**MAKERERE**  **UNIVERSITY**

**COLLEGE OF COMPUTING AND INFORMATION SCIENCES**

**(YEAR II) RECESS TERM**

**CONCEPT PAPER FOR: KIDNEY\_DISEASE PREDICTION**

**PROJECT MEMBERS**

|  |  |  |
| --- | --- | --- |
| **NAME** | **REGISTRATION NUMBER** | **STUDENT NUMBER** |
| Kabahinda Rovence | 17/U/4261/PS | 217003233 |
| Meleby Nyika Jino | 17/x/18244/PS | 217019537 |
| Arikosi Michael Okurut | 17/U/2900/PS | 217004064 |
| Mwanje Mike | 17/U/6856/PS | 217004074 |

**PROJECT COORDINATOR**: MR. KAMULEGEYA GRACE

**PROBLEM STATEMENT: KIDNEY\_DISEASE PREDICTION**

The dataset kidney\_disease.csv is to help us predict whether a patient has the chronic kidney disease or not basing on the health records.

**BACKGROUND**

The Kidney\_disease data set is semi-structured and multivariate, taken for a two month period with 400 rows and 25 real features (14 nominal and 11 numeric) along with some missing values. It was provided by Dr.P.Soundarapandian.M.D.,D.M (Senior Consultant Nephrologist) from Apollo Hospitals, Tamilnadu, India. A detailed description of the features is given below:

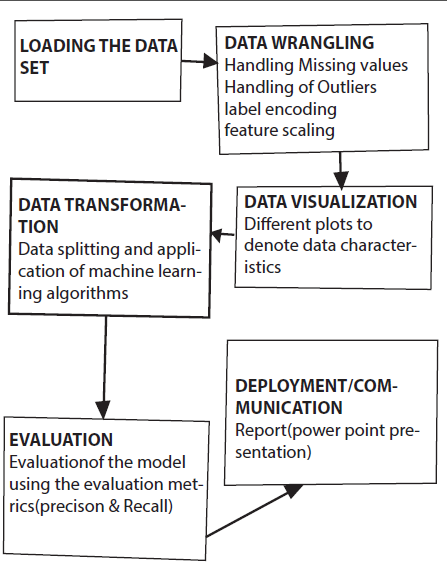
**Description of the features**

|  |  |  |
| --- | --- | --- |
| **FEATURE** | **LABEL** | **DESCRIPTION** |
| Age | age | Age of the patient in years(numeric) |
| Blood Pressure | pb | Blood pressure in mm/Hg(numeric) |
| Specific Gravity | sg | is a measure of the concentration of solutes in the urine(numeric) |
| Albumin | al | Albumin levels of patient(nominal) |
| Sugar | su | Sugar levels of patient(nominal) |
| Red blood cells | rbc | State of red blood cells(nominal) |
| Pus cell | pc | State of pus cell(nominal) |
| Pus cell clumps | pcc | Presence or absence of pus cell clumps(nominal) |
| Bacteria | ba | Presence or absence of bacteria(nominal) |
| Blood glucose random | bgr | Blood glucose level in milligrams/decalitre(numerical) |
| Blood urea | bu | Blood urea level in milligrams/decalitre(numerical) |
| Serum creatinine | sc | serum level in mgs/dl(numerical) |
| Sodium | sod | sodium level in mEq/L(numerical) |
| Potassium | pot | Potassium level in mEq/L(numeric) |
| Hemoglobin | hemo | Hemo in gms(numeric) |
| Packed cell volume | pcv | is a measurement of the proportion of blood that is made up of cells (numeric) |
| White blood cell count | wc | Number of white blood cells per cubic mm(numeric) |
| Red blood cell count | rc | Number of red blood cells in millions per cubic mm(numeric) |
| Hypertension | htn | Presence of hypertension in the patient(nominal) |
| Diabetes mellitus | dm | Presence or absence of high blood sugars in patient(nominal) |
| Coronary artery disease | cad | Presence or absence of coronary artery disease in patient(nominal) |
| Appetite | appet | Good or poor appetite(nominal) |
| Pedal edema | pe | Accumulation of fluid in the feet and lower legs(nominal) |
| Anemia | ane | Presence or absence of anaemia(nominal) |
| Class | class | Chronic kidney disease(ckd) or not chronic kidney disease(notckd)(nominal) |

**DATA ANALYTICS APPROACH (DATA PIPELINE)**

*Data pipeline* allows one to transform data from one representation to another through a series of steps as described below:

**Illustration**



**Loading the dataset:**

The data set is loaded into the jupyter notebook using python pandas and numpy libraries.

**Data wrangling/ cleaning:**

Under this step there a series of activities that take place according to one’s data set. These include;

* **Label encoding**: Incase the features are a mixture of strings and numeric data, we do label encoding to have the string data converted to numerical values for easy manipulation.
* **Missing values**: if there are values missing in the data set they are addressed under data cleaning using the different approaches which include interpolation, backward fill, forward fill, replacing the missing values with a zero, among others.
* **Feature scaling**: This is done by eliminating outliers thereby increasing correlation hence increasing the accuracy of the data.

**Visualization:**

This is placing data into a visual context so that patterns, trends and correlations that might not otherwise be detected can be exposed.

In python there are multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has an excellent library for all this. The following are a few popular plotting libraries that we intend to use during our data visualization:

* [**Matplotlib:**](https://matplotlib.org/)It is a low-level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code. It is good for creating basic graphs like line charts, bar charts, histograms, scatter plots, heat maps and many more that we shall use to visualize the kidney\_disease dataset.
* [**Pandas Visualization:**](https://pandas.pydata.org/pandas-docs/stable/visualization.html) this provides data structures, such as data frames, and data analysis tools like the visualization tools that we intend to use during our visualization of data.
* [**Seaborn:**](https://seaborn.pydata.org/)This is based on Matplotlib and provides a high-level interface for creating attractive graphs.

**Tranformation of machine learning algorithms:**

**Train /Test split**: The data set is split into training data and test data. From*Sklearn***,**sub-library *model\_selection*, we are to import the *train\_test\_split* so that we can well split to training and test sets. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. This data is then fed into the different machine learning algorithms (K-nearest neighbour, k-means, logistic regression, random forest, among others). The respective performance results are obtained from which the best performance algorithm is chosen and a model is built.

**Evaluation:**

When a model is built for a classification problem everyone always wants to look at the accuracy of that model as the number of correct predictions from all predictions made.

The robustness of the model must be evaluated for making predictions on unseen data using **cross-validation and multiple cross-validation** where we shall use classification accuracy (the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage) and average classification accuracy.

Once the model is believed to make robust predictions then a decision whether it is a good enough model to solve our problem shall be made. Classification accuracy alone is typically not enough information to make this decision therefore we shall carry out **Precision** and **Recall** **performance measures** to evaluate the model for a binary classification problem.

## After this we intend to use a clean and unambiguous way to present the prediction results of a classifier by using a Confusion Matrix. For a binary classification problem the table has 2 rows and 2 columns. Across the top is the observed class labels and down the side are the predicted class labels. Each cell contains the number of predictions made by the classifier that fall into that cell.

|  |  |  |
| --- | --- | --- |
|  | Positive | Negative |
| Positive | True Positive | False Positive |
| Negative | False Negative | True Negative |

**DEPLOYMENT AND OR COMMUNICATION**

To transform our code into an asset that can be leveraged and consumed by many different users and roles, for our case we intend to use the following:

* Reports for sharing and disseminating information through a combination of computational results, visualizations, and data narratives

**REFERENCES:**

<https://archive.ics.uci.edu/ml/datasets/chronic_kidney_disease>