Project Proposal: Final Report

Predicting Chronic Kidney Disorder (CKD) & model creation.

The term “chronic kidney disease” means lasting damage to the kidneys that can get worse over time. If the damage is very bad, your kidneys may stop working. This is called kidney failure, or end-stage renal disease (ESRD). If your kidneys fail, you will need dialysis or a kidney transplant in order to live.

Anyone can get CKD. Some people are more at risk than others. Some things that increase your risk for CKD include:

- Diabetes.
- High blood pressure (hypertension).
- Heart disease.
- Having a family member with kidney disease.
- Being over 60 years old.

What are the symptoms of kidney failure?

You may notice one or more of the following symptoms if your kidneys are beginning to fail:

- Muscle cramps.
- Nausea and vomiting.
- Not feeling hungry.
- Swelling in your feet and ankles.
- Too much urine (pee) or not enough urine.
- Trouble catching your breath.
- Trouble sleeping.
- Itching

Complications of CKD

Your kidneys help your whole-body work properly. When you have CKD, you can also have problems with how the rest of your body is working. Some of the common complications of CKD include anemia, bone disease, heart disease, high potassium, high calcium and fluid buildup.

Introduction to project:

For many people, the only way to know if you have kidney disease is to get your kidneys checked with blood and urine tests.Dataset consist of testing that is done over 2 months of period. Model is created to predict CKD from testing and symptoms collected in the dataset. We will follow the data-science modelling steps,

Initially data-cleaning needs to be performed so that all the missing value’s marked by “?” should be replaced with NaN which is recognised by Pandas dataframe.

Next we will perform some Exploratory data analysis (EDA), to get knowledge about the features (columns) present in the data set, based on which we will consider which features to include in our modelling process.

Perform PCA analysis on the numerical attributes of the dataset, to find how many attributes are relevant, to class prediction columns.

Replacing the null rows with suitable values, delete rows which consist of all null values if any.

Correlation analysis, highly correlated variables to be eliminated, bias is included due to highly correlated variables.

Creating a classification model to predict ckd based on the selected features.

Dataset Description:

Dataset which is being used is downloaded from UCI machine learning repository. Where it was submitted by Dr. P. Soundarapandian (M.D, D.M) (Senior Consultant Nephrologist), Apollo Hospitals, Managiri, TN, India. Created by L. Jerlin Rubini (Research Scholar) Alagappa University

Dataset consist of null values denoted by '?'. It has 400 observations of which (250 predict CKD, 150 predict NOTCKD)

Dataset has 24 attributes and 1 class = 25(columns) [11 numerical,14 nominal]

Following are the list of feature attributes present in the dataset.

1.Age(numerical)age in years.
2.Blood Pressure(numerical) bp in mm/Hg.
3.Specific Gravity(nominal) sg - (1.005,1.010,1.015,1.020,1.025) .
4.Albumin(nominal) al - (0,1,2,3,4,5).
5.Sugar(nominal) su - (0,1,2,3,4,5).
6.Red Blood Cells(nominal)rbc - (normal,abnormal).
7.Pus Cell (nominal) pc - (normal,abnormal)
8.Pus Cell clumps(nominal)pcc - (present,notpresent)
9.Bacteria(nominal) ba - (present,notpresent)
10.Blood Glucose Random(numerical) bgr in mgs/dl
11.Blood Urea(numerical) bu in mgs/dl
12.Serum Creatinine(numerical) sc in mgs/dl
13.Sodium(numerical) sod in mEq/L
14.Potassium(numerical) pot in mEq/L
15.Hemoglobin(numerical) hemo in gms
16.Packed Cell Volume(numerical)
17.White Blood Cell Count(numerical) wc in cells/cumm
18.Red Blood Cell Count(numerical) rc in millions/cmm
19.Hypertension(nominal) htn - (yes,no)
20.Diabetes Mellitus(nominal) dm - (yes,no)
21.Coronary Artery Disease(nominal) cad - (yes,no)
22.Appetite(nominal) appet - (good,poor)
23.Pedal Edema(nominal) pe - (yes,no)
24.Anemia(nominal) ane - (yes,no)
25.Class (nominal) class - (ckd, notckd)

EDA and Data Cleaning :

- In the dataset we observe that 11 features are numerical and the rest of features are categorical. In initial reading of data from csv file "?" is not recognized by pandas as null value so we need to replace it with a np.nan value which would be recognized by pandas. we can see code implementing that in code block [3] of the ipyn notebook.
- Code block [4] removes the ' ' from column names which was read by pandas .
  Due to the '' present in front of the columns names and read all the features columns as object data type, instead of being all column numerical ,sklearn's learning model don't recognize object data type ,code block [7] shows that, so we convert all the feature space to float64 type which will be correctly predicted by the sklearn, learning model.
- Code block [9] forces the categorical features space to representative numerical code. where the sample test predicting ckd is coded as 1 & not_ckd as 0.
- Handling Null values: Code-Block [11] shows in detail statistics of all feature space along with count of all null values in the features space.
- Code block [13] arranges all feature columns in first numerical values and then categorical value so feature space is organized.
- Code block [15] is where we are replacing all the null values by the particular feature space mean value.In case we want to change our strategy for handling null values we can change the mean function with median function. Code Block [16] to [20] is used to convert categorical features into representative numerical code. For that we are using pandas "get_dummies" method . So in code block [19] we can see results of all steps above

We get a clean data with none null values which can be used for further steps
Supervised data: To make our data supervised we need to put the variable to be predicted to last column. Code block [20] does that. it pops the class column to the last column from the middle.

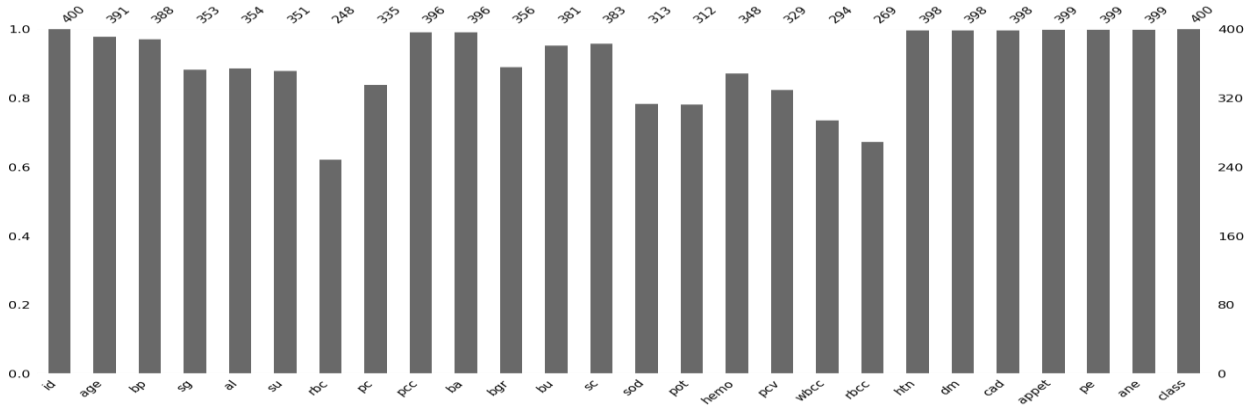


figure showing distribution of null value in the feature space

Visualisation of supervised data

class imbalance:

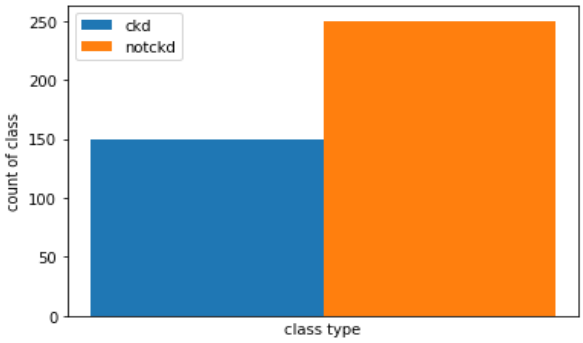
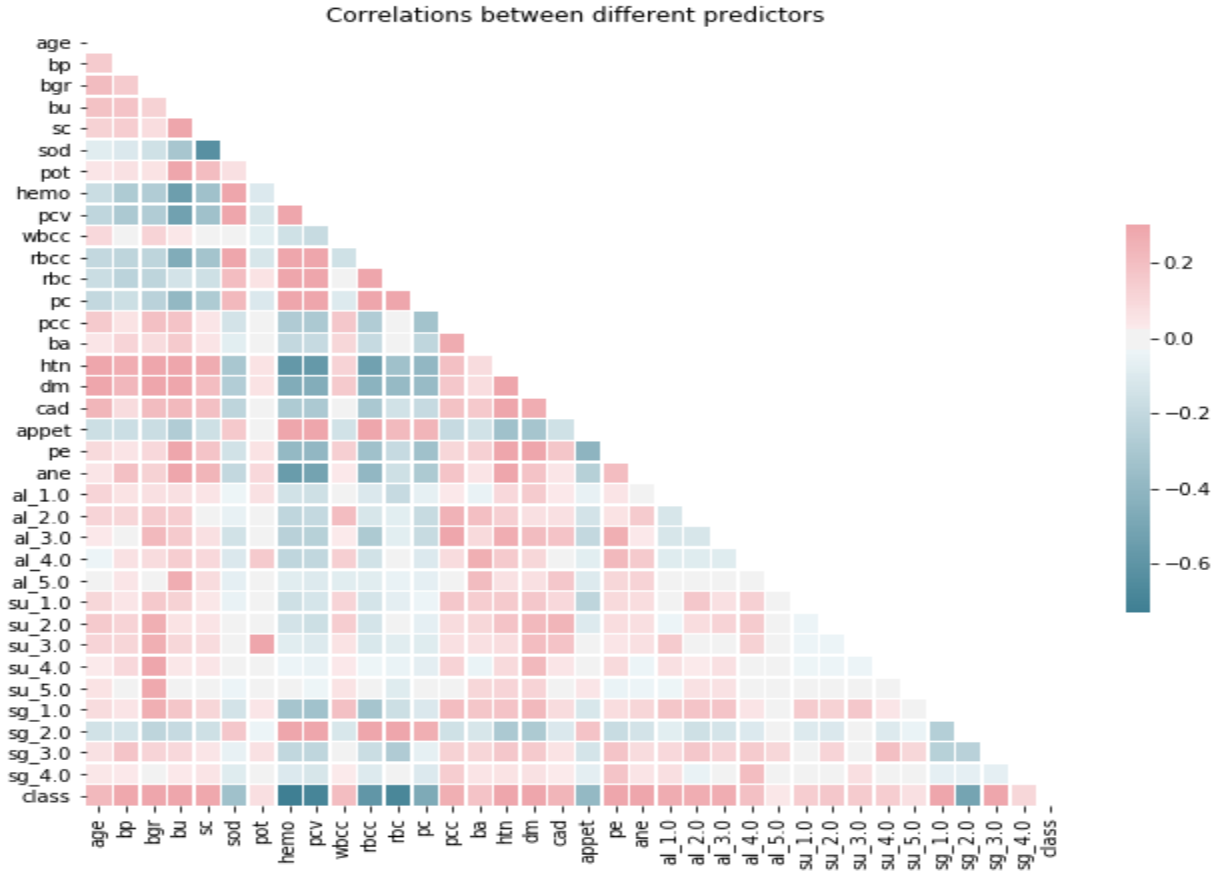


figure showing distribution of prediction class in the sample space. In the classification task, class imbalance of the prediction class can cause the model to be biased.

Class imbalance, i.e the ratio of classes for which predictions need to be made. The figure below shows the class barplot count of the total 400 records present in the dataset. which look skewed, due to this error may creep in the final model, hence resampling needs to be done in final model creation.

Code block [23]-[26] shows the hist plot of all the numerical feature space with their class variable.

correlation(heatmap):



Above correlation heatmap plot shows how strong positive or negative correlation are present between different features of the supervised dataset. Particularly no feature space has high positive or negative correlation . which will cause the model to be biased in prediction task .

Addressing class imbalance:

sklearn library provides several methods to address class imbalance problems.Below i have listed the techniques that can be used.

1. downsampling of the majority class
2. upsampling of the minority class
3. most-frequent sampling method
4. Synthetic Minority Oversampling Technique

We have checked the performance of various techniques on prediction tasks using logistic regression with all the methods mentioned above. We have calculated model performance for all the techniques using sklearn's metrics library . Performance of upsampling and SMOTE technique has been showing better performance than the rest so , In the final prediction task we can use any of those techniques. But we will use resampling techniques.

Performance metric of most-frequent method

Accuracy Score-0.9833333333333333
F1 Score - 0.9863013698630138
Recall Score -0.972972972972973

Performance metric of upsampling method
Accuracy Score-0.98
F1 Score - 0.9833333333333333
Recall Score - 0.9672131147540983

Performance metric of undersampling method

Accuracy Score-0.98
F1 Score - 0.9833333333333333
Recall Score - 0.9672131147540983

Performance metric of most-frequent method

Accuracy Score-0.98
F1 Score - 0.9833333333333333
Recall Score - 0.9672131147540983

PCA Analysis:

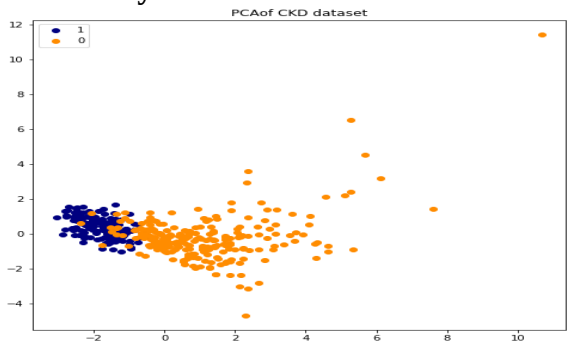


Figure above shows the scatter plot of pca plot when numerical features were converged into two dimension space, distribution shows clearly that we can create a model classifying ckd to not_ckd sample as a single line can be drawn to separate both classes.

But it is not so straightforward as the explained variance plot shows quite linear dependence to the number of feature space obtained from pca algorithm.

So it won't be easy to use pca converted feature space in our final modelling process.

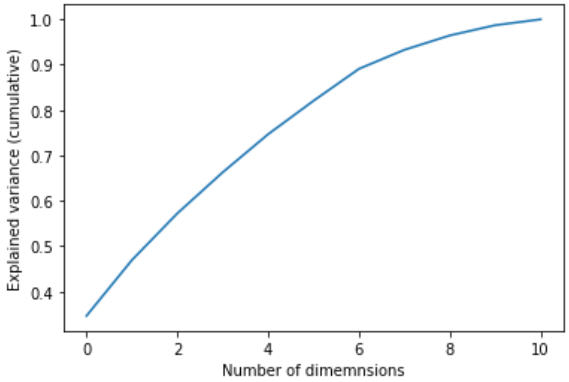
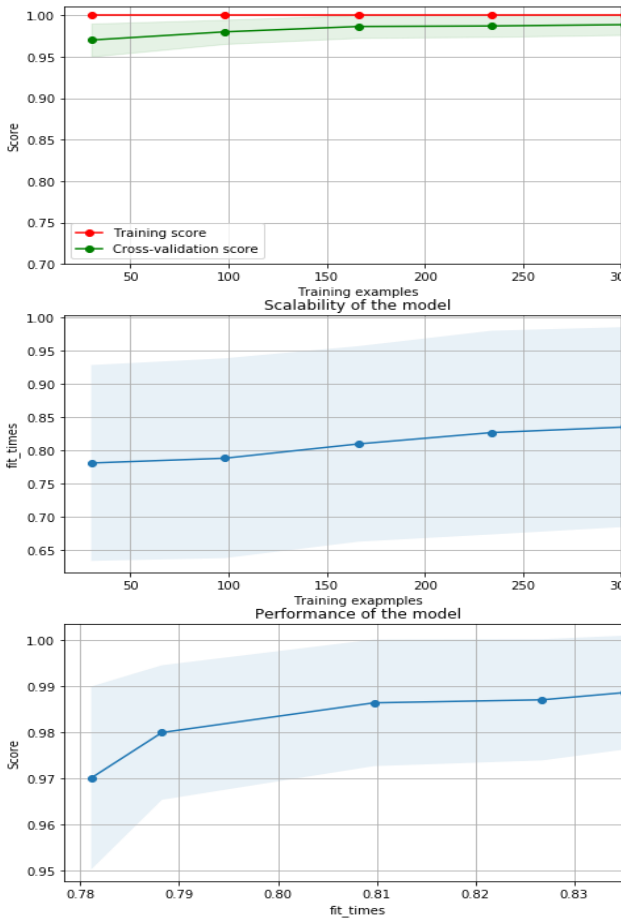


Figure above shows no sharp decline in the explained variance value to the number of dimensions, where it's hard to pick a point where a few numbers of dimensions can be used to explain a substantial amount of variance in feature space. So we won't use the pca converted dimension in our final model.

Validation and Learning Curves



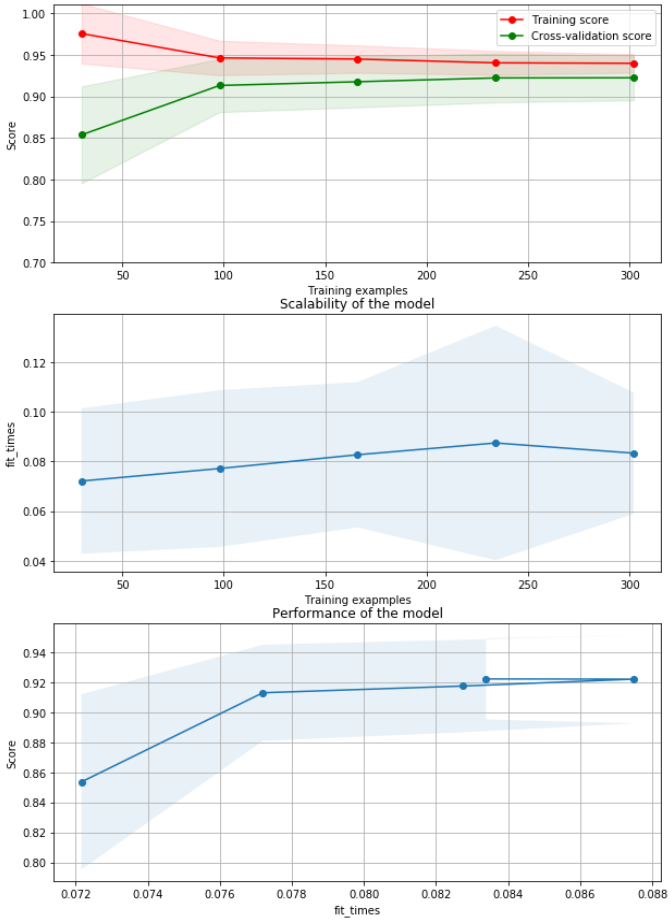
Learning curves along with Cross-validation curves shows how well training data performs on the model created .

Above two plots shows plot of learning curve along with the cross validation curve.Learning curve shows performance on test dataset and cross validation line is showing how accuracy score behaves when it is used for prediction on cross validated data.

For a better model learning curve and cross validated curve should converge to a lowest accuracy score, along with cross validation score should increase as training progresses towards the end .

Learning curve should not decrease to a very low value as training progresses to the end .

from the analysis, of above two curves and others plot , can be find in the ipy notebook we



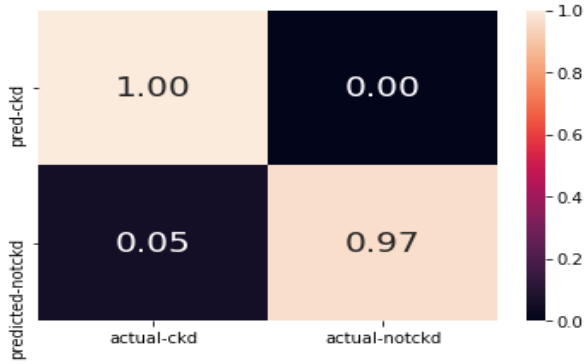
can see that plot for random forest and logistic regression show's a stable prediction characteristics on our supervised dataset.

We can see the prediction metrics of the following two classification algorithm to finalize a prediction classification model for our ckd prediction task

Model performance of Logistic Regression

class	precision	recall	f1-score	support
0	0.95	1.00	0.97	39
1	1.00	0.97	0.98	61
accuracy	-	-	0.98	100
macro -avg	0.98	0.98	0.98	100
weighted-avg	0.98	0.98	0.98	100

Chart above shows a classification report of the logistic regression algorithm performed on the dataset. we have incorporated a upsampling of the training data in the model prediction task for the modeling procedure



Confusion matrix shows how good a logistic regression model is in predicting the class of ckd and not_ckd. The other two boxes of prediction are the number of mis predictions performed by model, value annotated in the box is in terms of percentages.

Logistic regression does a good job of predicting ckd in a sample with 100% accuracy but in the not_ckd prediction tak it does a 5% mispredicting it as not_ckd while it is ckd predicted sample

Model Performance of Random Forest Classification

class	precision	recall	f1-score	support
0	1.00	1.00	1.00	39
1	1.00	1.00	1.00	61
accuracy	-	-	1.00	100
macro -avg	1.00	1.00	1.00	100
weighted-avg	1.00	1.00	1.00	100

Chart above shows a classification report of a random forest classifier performed on a dataset .

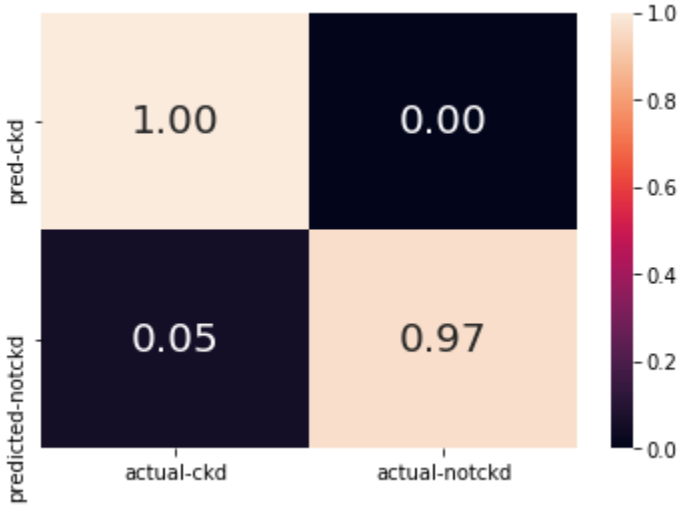
Classification report shows a value of all 100 % correct metrics values. shows perfect prediction results .But if new data arrives, the model may not have such prediction results. so we need to find a way to avoid model overfitting on future data.

Confusion matrix shows how perfectly random forest is doing the task of classification testing samples into ckd and not_ckd.

So it predicts 100% correct ckd and 100% correct not_ckd values .

Ensemble of RandomForest and logistic Regression

Finally we used sklearn's VotingClassifier method to create an ensemble of our above two models.Confusion matrix for the prediction task on test data set is shown below



Here we see results similar to logistic regression, hence the perfect prediction accuracy is overcome by logistic regression classifier. Hence if random forest was doing overfitting on test data then the ensemble of above to model would help us in avoiding this problem of overfitting in future

Conclusion: Logistic regression alone could have performed well in this task of predicting ckd on sample test but, creating an ensemble of best performing methods avoids models to avoid overfitting and underfitting on the future task of prediction.