Data Science Capstone Project 1

Capstone Project : In-depth Analysis (Machine Learning)

Learning Objective

- Practice identifying which supervised and unsupervised learning techniques are best suited for your Capstone Project data.
- Utilize supervised and unsupervised learning techniques to build predictive models.

Criteria	Meets Expectations
Completion	❏ A 2-3 page report on the steps and findings from machine learning in-depth analysis, uploaded to GitHub.
Process and understanding	❏ The submission shows that the student applied appropriate techniques to build predictive models. ❏ The submission shows that the student applied steps to build predictive models for the data in their capstone project. ❏ The submission shows that a hypothesis was developed. ❏ The submission includes a justification of the machine learning technique, and features selection and evaluation metrics/techniques utilized.
Presentation	❏ The submission is complete and uploaded in full.

Excellence: The submission demonstrates use of innovative ways to visualize data and uses algorithms not covered in the course with good justification and understanding, or applied existing algorithms in an innovative way, perhaps with really clever feature design.

For reference, review how this interim project fits into the Overall Capstone Project 1 Rubric.

1. Framing the Problem

Supervised classification algorithms will be used to classify the building permit issue times into time ranges. A classification approach was used because regression-based approaches are sensitive to outliers, which are present in many of the independent variables.

The number of classes could largely affect model performance due to class imbalance. Binary class or three-class outcomes were considered for this project, although more classes may be used. The class time ranges were chosen to reduce class imbalance issues if possible, more specifically:

- Binary outcome:
    - class 0: 1 month to 3 months
    - class 1: > 3 months
- Multi-class outcome:
    - class 0: 1 month to 3 months
    - class 1: 3 months to 6 months
    - class 2: >6 months

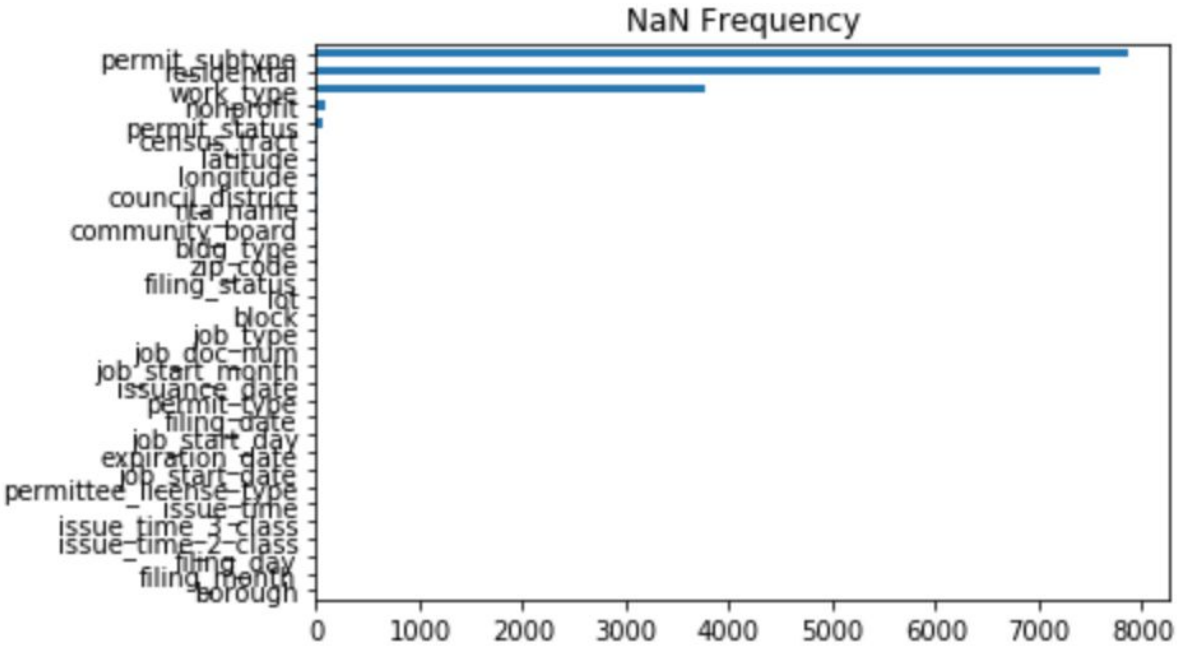
Some classification algorithms to be explored include:

- Logistic Regression (LR)
- K-Nearest Neighbors (KNN)
- Classification and Regression Trees (CART)
- Support Vector Machine (SVM)
- Random Forest (RF)
- Naive Bayes (NB)
- Gradient Boosted Trees (XGBoost)

2. Data Transformation

The filing date and the job start date were feature engineered to capture information about the month and day of week. Features were then converted to appropriate data types like category, int64, float64 and datetime64.

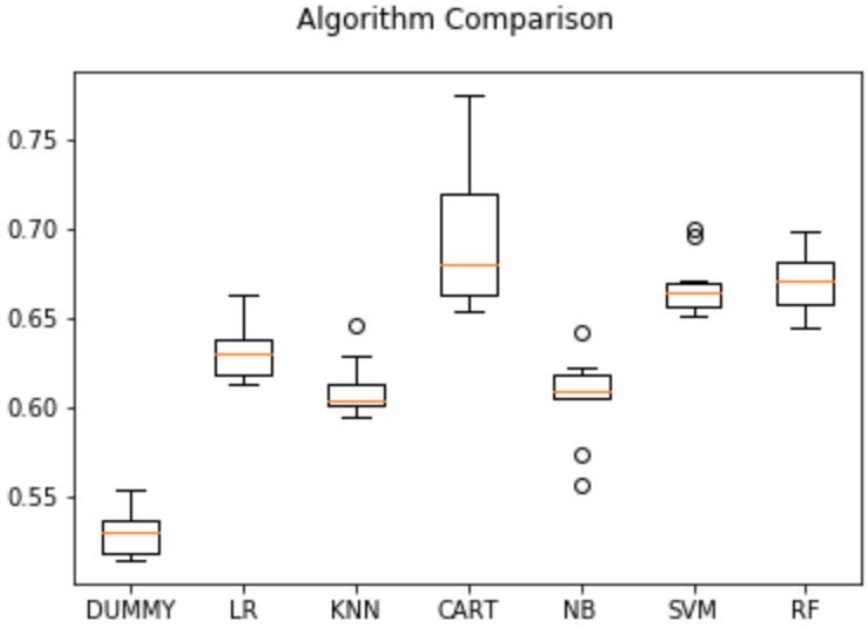
Dummy variables were created for the categorical features, even the NaN values. Missing/NaN values for the continuous variables (latitude and longitude) were imputed since only 23 out of 17589 records were missing.



Features were then scaled using the standard scale, which results in values between -1 and 1.

3. Compare Machine Learning Models

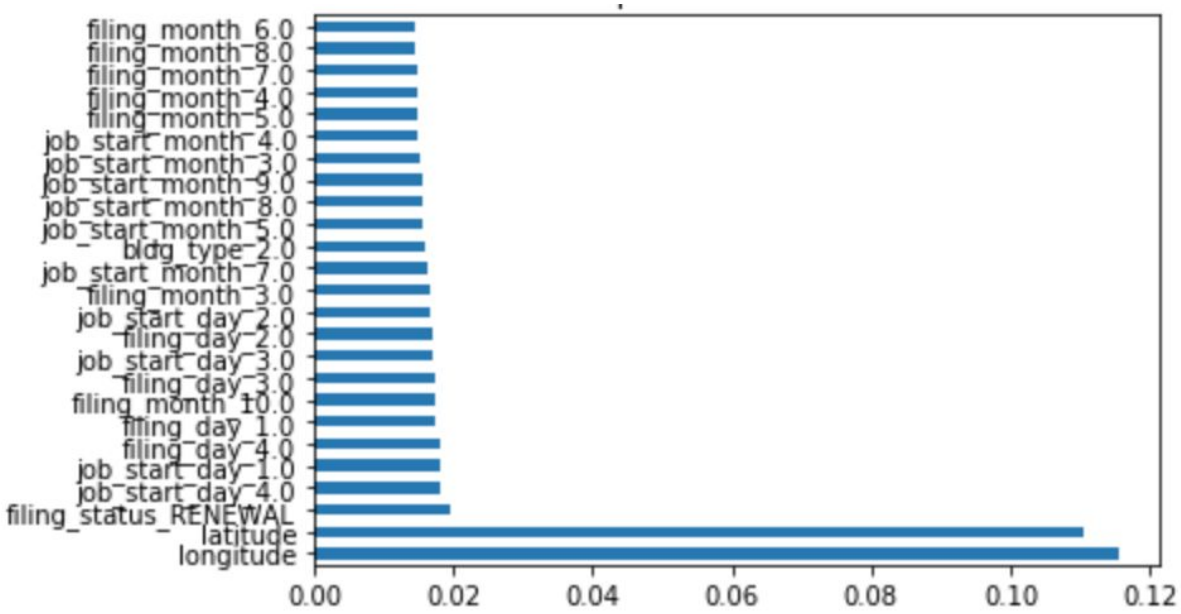
Before tuning hyperparameters and other optimizations, it'd be a good idea to figure out which algorithm may work best for our given problem. Let's try a mixture of modeling types, e.g. linear and nonlinear functions or parametric and nonparametric. Random Forest, CART and SVMs have the highest median accuracy based on a 10-fold cross-validation.



4. Feature Importance

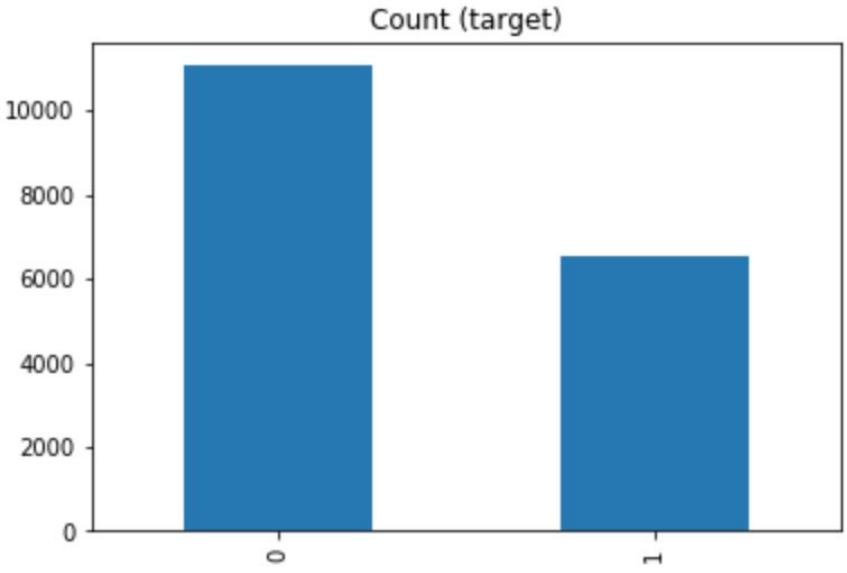
After splitting, let's identify the most important features in our dataset that contribute most to our prediction variable. Irrelevant features would decrease accuracy especially for linear algorithms, e.g. linear regression and logistic regression. Here are some other benefits:

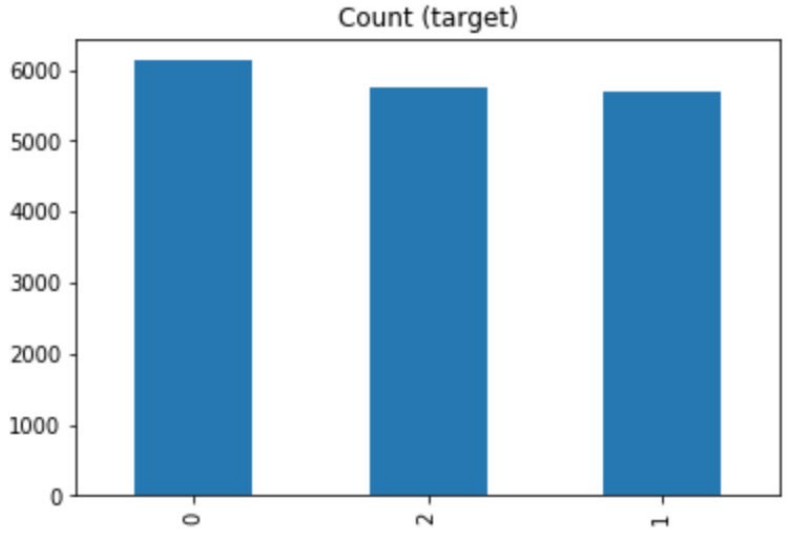
- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.
- Improves Accuracy: Less misleading data means modeling accuracy improves.
- Reduces Training Time: Less data means that algorithms train faster.



5. Class Imbalance

The classes were purposefully chosen to reduce the effects of class imbalance.





6. Model Enhancements

Based on the initial non-optimized runs of several kinds of algorithms in section 11, I will now focus on running tree-based classifiers as they seem to have the best initial metrics.
Some classification metrics to consider include:

- **Classification Accuracy**: Number of correct predictions made as a ratio of all predictions made
    - Note: Suitable when there are an equal number of observations in each class. Predictions and prediction errors are equally important
- **Logarithmic Loss**: Evaluates the predictions of probabilities of membership to a given class. The scalar probability between 0 and 1 can be seen as a measure of confidence for a prediction by an algorithm. Predictions that are correct or incorrect are rewarded or punished proportionally to the confidence of the prediction.
    - Note: Smaller logloss is better with 0 representing a perfect logloss.
- **Area Under ROC Curve**: Evaluates a model's ability to discriminate between positive and negative classes
    - Note: Used for binary classification problems. 1.0 = Perfect, 0.5 = Random
- **Confusion Matrix**: Presents the accuracy of a model with two or more classes
- **Classification Report**: Displays the precision, recall, F1-score and support for each class

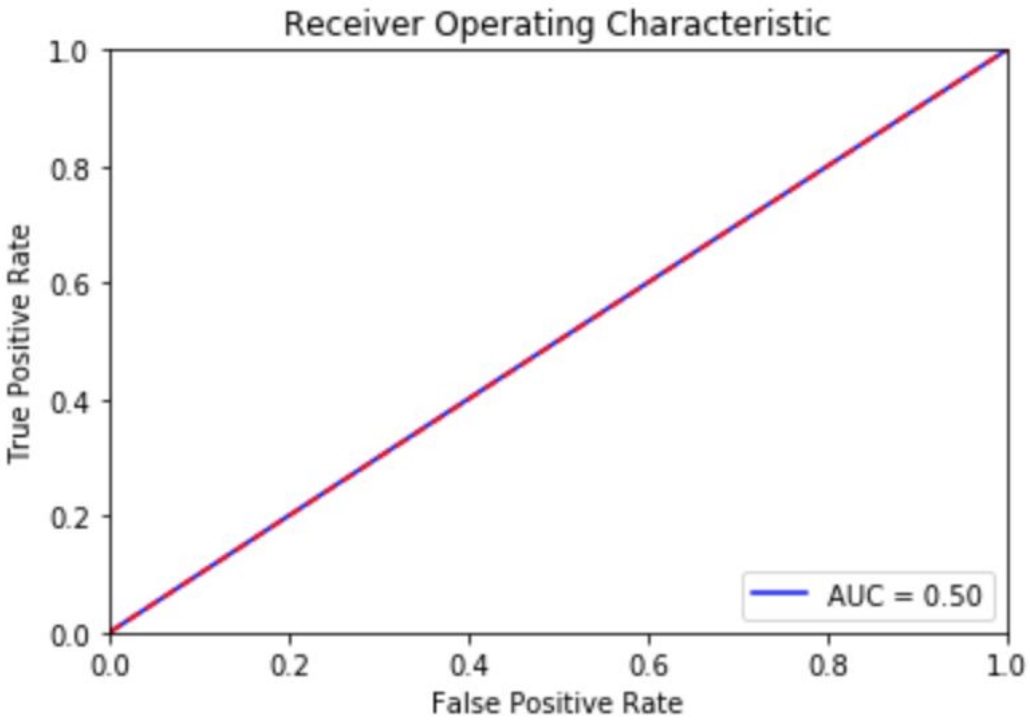
Starting with a Dummy Classifier, we may use this as a benchmark for other supervisied classification algorithms.

```
neg_log_loss : -15.922 (0.455)
accuracy : 0.537 (0.015)
Classification Report:
              precision    recall  f1-score   support

          0       0.63      0.63      0.63      2730
          1       0.39      0.39      0.39      1668

avg / total       0.54      0.54      0.54      4398
```

The Receiver Operator Characteristic (ROC) curve shows an almost 45 degree line, which is to be expected when a classifier is as good as flipping a coin



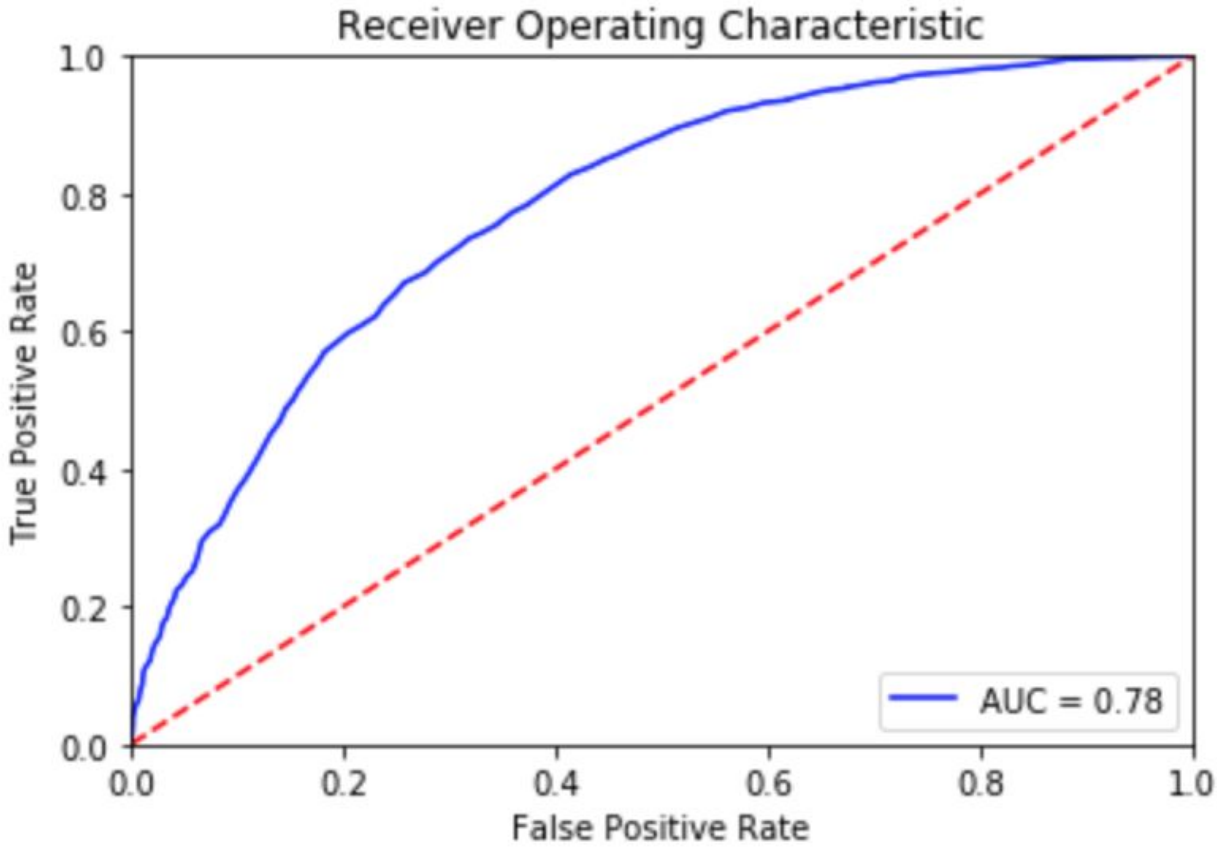
Random forest classifiers are an ensemble of decision tree classifiers. They have the key advantages of computational efficiency and an ability to handle high dimensions well. Because

randomness is introduced in the selection of records and features, the random forest reduces overfitting, which decision trees are more prone.

```
neg_log_loss : -0.565 (0.011)
accuracy : 0.691 (0.016)
Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.92      0.79      2730
           1       0.70      0.32      0.44      1668

avg / total       0.69      0.69      0.65      4398
```



Gradient Boosted Trees are sequentially grown trees that aims to improve predictions from previously grown trees. In boosting, because the growth of a particular tree takes into account the

other trees that have already been grown, smaller trees are typically sufficient. Using smaller trees can aid in interpretability as well; for instance, using stumps leads to an additive model.

Random Forest on the other hand typically does not use decision stumps, but uses rather complex trees.

```
neg_log_loss : -0.551 (0.016)
accuracy : 0.714 (0.022)
Classification Report:
             precision    recall  f1-score   support

          0       0.71      0.92      0.80      2730
          1       0.74      0.40      0.52      1668

avg / total       0.72      0.72      0.69      4398
```