**A - Model Selection**

Selecting an adequate prediction model was a critical part of our project. Our model selection process consisted of thirty-six model run on three separate datasets. As described in the data engineering section above, each dataset used a distinct missing values imputation process. We then proceeded to select the dataset for which the models yield the best prediction capabilities. In this section, we describe the twelve model that we run before describing how we selected the Gradient Boosted Regressor as our best model.

***Models Description***

We run eight classification models and four regression models for a total of twelve models.

*Classification Models Run*

We run the following eight classification models: The Latent Dirichlet allocation (LDA) model, the K-Nearest Neighbors model, three Support Vector Machine models, the Gradient Boosted Classifier, and the Random Forest classifier. Below is a brief description of these classification models.

*The Latent Dirichlet allocation (Blei, Ng and Jordan, 2019)* - generative probabilistic model for collections of discrete data such as text corpora. LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities

*KNN – K-Nearest Neighbors* (*Pedregosa et al., 2019)-* The KMeans algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the *inertia* or within-cluster sum-of-squares (see below). This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields. The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean μj of the samples in the cluster. The means are commonly called the cluster “centroids”; note that they are not, in general, points from X, although they live in the same space. The K-means algorithm aims to choose centroids that minimize the inertia, or within-cluster sum-of-squares criterion

*SVM – Support Vector Machine (Gandhi, 2019)* - The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

*Gradient Boosted Classifier (Ke et al., 2019)* - Gradient boosting decision tree (GBDT) is a widely-used machine learning algorithm, due to its efficiency, accuracy, and interpretability. GBDT achieves state-of-the-art performances in many machine learning tasks, such as multi-class classification [2], click prediction , and learning to rank. *(Ke et al., 2019).* Boosting is an ensemble learning technique. Conceptually, these techniques involve: 1. learning base learners; 2. using all of the models to come to a final prediction. Additive modelling is at the foundation of Boosting algorithms. The idea is simple- form a complex function by adding together a bunch of simpler terms. In Gradient Boosting, a number of simpler models are added together to give a complex final model. (Mahto, 2019).

*Random Forest (Pedregosa et al., 2019)* – The random forests classifier is an ensemble learning technique. With the random forests classifier each tree in the ensemble is built from a sample drawn with replacement (i.e., a bootstrap sample) from the training set. In addition, when splitting a node during the construction of the tree, the split that is chosen is no longer the best split among all features. Instead, the split that is picked is the best split among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases (with respect to the bias of a single non-random tree) but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model.

*Regression Models Run*

We run the following four regression models: The Linear Regression model, the Gradient Boosted Regression model, the Random Forest Regressor, and the XGBoost model. Below is a brief description of these regression models.

*Linear Regression (Pedregosa et al., 2019)* – Simple linear regression has a form Y = aX + b where Y and X are called dependent and independent variables. These terms are interchangeably used with a response and explanatory variable. Multiple linear regression extends the polynomial with a greater number of independent variables such as Y= aX + bX1 + c where Y = Weight, X = Height, and X1 = Gender. If the goal is a prediction or error reduction, linear regression can be used to fit a predictive model to an observed dataset. Moreover, the technique can be applied to quantify the strength of the relationship if the goal is to explain variation in Y that can be attributed to variation in the Xs. (Kim, 2019)

*Gradient Boosted Regression (Pedregosa et al., 2019)* - Gradient Tree Boosting or Gradient Boosted Regression Trees (GBRT) is a generalization of boosting to arbitrary differentiable loss functions. GBRT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems. Gradient Tree Boosting models are used in a variety of areas including Web search ranking and ecology. The advantages of GBRT are:

* Natural handling of data of mixed type (= heterogeneous features)
* Predictive power
* Robustness to outliers in output space (via robust loss functions)

*Random Forest Regressor (Fabisch et al., 2019)* - A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The underlying principles and methods are for the most part similar to the one described above for the Random Forest Classifier.

*XGBoost (Chen and Guestrin, 2019)* - In his paper, Chen and Guestrin describe a scalable end- to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. They propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, the authors provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost can scale beyond billions of examples using far fewer resources than existing systems and yield better results.

*Testing of the Run Models*

After fitting the above models, we tested each model using the following techniques:

Classification models testing

We tested each classification model by computing each model fit accuracy, precision, recall and F1 score on the testing dataset.

* Accuracy is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.
* Precision is the number of positive predictions divided by the total number of positive class values predicted
* Recall  is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.
* F1 Score is the 2\*((precision\*recall)/(precision+recall)). The F1 score conveys the balance between the precision and the recall.

Below is a summary of the computed values:



Based on the above results we pre-identify the Gradient Boosted Classifier as a good prediction model for our specific problem subject to further testing.

Regression models testing

We tested each regression using the RMSE on the testing dataset.

Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. (Statistics How To, 2019)

Below is a summary of the computed values:

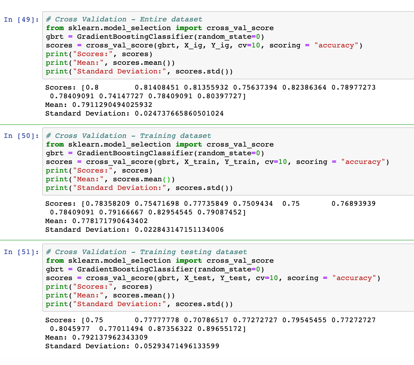


While most of our regression model were performing well base on these initial computed RMSE values, we decided to retain our good performing Gradient Boosted Classifier for further generalizability testing. This was with the option to come back to a good performing regression model if we were to identify some generalization issues related to the retained Gradient Boosted Classifier.

**B - Selected Model Validation – Gradient Boosted Classifier**

Our main goal was to ensure that our selected model was generalizable and not overfitting or underfitting. To that end, we run the following tests:

* Cross Validation - Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation. (Brownlee, 2019). We checked the validity of our accuracy score by running a 10 folds cross validation on the entire dataset, the test dataset, and the training datasets. Further we compared the cross-validation results to ensure their consistency thus that the model was not overfitting. Below is the result of our computation:



* Accuracy Paradox **-** The accuracy paradox is the paradoxical finding that accuracy is not a good metric for predictive models when classifying in predictive analytics. We checked for the accuracy paradox by running a confusion matrix and a Sklearn classification report. A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. (GeeksforGeeks, 2019). Below is the result of our computations:





* Area Under The Curve (AUC) Receiver Operating Characteristics (ROC) curve - AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is. (Narkhede, 2019). We checked how much our selected model is capable to distinguish between classes. Below is the result of our computations:



Based on the above test, we concluded that our Gradient Boosted Classification model is performing good predictions for our use case and is generalizable.

**References:**

Blei, D., Ng, A. and Jordan, M. (2019). *Latent Dirichlet Allocation*. [online] Jmlr.org. Available at: http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf [Accessed 4 Jun. 2019].

Gandhi, R. (2019). *Support Vector Machine — Introduction to Machine Learning Algorithms*. [online] Towards Data Science. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 [Accessed 4 Jun. 2019].

Mahto, K. (2019). *Demystifying Maths of Gradient Boosting*. [online] Towards Data Science. Available at: https://towardsdatascience.com/demystifying-maths-of-gradient-boosting-bd5715e82b7c [Accessed 4 Jun. 2019].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2019). *1.11. Ensemble methods — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/ensemble.html#forest [Accessed 4 Jun. 2019].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2019). *2.3. Clustering — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/clustering.html#k-means [Accessed 4 Jun. 2019].

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T. (2019). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. [online] Papers.nips.cc. Available at: https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf [Accessed 4 Jun. 2019].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2019). *sklearn.linear\_model.LinearRegression — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear\_model.LinearRegression.html [Accessed 4 Jun. 2019].

Kim, S. (2019). *Linear Regression*. [online] Towards Data Science. Available at: https://towardsdatascience.com/regression-analysis-linear-regression-239df26a94ac [Accessed 4 Jun. 2019].

Fabisch, A., Passos, A., Gollonet, A., Joly, A., Gorgolewski, C., Cournapeau, D., Warde-Farley, D., Duchesnay, E., Fabian Pedragosa, F., Louppe, G., Schreiber, J., Vanderplas, J., Grobler, J., Millman, J., Kastner, K., Buitinck, L., Kumar, M., Blondel, M., Brucher, M., Dawe, N., Losi, P., Prettenhofer, P., Rajagopalan, R., Layton, R., Weiss, R., Ghosh, S., Du, S., Jones, T., Dubourg, V., Michel, V., Fritsch, V. and Li, W. (2019). *3.2.4.3.6. sklearn.ensemble.GradientBoostingRegressor — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html [Accessed 4 Jun. 2019].

Fabisch, A., Passos, A., Gollonet, A., Joly, A., Gorgolewski, C., Cournapeau, D., Warde-Farley, D., Duchesnay, E., Pedragosa, F., Louppe, G., Schreiber, J., Vanderplas, J., Grobler, J., Millman, J., Kastner, K., Buitinck, L., Kumar, M., Blondel, M., Brucher, M., Dawe, N., Losi, P., Prettenhofer, P., Rajagopalan, R., Layton, R., Weiss, R., Ghosh, S., Du, S., Jones, T., Dubourg, V., Michel, V., Fritsch, V. and Li, W. (2019). *3.2.4.3.2. sklearn.ensemble.RandomForestRegressor — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html [Accessed 4 Jun. 2019].

Chen, T. and Guestrin, C. (2019). *XGBoost: A Scalable Tree Boosting System*. [online] Arxiv.org. Available at: https://arxiv.org/pdf/1603.02754.pdf [Accessed 4 Jun. 2019].

Brownlee, J. (2019). *Classification Accuracy is Not Enough: More Performance Measures You Can Use*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ [Accessed 4 Jun. 2019].

Statistics How To. (2019). *RMSE: Root Mean Square Error - Statistics How To*. [online] Available at: https://www.statisticshowto.datasciencecentral.com/rmse/ [Accessed 4 Jun. 2019].

GeeksforGeeks. (2019). *Confusion Matrix in Machine Learning - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/confusion-matrix-machine-learning/ [Accessed 4 Jun. 2019].

Brownlee, J. (2019). *A Gentle Introduction to k-fold Cross-Validation*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/k-fold-cross-validation/ [Accessed 4 Jun. 2019].

Narkhede, S. (2019). *Understanding AUC - ROC Curve*. [online] Towards Data Science. Available at: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5 [Accessed 4 Jun. 2019].