Evaluation Metrics

We have used MAPE as an evaluation metrics for our models :

MAPE stands for Mean Average Percentage Error. The MAPE of a model gives us an idea about how well it can forecast. First, accuracy measurements are useful when deciding which model to use for forecasting. Typically, the model with the best (lowest) MAPE should be used for forecasting. We calculate the MAPE on a test set, a small period of observations that occur immediately prior to the start of our intended prediction.

Linear Regression :

Linear regression is used to predict a quantitative response Y from the predictor variable X.

It is made with an assumption that there's a linear relationship between X and Y.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

Where:

- *y* is the response
- *β* values are called the **model coefficients**. These values are "learned" during the model fitting/training step.
- *β0* is the intercept
- *β1* is the coefficient for *X1* (the first feature)
- *βn* is the coefficient for *Xn* (the nth feature)

Parameters used by us :

Y = int_rate
X = grade_C, grade_D, grade_E, grade_F, grade_G, total_rec_int, total_pymnt_inv, funded_amnt_inv, sub_grade_B5, sub_grade_B4, sub_grade_C5, sub_grade_C4, sub_grade_C3, sub_grade_B4, sub_grade_D5

Alpha = 1

Significance of parameters used :

Y (Target Variable) = Y is a required variable**. We s**pecify the column to use as the dependent variable.

- For a regression model, this column must be numeric (Real or Int).

We have used int_rate as a target variable because we have based all our evaluation and modeling with respect to interest rate.

X (Predictor Variable) = X is also required variable. We specify the names of the columns on which the target variable depends.These are called independent variables.

We have used grade_C, grade_D, grade_E, grade_F, grade_G, total_rec_int,total_ pymnt_inv, funded_amnt_inv, sub_grade_B5, sub_grade_B4, sub_grade_C5, sub_grade_C4, sub_grade_C3, sub_grade_B4, sub_grade_D5 as our predictor variable as after using feature selection tools like lasso cross validation and feature hasher we came to a conclusion that these are the best predictors.

Alpha = 1

The alpha parameter controls the distribution between the ℓ1ℓ1 (LASSO) and ℓ2ℓ2 (ridge regression) penalties. The penalty is defined as
P(α,β)=(1−α)/2||β||22+α||β||1=∑j[(1−α)/2β2j+α|βj|]

a value of 1.0 represents LASSO, and a value of 0.0 produces ridge regression.

MAPE obtained after Linear Regression

The value of MAPE obtained after linear regression is : 8.50

```
[492]: test_int = y_test.tolist()
       train_int = y_train.tolist()

[493]: #calculate mape
       mape = np.mean(np.abs((test_int-predictions) / test_int)) * 100
       print('MAPE of test data:',mape)
       mape1 = np.mean(np.abs((train_int-predictionx) / train_int)) * 100
       print('MAPE of train data:',mape1)

       MAPE of test data: 8.517685129540919
       MAPE of train data: 8.506494286461157
```

Auto ML – General Linear Regression

Generalized Linear Models (GLM) estimate linear models for outcomes following exponential distributions.

Parameters used for GLM in h2o

Family : Gaussian

family: Specify the model type.If the family is **gaussian**, the data must be numeric (**Real** or **Int**).Since we had converted all the categorical data to numerical data we use Gaussian family for linear regression.

Lambda_search : True

lambda_search: Lambda is the regularization strength. Specify whether to enable lambda search, starting with lambda max (the smallest λλ that drives all coefficients to zero). We can always put in the lambda manually but since GLM automatically uses the best value of lambda and uses the same for further analysis we used lambda_search.

MAPE obtained after Generalized Linear Model

MAPE : 8.53

```
[565]: test_predict = glm.predict(test)
       test_predict

       glm prediction progress: |████████████████████████████████████████| 100%
```

predict
13.5339
12.0714
8.79649
8.63606
8.85729
8.73362
16.0235
11.9873
12.7165
9.03468

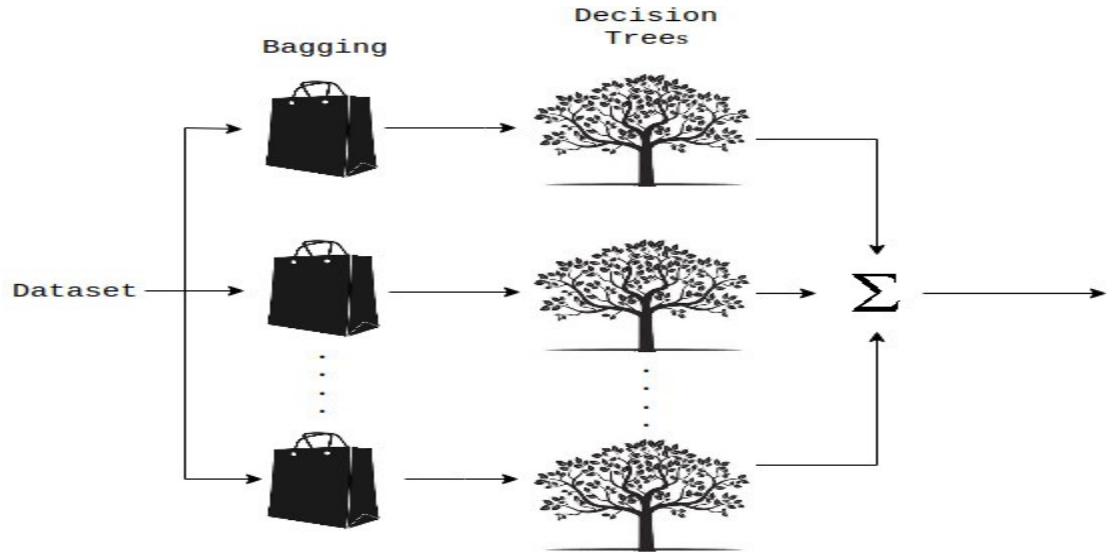
```
[565]:

[566]: MAPE(test, test_predict)

[566]: array([8.537])
```

Interpretations after comparing MAPE observed from linear regression and GLM (h20)

1. *Interpretability : We observed that MAPE observed from manual linear regression and Auto ML H2o GLM differ from each other by 0.03 . So it can be clearly deduced that the two methods aren't very different from each other and it's safe to say that these two methods can be used interchangeably.*
2. *Reproducibility : Since reproducibility is defined by seed specified by the random number generator (RNG) seed for algorithm components dependent on randomization. The seed is consistent for each H2O instance so that we can create models with the same starting conditions in alternative configurations.So it can be concluded that linear regression is reproducible.So it can be predicted that if we create different models using the same seed and parameters as we did both the models would predict the same MAPE.*

Random Forest :

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees



Parameters used by us :

Y = int_rate
X = grade_C, grade_D, grade_E, grade_F,grade_G, total_rec_int, total_pymnt_inv, funded_amnt_inv, sub_grade_B5, sub_grade_B4, sub_grade_C5, sub_grade_C4, sub_grade_C3, sub_grade_B4, sub_grade_D5

n_estimators = 10
random_state = 42

Significance of parameters used :

1.n_estimators : (number of trees)

2.random_state : seed : Specify the random number generator (RNG) seed for algorithm components dependent on randomization. The seed is90op consistent for each H2O instance so that you can create models with the same starting conditions in alternative configurations.

MAPE obtained after Random Forest Regression

The value of MAPE obtained after linear regression is : 3.61

```
[517]: rf = RandomForestRegressor(n_estimators= 200,random_state=42)

[518]: evaluate (rf,train_features,test_features,train_labels,test_labels)

       Model Performance
       Average Error(Train Data): 0.1546 of int rate.
       Average Error(Test Data): 0.4197 of int rate.
       Accuracy(Train Data) = 98.67%.
       Accuracy(Test Data) = 96.39%.
       Mape(Train Data): 1.3274 of int rate
       Mape(Test Data): 3.6064 of int rate
```

Auto ML – Distributed Random Forest

Distributed Random Forest (DRF) is a powerful classification and regression tool. When given a set of data, DRF generates a forest of classification or regression trees, rather than a single classification or regression tree. Each of these trees is a weak learner built on a subset of rows and columns. More trees will reduce the variance. Both classification and regression take the average prediction over all of their trees to make a final prediction, whether predicting for a class or numeric value.

Parameters used for DRF in h2o

Y = int_rate
X = grade_C, grade_D, grade_E, grade_F, grade_G, total_rec_int, total_pymnt_inv, funded_amnt_inv, sub_grade_B5, sub_grade_B4, sub_grade_C5, sub_grade_C4, sub_grade_C3, sub_grade_B4, sub_grade_D5

max_depth : 60

ntrees : 10

Stopping_rounds : 2

Seed : 1000000

score_each_iteration : True

Significance of parameters used :

1. **max_depth :** max_depth: Specify the maximum tree depth. Higher values will make the model more complex and can lead to overfitting. Setting this value to 0 specifies no limit. This value defaults to 20.
2. **ntrees :** ntrees: Specify the number of trees.
3. **Stopping_rounds :** stopping options are used to increase performance by restricting the number of models that get built. The model will stop if the **ratio** between the best moving average and reference moving average is more or equal **1-1e-3** (the misclassification is the less the better metric, for the more the better metrics the ratio have to be less or equal **1+1e-3** to stop).
4. **Seed** : Specify the random number generator (RNG) seed for algorithm components dependent on randomization. The seed is consistent for each H2O instance so that you can create models with the same starting conditions in alternative configurations.
5. **score_each_iteration :** score_each_iteration: (Optional) Enable this option to score during each iteration of the model training.

MAPE obtained after Distributed Random Forest Model

MAPE : 3.80

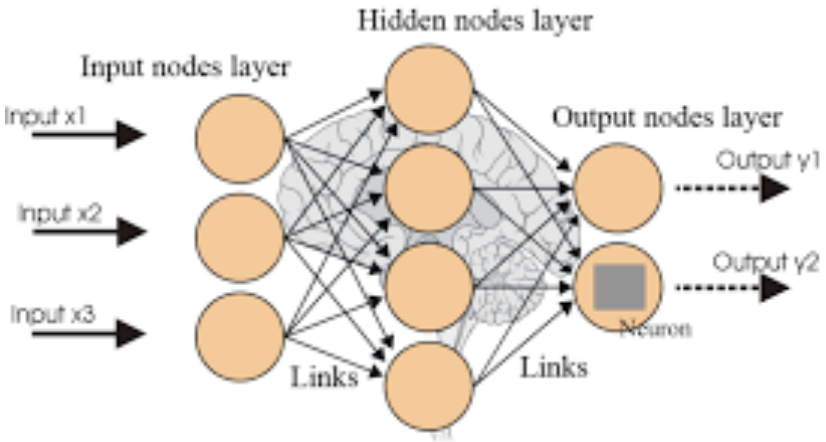
```
[563]: mape = 0

       for i, j in zip(test['int_rate'].as_data_frame().values, test_predict.as_data_frame()
           mape += np.abs((i-j)/i)

       mape = (mape/test.shape[0])*100

       mape

[563]: array([3.802])
```

Interpretations after comparing MAPE observed from random forest regression and DRF (h2o)

1. *Interpretability : We observed that MAPE observed from distributed random forest regression and Auto ML h2o GLM differ from each other by* ***0.2*** *. So it can be clearly deduced that the two methods aren't very different from each other and it's safe to say that these two methods can be used interchangeably.*
2. *Reproducibility : Since reproducibility is defined by seed specified by the random number generator (RNG) seed for algorithm components dependent on randomization. The seed is consistent for each H2O instance so that we can create models with the same starting conditions in alternative configurations. It can be clearly stated that if the seed and the rest of the parameters remain same for both the models different models would give same results*

Neural Networks :

Neural Networks are a class of models within the general **machine learning** literature. **Neural networks** are a specific set of algorithms that have revolutionized **machine learning**. They are inspired by biological **neural networks** and the current so-called deep **neural networks** have proven to work quite well



Parameters used by us :

Y = int_rate
X = grade_C, grade_D, grade_E, grade_F, grade_G, total_rec_int, total_pymnt_inv, funded_amnt_inv, sub_grade_B5, sub_grade_B4, sub_grade_C5, sub_grade_C4, sub_grade_C3, sub_grade_B4, sub_grade_D5

Optimizers : SGD

epochs : 15

mini_batch_size : 10

hidden : 2

Activation : RELU

Significance of parameters used :

Optimizers : Gradient descent is an iterative machine learning optimization algorithm to reduce the cost function.

epochs : Specify the number of times to iterate (stream) the dataset. The value can be a fraction.

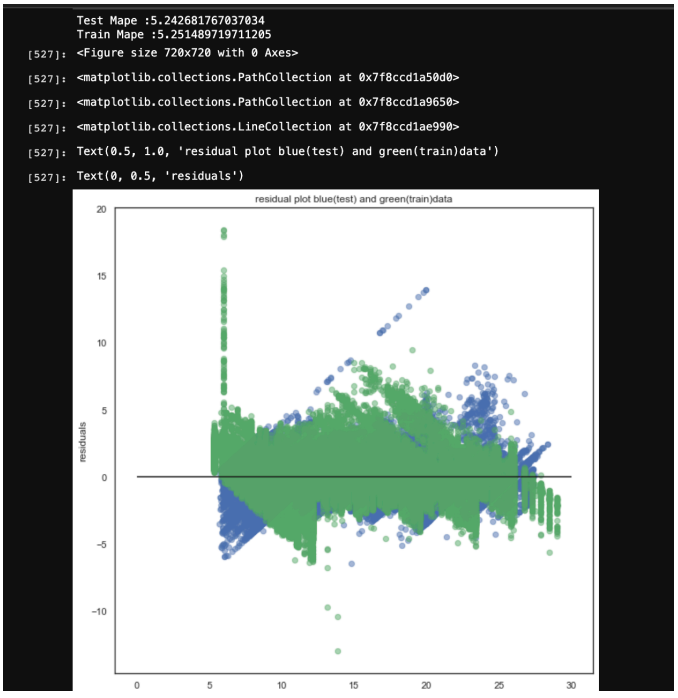
hidden: Specify the hidden layer sizes (e.g., 100,100). The value must be positive.

mini_batch_size: Specify a value for the mini-batch size. (Smaller values lead to a better fit; larger values can speed up and generalize better.)

activation: Specify the activation function (Tanh, Tanh with dropout, Rectifier, Rectifier with dropout, Maxout, Maxout with dropout).We used RELU (Rectified Linear Unit.).

MAPE obtained after Neural Network Regression

The value of MAPE obtained after neural network regression is :5.25



Auto ML – Deep Neural Network

H2O's Deep Learning is based on a multi-layer feedforward artificial neural network that is trained with stochastic gradient descent using back-propagation. The network can contain a large number of hidden layers consisting of neurons with tanh, rectifier, and maxout activation functions. Advanced features such as adaptive learning rate, rate annealing, momentum training, dropout, L1 or L2 regularization, checkpointing, and grid search enable high predictive accuracy. Each compute node trains a copy of the global model parameters on its local data with multi-threading (asynchronously) and contributes periodically to the global model via model averaging across the network.

Parameters used for DRF in h2o

Y = int_rate
X = grade_C, grade_D, grade_E, grade_F, grade_G, total_rec_int, total_pymnt_inv, funded_amnt_inv, sub_grade_B5, sub_grade_B4, sub_grade_C5, sub_grade_C4, sub_grade_C3, sub_grade_B4, sub_grade_D5

hidden = [50, 25]

activation = 'Tanh'

distribution = 'gaussian'

mini_batch_size = batch

epochs = ep

Significance of parameters used :

1. **epochs :** Specify the number of times to iterate (stream) the dataset. The value can be a fraction.
2. **hidden**: Specify the hidden layer sizes (e.g., 100,100). The value must be positive.
3. **mini_batch_size**: Specify a value for the mini-batch size. (Smaller values lead to a better fit; larger values can speed up and generalize better.)
4. **activation**: Specify the activation function (Tanh, Tanh with dropout, Rectifier, Rectifier with dropout, Maxout, Maxout with dropout).We used RELU (Rectified Linear Unit.).

MAPE obtained after Neural Network Model

MAPE : 4.7

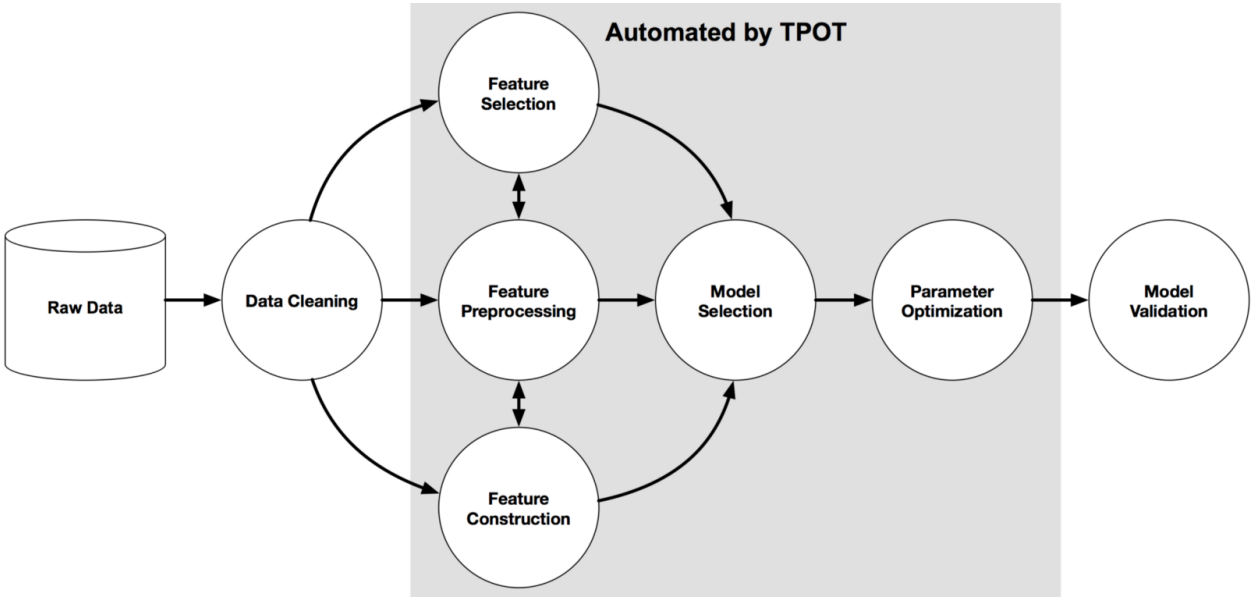
```
[569]: results

[569]: ['epoch: 10, batch: 10, MAPE: [5.684]',
        'epoch: 10, batch: 20, MAPE: [5.278]',
        'epoch: 10, batch: 40, MAPE: [5.453]',
        'epoch: 10, batch: 60, MAPE: [5.277]',
        'epoch: 10, batch: 80, MAPE: [5.078]',
        'epoch: 10, batch: 100, MAPE: [5.265]',
        'epoch: 50, batch: 10, MAPE: [5.459]',
        'epoch: 50, batch: 20, MAPE: [5.85]',
        'epoch: 50, batch: 40, MAPE: [5.403]',
        'epoch: 50, batch: 60, MAPE: [4.755]',
        'epoch: 50, batch: 80, MAPE: [5.362]',
        'epoch: 50, batch: 100, MAPE: [5.544]',
        'epoch: 100, batch: 10, MAPE: [5.315]',
        'epoch: 100, batch: 20, MAPE: [5.284]',
        'epoch: 100, batch: 40, MAPE: [5.308]',
        'epoch: 100, batch: 60, MAPE: [4.915]',
        'epoch: 100, batch: 80, MAPE: [5.157]',
        'epoch: 100, batch: 100, MAPE: [5.364]']
```

Interpretations after comparing MAPE observed from neural network and Deep Neural Network(h2o)

1. *Interpretability : We observed that MAPE observed from distributed random forest regression and Auto ML h2o GLM differ from each other by 0.5 . So it can be clearly deduced that the two methods aren't very different from each other and its safe to say that these two methods can be used interchangeably.*
2. *Reproducibility : Since reproducibility is defined by seed specified by the random number generator (RNG) seed for algorithm components dependent on randomization. Neural Network's cost function's is non convex and so solutions can get stuck in a local optima and there are multiple local optima. Also this depends on how you initialize the weights.*

Tpot in machine learning

TPOT is meant to be an assistant that gives you ideas on how to solve a particular machine learning problem by exploring pipeline configurations that you might have never considered, then leaves the fine-tuning to more constrained parameter tuning techniques such as grid search.



TPOT has what its developers call a genetic search algorithm to find the best parameters and model ensembles. It could also be thought of as a natural selection or evolutionary algorithm. TPOT tries a pipeline, evaluates its performance, and randomly changes parts of the pipeline in search of better performing algorithms.

Output obtained from TPOT :

```
[25]: mape = 0

      for i,j in zip(y_test, results):
          mape += np.abs((i-j)/i)
      mape = (mape*100)/y_test.shape[0]
      mape

[25]: 8.547534090813704
```

Auto SKlearn

Regressor by Auto-Sklearn when provided with the input of lending club dataset to get the outcome of Interest rate it ran across to predict the best model and provided the following outcome :

R2 score: 0.850566830796188

```
Possible set intersection at position 3
[(0.940000, SimpleRegressionPipeline({'data_preprocessing:categorical_transformer:categorical_encoding:__choice__': 'no_encoding', 'data
_preprocessing:categorical_transformer:category_coalescence:__choice__': 'no_coalescense', 'data_preprocessing:numerical_transformer:imp
utation:strategy': 'most_frequent', 'data_preprocessing:numerical_transformer:rescaling:__choice__': 'none', 'feature_preprocessor:__cho
ice__': 'nystroem_sampler', 'regressor:__choice__': 'liblinear_svr', 'feature_preprocessor:nystroem_sampler:kernel': 'rbf', 'feature_pre
processor:nystroem_sampler:n_components': 305, 'regressor:liblinear_svr:C': 18080.177016227895, 'regressor:liblinear_svr:dual': 'False',
'regressor:liblinear_svr:epsilon': 0.0017155220175954669, 'regressor:liblinear_svr:fit_intercept': 'True', 'regressor:liblinear_svr:inte
rcept_scaling': 1, 'regressor:liblinear_svr:loss': 'squared_epsilon_insensitive', 'regressor:liblinear_svr:tol': 7.035440846539455e-05,
'feature_preprocessor:nystroem_sampler:gamma': 0.00025844019785412086},
dataset_properties={
  'task': 4,
  'sparse': False,
  'multioutput': False,
  'target_type': 'regression',
  'signed': False})),
(0.060000, SimpleRegressionPipeline({'data_preprocessing:categorical_transformer:categorical_encoding:__choice__': 'one_hot_encoding',
'data_preprocessing:categorical_transformer:category_coalescence:__choice__': 'minority_coalescer', 'data_preprocessing:numerical_transf
ormer:imputation:strategy': 'median', 'data_preprocessing:numerical_transformer:rescaling:__choice__': 'robust_scaler', 'feature_preproc
essor:__choice__': 'random_trees_embedding', 'regressor:__choice__': 'decision_tree', 'data_preprocessing:categorical_transformer:catego
ry_coalescence:minority_coalescer:minimum_fraction': 0.027537836751886195, 'data_preprocessing:numerical_transformer:rescaling:robust_sc
aler:q_max': 0.9003944797470034, 'data_preprocessing:numerical_transformer:rescaling:robust_scaler:q_min': 0.2566610073208164, 'feature_
preprocessor:random_trees_embedding:bootstrap': 'True', 'feature_preprocessor:random_trees_embedding:max_depth': 5, 'feature_preprocesso
r:random_trees_embedding:max_leaf_nodes': 'None', 'feature_preprocessor:random_trees_embedding:min_samples_leaf': 1, 'feature_preprocess
or:random_trees_embedding:min_samples_split': 2, 'feature_preprocessor:random_trees_embedding:min_weight_fraction_leaf': 1.0, 'feature_p
reprocessor:random_trees_embedding:n_estimators': 10, 'regressor:decision_tree:criterion': 'friedman_mse', 'regressor:decision_tree:max_
depth_factor': 0.23884067765544847, 'regressor:decision_tree:max_features': 1.0, 'regressor:decision_tree:max_leaf_nodes': 'None', 'regr
essor:decision_tree:min_impurity_decrease': 0.0, 'regressor:decision_tree:min_samples_leaf': 11, 'regressor:decision_tree:min_samples_sp
lit': 13, 'regressor:decision_tree:min_weight_fraction_leaf': 0.0},
dataset_properties={
  'task': 4,
  'sparse': False,
  'multioutput': False,
  'target_type': 'regression',
  'signed': False})),
]
R2 score: 0.850566830796188
```