Knight-and-Queens Problem Using Local Search

Abhishek Jallawaram
Department of Computer Science (CS580)
George Mason University
VA, USA
ajallawa@gmu.edu or G01373042

***Abstract*—The document depicts the implementation of various machine learning algorithms to achieve the best model.**

***Index Terms*—K Nearest Neighbours, Decision Trees, Random Forests,Logistic Regression, K Nearest Neighbours Regression, Decision Tree Regression, Random Forest Regression**

I. Introduction

Implementation of machine learning algorithms to achieve the best classification model for bank-additional-full dataset and best regression model for Bug2 algorithm.

II. Problem Scenario

A. Classification

The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y).

B. Regression

The objective in motion planning is to compute a path from an initial position to a goal position while avoiding collisions with obstacles.

III. Classification

A. Data Pre-Processing

The given dataset is a combination of categorical and nominal attributes. The categorical data is converted into numerical data by using the Label encoder or One Hot Encoder.

Label Encoding: Assign each categorical value an integer value based on alphabetical order.

One Hot Encoding: Create new variables that take on values 0 and 1 to represent the original categorical values.

StandardScaler follows Standard Normal Distribution wherein, it makes mean = 0 and scales the data to unit variance.

The different classification methods are used on the scaled data to determine the best model which provides the best metrics. Classification Models used:

- K Nearest Neighbors
- Decision Trees
- Random Forests
- Logistic Regression

B. K Nearest Neighbors

K-NN classification is used to predict the rating of the test data based on the class label majority vote.The label that is most frequently represented around a given data point is used. It will be necessary to calculate the distance between the current point and the other data points in order to discover which data points are closest to a specific point. The K value in the K-NN algorithm defines how many neighbors will be checked to determine the classification of a specific point.In order to avoid either overfitting or underfitting, several values of k must be considered when defining it.The value of K is generated by K-fold cross validation by splitting the in a ratio. The algorithm uses Euclidean distance or cosine similarity as the similarity function and compares the test row with the K neighbours(number of neighbours to compare) and provides a probabilistic value to predict the rating. The different accuracy values for different features and features were used to determine the best model.

The K-value is selected using the cross validation to avoid overfitting and underfitting the data. K-fold.The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. Train Test split is performed to validate the accuracy. Different metrics have been used to generate the optimal value of k as shown below.

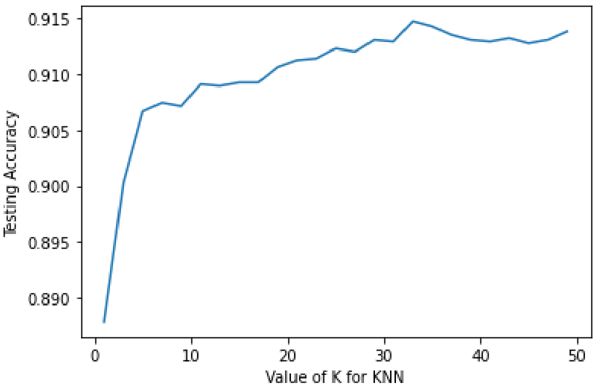


Fig. 1. Accuracy vs K.

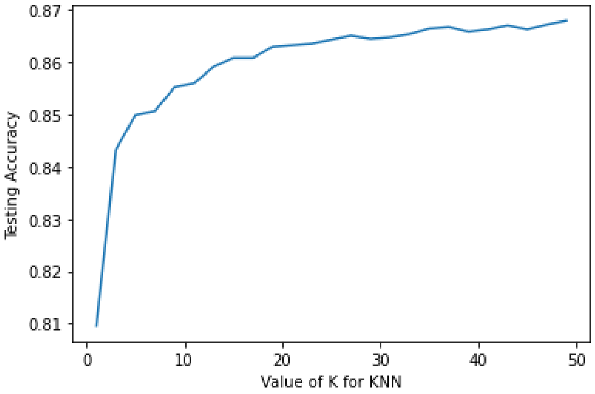


Fig. 2. Accuracy vs K.

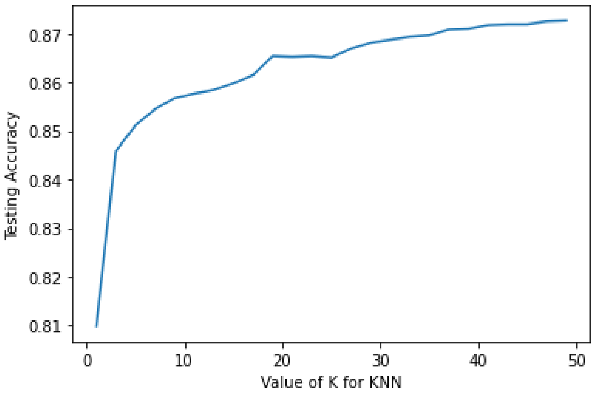


Fig. 3. Accuracy vs K.

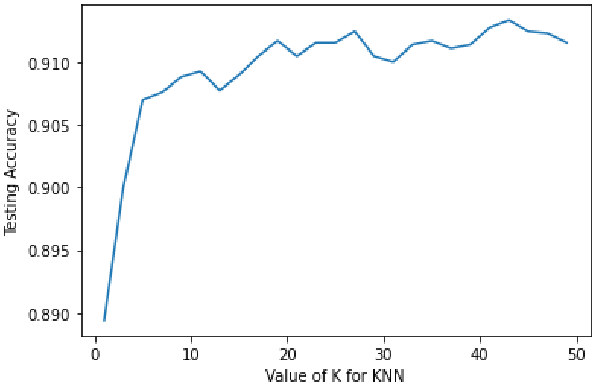
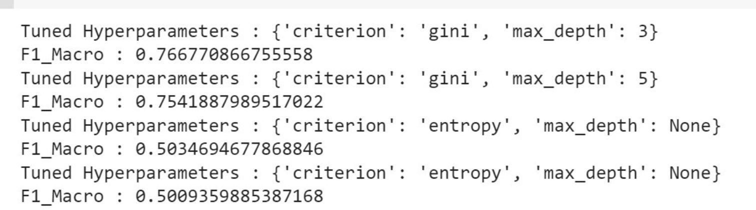


Fig. 4. Accuracy vs K.

C. *Decision Tree:*

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The hyper-parameters are tuned to attain the best F1 score.



```
Tuned Hyperparameters : {'criterion': 'gini', 'max_depth': 3}
F1_Macro : 0.766770866755558
Tuned Hyperparameters : {'criterion': 'gini', 'max_depth': 5}
F1_Macro : 0.7541887989517022
Tuned Hyperparameters : {'criterion': 'entropy', 'max_depth': None}
F1_Macro : 0.5034694677868846
Tuned Hyperparameters : {'criterion': 'entropy', 'max_depth': None}
F1_Macro : 0.5009359885387168
```

Fig. 5. Hyper Parameter Tuning

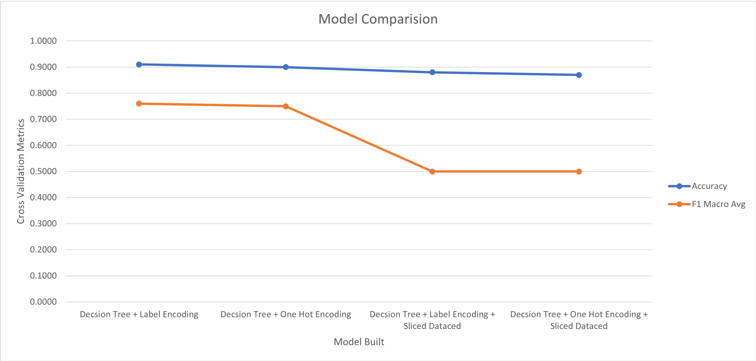


Fig. 6. Accuracy vs F1 Score

D. *Random Forest:*

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

- Random Forest + Label Encoding
  Tuned Hyperparameters :
  {’criterion’: ’gini’, ’max_depth’: None}
  F1_Macro : 0.757156159941509
- Random Forest + One Hot Encoding
  Tuned Hyperparameters :
  {’criterion’: ’gini’, ’max_depth’: None}
  F1_Macro : 0.7287216879056897
- Random Forest + Label Encoding + Sliced Data
  Tuned Hyperparameters :
  {’criterion’: ’gini’, ’max_depth’: None}
  F1_Macro : 0.4847235735331929
- Random Forest + One Hot Encoding + Sliced Data
  Tuned Hyperparameters :
  {’criterion’: ’gini’, ’max_depth’: None}
  F1_Macro : 0.48152371054911497

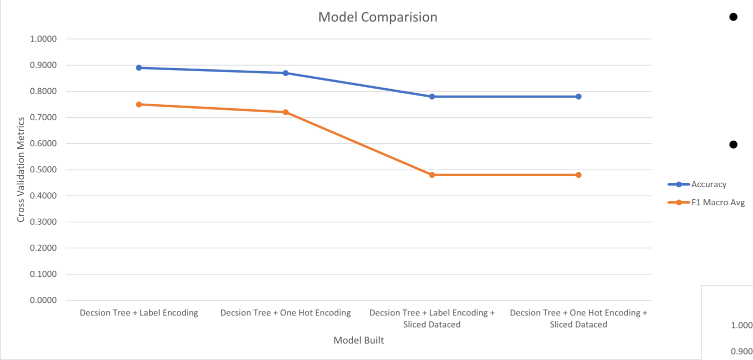


Fig. 7. Hyper Parameter Tuning

E. Naive Bayes - Bernoulli

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

- Naive Bayes - Bernoulli + Label Encoding
  Tuned Hyperparameters :
  {'alpha': 1e-05}
  F1_Macro : 0.5865106692869888
- Naive Bayes - Bernoulli + One Hot Encoding
  Tuned Hyperparameters :
  {'alpha': 1}
  F1_Macro : 0.6406165554349739
- Naive Bayes - Bernoulli + Label Encoding + Sliced Data
  Tuned Hyperparameters :
  {'alpha': 1e-05}
  F1_Macro : 0.4703104591580428
- Naive Bayes - Bernoulli + One Hot Encoding + Sliced Data
  Tuned Hyperparameters :
  {'alpha': 100}
  F1_Macro : 0.4712779129997885

F. Logistic Regression

In statistics, the logistic model (or logit model) is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables. In regression analysis, logistic regression[1] (or logit regression) is estimating the parameters of a logistic model (the coefficients in the linear combination).

- Logistic Regression + Label Encoding
  Tuned Hyperparameters :
  {'solver': 'newton-cg'}
  F1_Macro : 0.7248510662537384
- Logistic Regression + One Hot Encoding
  Tuned Hyperparameters :
  {'solver': 'newton-cg'}
  F1_Macro : 0.7226224779401325
- Logistic Regression + Label Encoding + Sliced Data
  Tuned Hyperparameters :
  {'solver': 'newton-cg'}
  F1_Macro : 0.4703104591580428
- Logistic Regression + One Hot Encoding + Sliced Data
  Tuned Hyperparameters :
  {'solver': 'newton-cg'}
  F1_Macro : 0.4707252420988038

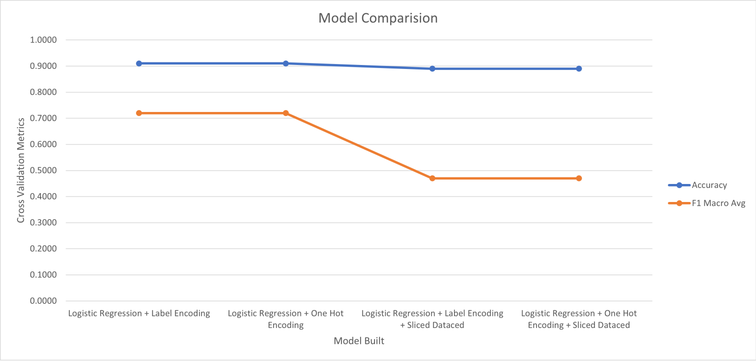


Fig. 8. Hyper Parameter Tuning

G. Observation

The given data is imbalanced and the accuracy for a given class is not the accurate parameter. F1 score would provide a better metric for comparison.All the models were tuned to attain the model with best F1_Macro score.

We have observed the highest accuracy for Logistic regression. The best models with F1 score hyper-tuned K Nearest Neighbors and Decision Trees.

IV. Regression - Bug2

A. Data Pre-Processing

StandardScaler follows Standard Normal Distribution wherein, it makes mean = 0 and scales the data to unit variance.

B. K Nearest Neighbor Regression

Regression model based on K Nearest Neighbors.The target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

Root mean square error (RMSE) is considered as the metric for evaluation.

Tuned Hyper-parameters :
{'n_neighbors': 6, 'p': 2, 'weights': 'distance'}

After hypertuning RMSE value for k=6 is: 9.331563506639315

C. Decision Tree Regression

Regression model based on Decision Trees. RMSE value for Decision Tree is 9.954203689533083

D. Random Forest Regression

Regression model based on Random Forests. RMSE value for Decision Tree is 9.714185546109052

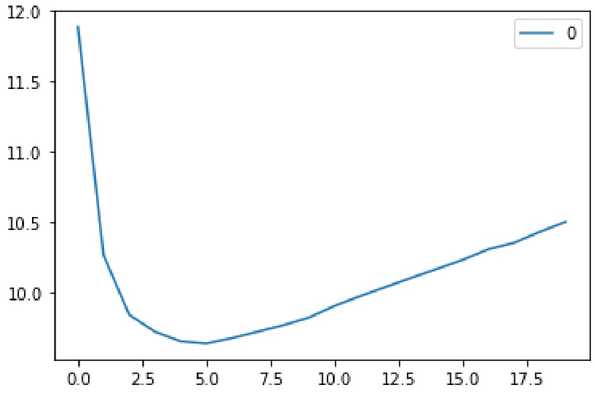


Fig. 9. Root mean square error vs K

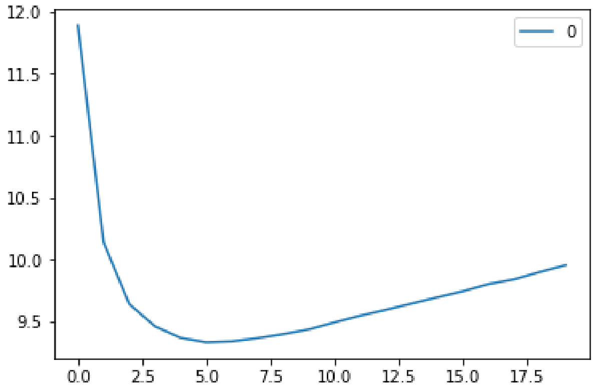


Fig. 10. Root mean square error vs K

E. Neural Networks Regression

Regression model based on Neural Networks. RMSE value for Decision Tree is 14.01418554610932

F. Adaboost Regression

Regression model based on Adaboost. RMSE value for Decision Tree is 38.454238554610319

G. Support Vector Regression

Regression model based on Support Vector Model. RMSE value for Decision Tree is 22.688435408561634

H. Observation

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 to spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit.

One way to assess how well a regression model fits a dataset is to calculate the root mean square error, which is a metric that tells us the average distance between the predicted values from the model and the actual values in the dataset.

The below graph can be plot based on the RMSE for the models. HyperTuning is performed with the below metrics for all the models to attain the best accuracy.

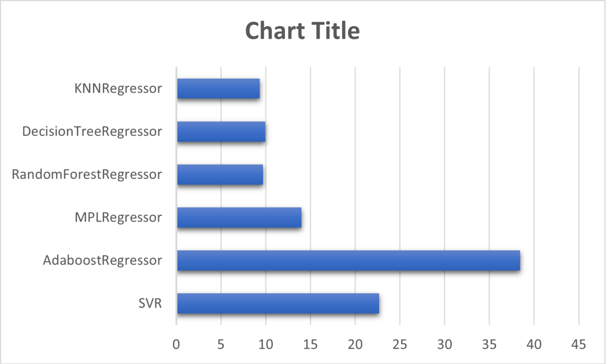


Fig. 11. Root mean square error vs K

References

[1] S. Russel, B. Noble, and P.Norvig, ‘Artificial Intelligence A Modern Approach - Fourth Edition’