RFR via cv approach
   6.4. Evaluate model RMSE with 96 PC's + test set

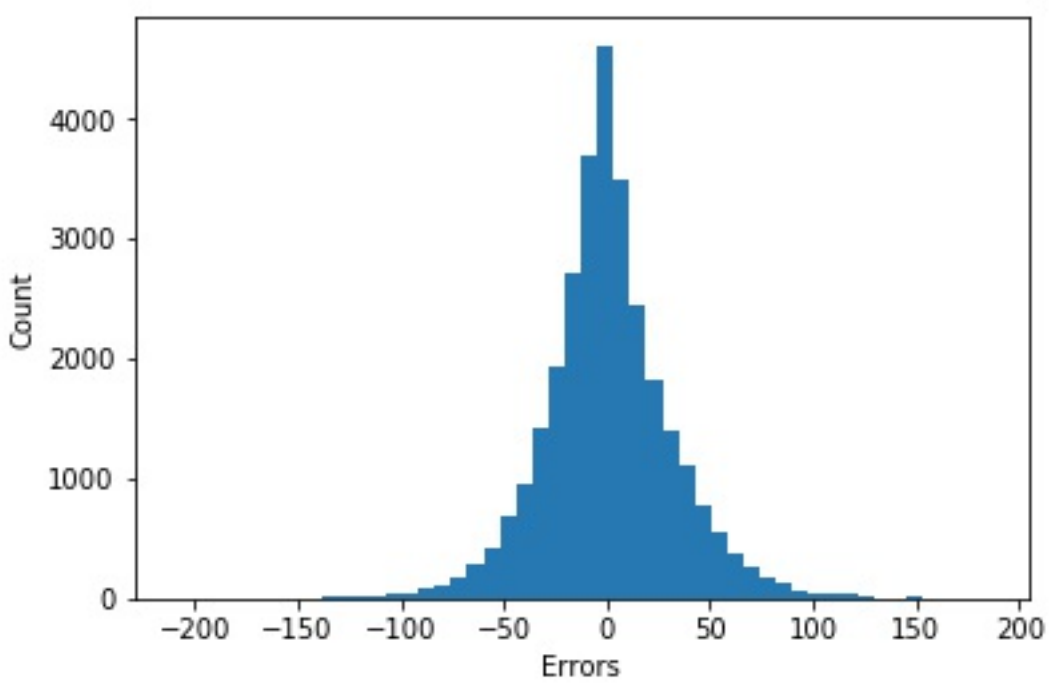
vii. Identified final model based on best performance metrics

Final Model Description
- Random Forest Regressor
- random_state=0
- n_estimators=200
- max_depth=None
- max_features=10
- min_samples_leaf=1
- min_samples_split=5

Performance Metrics and Error Distribution Review
Performance metrics were calculated by fitting a 30% hold out test set with the final trained model.

RMSE = 31.09
Accuracy = 93.11%

Absolute Error Distribution



Modeling Discussion

Potential improvements for future work as needed. The following steps could improve overall model performance and would be implemented if time allowed before model was moved to production.

- Develop the EDA more:
  - Review all variable distributions via box plots or two way plots with y variable
  - Remove near zero variance variables
  - Remove 100% unique variables - often an id column
  - Check for identical variables in terms of cdf or distributions
  - Remove variables that are more than 0.90 correlated
  - Identify and remove Outliers based on interquartile ranges or z-scores
- Feature Engineering:
  - Create K-means clustering
  - Use elbow plot or gap distance to iteratively determine best k
- Additional Machine Learning models to test:
  - Create a Neural Network model in keras on tensorflow
  - Gradient Boosted regression model

Model Implementation Instructions

Requirements:

- Python version 3.6.1

Packages:

- pandas
- numpy
- sklearn
- joblib
- math
- pickle
- warnings
- os

Steps:

Download and Unzip the `RF_ML.zip`

- Save the `run_model.py` and `rf_model.joblib` to your working directory

- Open the command line and change the directory to where you saved `run_model.py` and `rf_model.joblib`

- Type `python run_model.py` and press Enter

- Once you see a message that says `Enter file name:` type the name of your input CSV dataset for example `file.csv`

- Model should run in one minute or less on a laptop with 16GB of RAM

- When the model is complete you will see both the RMSE and MAE printed in the command line.

- The program will have written the model predictions out to a file in the same directory calling it `predictions.csv`