**Assignment 2**

**Author**: Aditya Mulik

**NUID**: 002127694

**Email ID**: mulik.a@northeastern.edu

**Question 1**: 1. Try different imputation methods on the Titanic Dataset, and evaluate classifier accuracies for each of these. How would you pick the most optimal imputation method?

1. Different Imputation methods on the Titanic Dataset and the classifier accuracies

| **No** | **Imputation Method** | **Applied On** | **Data Type** | **Accuracy** |
| --- | --- | --- | --- | --- |
| 1 | median |  |  |  |
| 2 | mean |  |  |  |
| 3 | Mode (Most frequent) |  |  |  |
| 4 |  |  |  |  |

1. How to pick the most optimal imputation method?

**Question 2**: Briefly describe how gradient boosting differs from bagging. Compare the performance of kNN, random forest classifier and gradient boosting on the Titanic dataset.

1. Gradient Boosting vs Bagging

| **No** | **Gradient Boosting** | **Bagging** |
| --- | --- | --- |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

1. Comparison of performance of accuracies of KNN Classifier, Random Forest Classifier and XGBoost (gradient boosting)

| **No** | **Algorithm** | **Accuracy/ Performance** |
| --- | --- | --- |
| 1 | K-nearest Neighbours Algorithm | 0.77 |
| 2 | Random Forest Classifier | 0.74 |
| 3 | XGBoost Classifier | 0.82 |

**Note**:

**Question 3**: Theoretically, increasing the number of decision trees (n\_estimators), increases classifier performance and/or generalizability. How would you design and evaluate a computational experiment to test this, on the Titanic dataset? Do you find the same relationship between n\_estimators and performance? If not what might be one reason?

**Question 4**: Pick any Kaggle regression dataset. Train, tune and evaluate performance of a Random Forest Regression model. How will you use the feature importance calculations to perform feature selection? Please demonstrate this using the Kaggle regression dataset you picked.

**Kaggle Dataset:**

<https://www.kaggle.com/c/house-prices-advanced-regression-techniques/>

**Accuracy/ Performance**: 0.86