DM Assignment Report

March 13, 2022

1 Problem Introduction

Build a binary classification model to predict if a person has `chronic kidney disease(ckd)` or he does not have that disease.

The dataset has

- 400 total datapoints
- 11 numerical features
- 13 categorical/nominal features
- 1 Target variable represented by the feature name `class`

2 Thought Process

We have a dataset of 400 datapoints and we need to build a binary classification model for a target called `class`. Here are the steps we intend to perform to tackle the problem at hand.

- Understand the underlying distribution of different features and get a descriptive summary of the dataset
- Perform cleanup of data
- Engineer some features which can give good performance in classification
- Build several ML Models with different estimators
- Evaluate the model holistically
- Explain future scope and what could be improved upon.

3 Data preprocessing/ Descriptive Summarization

Here are a few things which we observed in the dataset

- There are several missing values in the dataset for both categorical and numerical variables.
- There are several instances where the values in some columns have `\t` tab characters. We have removed them as a part of the data cleaning step.
- Features are not normally distributed and there are several potential outliers in the dataset.
- Numerical features have skewed distributions which are multimodal in nature.
- Categorical features have class imbalances in them

How did we handle them?

- Impute numerical columns using knn imputation
- Impute categorical columns using `most_frequent` observation.

Why imputation? Why not drop the features/rows altogether?

The dataset at hand has only 400 rows, dropping it any further would mean much less data than what we already have.

Also, we don't have too many features to worry about curse of dimensionality, so we need not drop the columns as well.

4 Feature Engineering

Blood Pressure / Age

We can construct a feature for blood pressure per unit age of the person. This will be an informative metric as far as the health of the individual is concerned i.e. to understand whether based on age the BP is high or low.

hemoglobin / rbcc

The proportion of hemoglobin to that of red blood cell counts is one more key information that can tell about the resistance power of an individual. Hemoglobin with iron must be high and also red blood cells count must also be high. This ratio will tell us the relative proportion of each of them in an individual.

5 Model Building

There is a plethora of models to choose from for the usecase at hand i.e. `binary classification` problem. However, we will restrict ourselves to the models which we have studied in the lectures. For being ambitious, we will also fit a model which is not covered in lecture but a natural extension of one of the models taught in the lectures.

1. Logistic Regression - Generally, this is the first choice of modelling any problem whichi nvolves classification. We can get a good baseline using this method. It is a high bias, low variance method.

2. Decision Tree - This models the problem as a set of if-else rules which partitions the space orthogonally into parallelopipeds in the n-feature hyperspace. These are easy to build and have high variance.

3. Random Forests - A natural extension of decision trees which employ the power of ensembling. They build a lot of weak classifiers and combine them to make a strong classification model.

These three models are available in `sklearn.linear_model`, `sklearn.tree` and `sklearn.ensemble` packages respectively which we use for building three varieties of models.

GridSearchCV

Apart from this, we also performed extensive hyperparameter tuning using the `GridSearchCV` method. What it does is it takes the cartesian product of all the hyperparameters and with each combination, performs k-fold (default k = 5) cross validation and saves the best model in `best_estimator_` instance variable of this GSCV object. This also helped us get the best hyper-parameters for the model of our choice.

6 Model Evaluation

Once we have our model, we need a way to tell how good it's performance is. This is what we refer to as model evaluation and there are some very robust metrics which we can use to assess our model.

1. Accuracy - Fraction of correct predictions overall.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2. Recall - How many examples of the positive class was the model able to identify correctly

$$Recall = \frac{TP}{TP + FN}$$

3. Precision - Out of all predicted positives, in how many of them was the model actually correct?

$$Precision = \frac{TP}{TP + FP}$$

4. F1-Score- To strike the goldilocks' balance between precision and recall.

$$F_1 Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

When the classes in target are balanced, accuracy is as good a metric as any but as in our example where we have imbalanced datasets we are interested more in being always correct when identifying `CKD` as opposed to misidentifying it i.e. if we are able to identify all `CKD` patients to be actually `CKD` patients, that is more important than identifying some `non-CKD` person to have `CKD` because in latter, we can confirm and let them go away later on. However the risk is high when a `CKD` person is identified to be `non-CKD`.

This means we need to find out precision and recall and strike a balance between them (Well, recall should be more but having good precision is also important).

With our models, we observed the following performance on train validation and test sets respectively

Accuracy Chart

	Train	Valid	Test
Logistic Reg	98.93	100	98.33
Decision Tree	100	93.33	95
Random Forest	100	100	98.33

F1 Chart

	Train	Valid	Test
Logistic Reg	98.85	100	98.22
Decision Tree	100	93.06	94.67
Random Forest	100	100	98.22

This is an example of baseline being pretty strong to break so to speak. But I will go with Random Forest over logistic Regression. Why? Because if we look at the confusion matrix of test set for these two, we get a 100% recall for CKD class in Random Forest whereas that isn't true for Logistic Regression.

I really care a lot about identifying all the `CKD` patients with a higher success ratio as opposed to `not CKD` ones. That's the logic behind this decision.

7 Future Scope/ Refinement opportunities

1. The dataset is an imbalanced dataset. We can introduce synthetic oversampling techniques such as SMOTE / Undersampling/ Oversampling to balance the dataset and then create a model.
2. The number of datapoints is pretty low and amongst them, we can see that the number of missing values is high. We have used knn imputation currently, but we could also use mean/median/other imputation techniques and compare their performance.
3. Use of neural networks and deep learning (Multi-Layer Perceptron) models for the analysis of this problem.
4. Domain understanding for even better feature engineering to milk the predictive performance of the model even further.