Saranya Krishnasami , Student Id - 300321456

Drug Consumption – Decade based Classification problem

The Drug classification problem analysis is commenced with the AS-IS Dataset from the UCI website. The initial set of assumptions are the data is collected from 1885 informants. Data is collected using 5 methodologies including Big Five personality traits (NEO-FFI-R), impulsivity (BIS-11), sensation seeking (ImpSS), and demographic information.

Psychologists have largely agreed that the personality traits of the Five Factor Model (FFM) are the most comprehensive and adaptable system for understanding human individual differences [11]. The FFM comprises Neuroticism (N), Extraversion (E), Openness to Experience (O), Agreeableness (A), and Conscientiousness (C). The data set has information about 18 psychoactive drugs.

Every respondent chose one of the 7 classes for each drug as 'Never Used' [C0], 'Used over a Decade Ago'[C1], 'Used in Last Decade'[C2], 'Used in Last Year'[C3], 'Used in Last Month'[C4], 'Used in Last Week'[C5], and 'Used in Last Day'[C7]. For the purpose classification in this Machine learning algorithm the **6 Drugs** under consideration are Alcohol, VSA, Nicotine, Legal Highs, Ecstasy, Amphet and every drug is subjected to a Binary Classification on the above-mentioned classes separated as "**Users** (1)" and "**Non-Users** (0)". The base classes 'Never Used' [C_0] and 'Used over a Decade Ago'[C_1] grouped as "**Users** (1)" while other classes grouped as "Non-Users".

This analysis activity will perform the machine learning using the statical and probabilistic models for this binary classification problem. The null hypothesis [H_0] we start this classification problem is there is no significant classification between Users and non-users of the respective drugs. The drugs under analysis such as **Alcohol, VSA, Nicotine, Legal Highs, Ecstasy, Amphet** will be binary classified using the models and accuracy for each drug will be calculated and analyzed for this activity.

GitHub Repository Link:

Modelling process:

The activity of model building to classify the drug consumption problem under consideration for this activity consists of the following steps,

- Feature Extraction
- Feature Selection
- Model Building – Decision tree [DT], Random Forest [RF], Support Vector machine [SVM], KNN [K – Nearest Neighbor]
- Evaluation
- Prediction
- Compare with the research paper

Step 1 :Feature Extraction:

Feature extraction is the process of converting raw unprocessed data into numerical values for the easiness of computation. The data set we have started the modelling process is already extracted and all

the features are extracted and normalized. Hence, for this activity there was no extra step done for extraction.

Step 2: Feature Selection:

Feature selection is essential for efficient predictive models. This allows us to filter out the input variables highly dependent on the response variables. In our case we are using ANOVA (Analysis of Variance) F measure to do feature selection with "Filter Based" technique in Scikit learn. In our analysis we have 12 input features, and the selection is based on the variance of the input parameters. With 8 selected features using the SelectK best Algorithm such as Age, Gender, Ethnicity, NScore, AScore, CScore, Impulse and SS.

Step 3: Model building:

Four models are used to classify this binary classification problem with the selected 8 features. The classifiers used are Decision tree, Random Forest, SVM and kNN.

Step 4: Evaluation:

The evaluation metrics used are accuracy of the model, precision, recall. To provide a better measure of accuracy ROC and AUC is used for all the 4 classifiers. The

Step 5: Prediction:

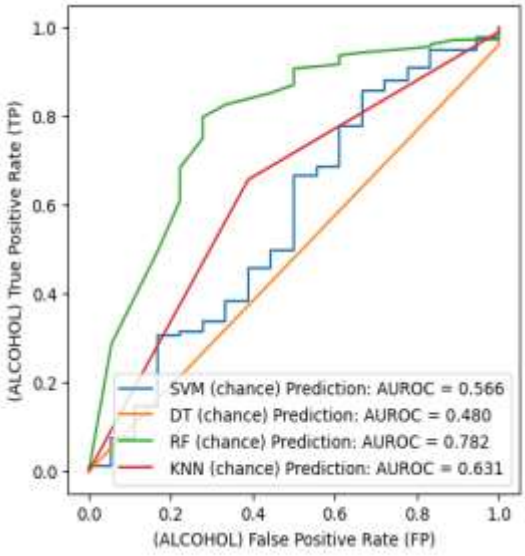
- The **Random Forest classifier is having the highest Recall (Sensitivity) rate for drugs Alcohol, Amphet, Nicotine and VSA**. (Refer Table 1.0)
- While the SVM classifier has a good prediction for Legal H and Ecstacy.

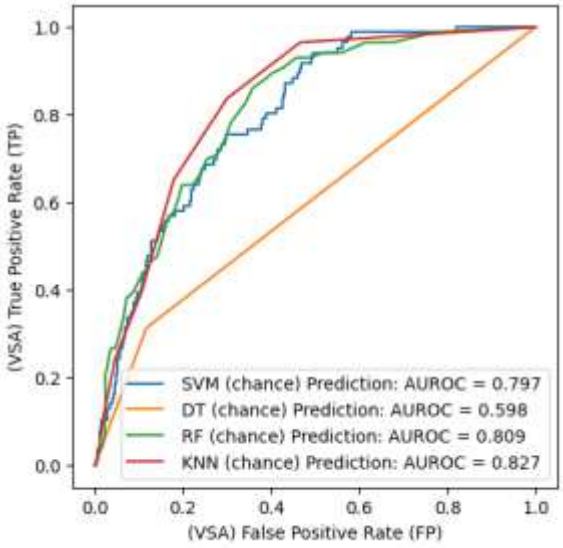
Feature Set : Age, Gender, Ethnicity, Nscore,Ascore,Cscore,Impulse,SS (8 Features)												
	Decision Tree			Random Forest			Support Vector Machine			K Nearest Neighbors		
	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Alcohol	0.935	0.96	0.97	0.971	1	0.97	0.971	1	0.97	0.971	1	0.971
VSA	0.82	0.2	0.32	0.852	0.08	0.39	0.861	0	0	0.853	0	0
Nicotine	0.601	0.69	0.72	0.711	0.8	0.76	0.702	0.86	0.78	0.688	0.76	0.78
LegalH	0.702	0.66	0.64	0.854	0.71	0.73	0.864	0.73	0.77	0.833	0.69	0.75
Amphet	0.651	0.54	0.5	0.691	0.53	0.56	0.702	0.61	0.57	0.707	0.48	0.6
Ecstacy	0.7	0.62	0.63	0.728	0.65	0.67	0.747	0.69	0.69	0.718	0.69	0.69

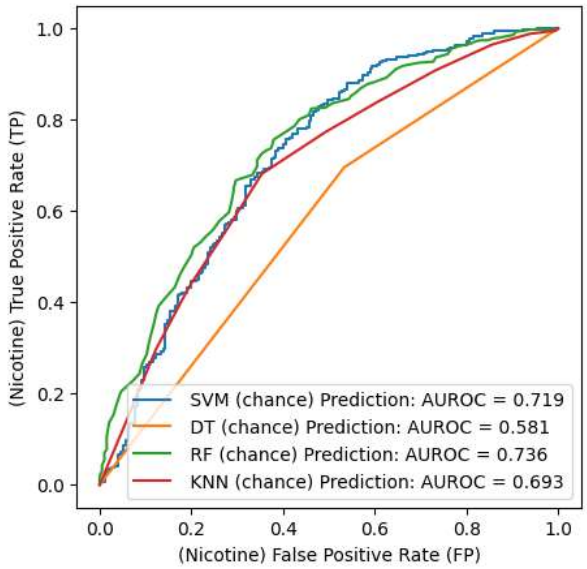
Table 1.0

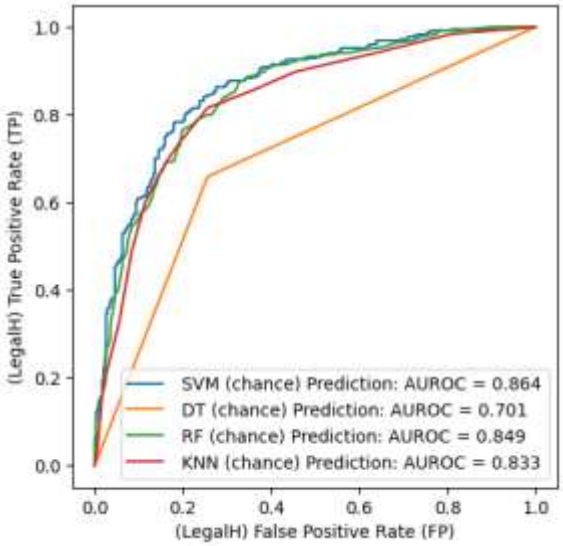
The users are classified as 1 and non-users as 0. The confusion matrix highlights that for certain drugs data is skewed more towards users (Such as Alcohol) and there are less samples for non-users (Such as Semer and VSA) which is reflected in the scores. Alcohol shows the highest accuracy for RF (0.785) and a precision of 0.99 for the True Negatives (Users) while the True positives (non-users) are almost null due to unavailability of data for non-users.

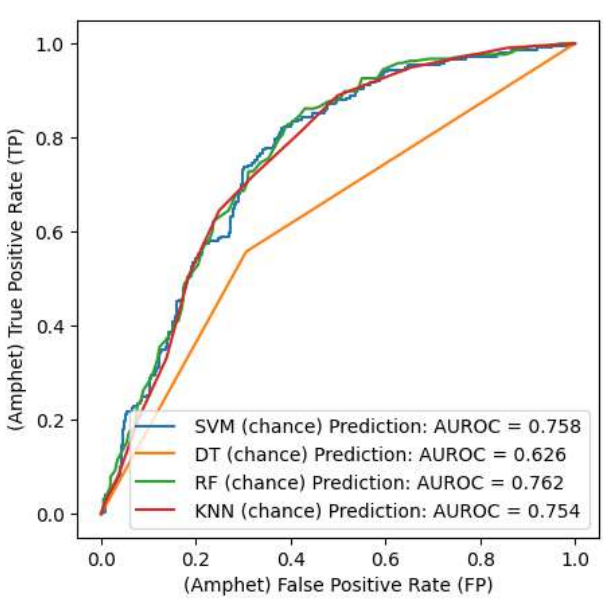
While for the VSA, there are more non-users than users in the sample set the prediction classifier that worked well is RF (0.825) with a sensitivity (Recall) rate (0.99) is high for non-users in RF and Precision (Positive predictive value) is 0.87. Which makes RF a good classifier for the Alcohol and VSA.

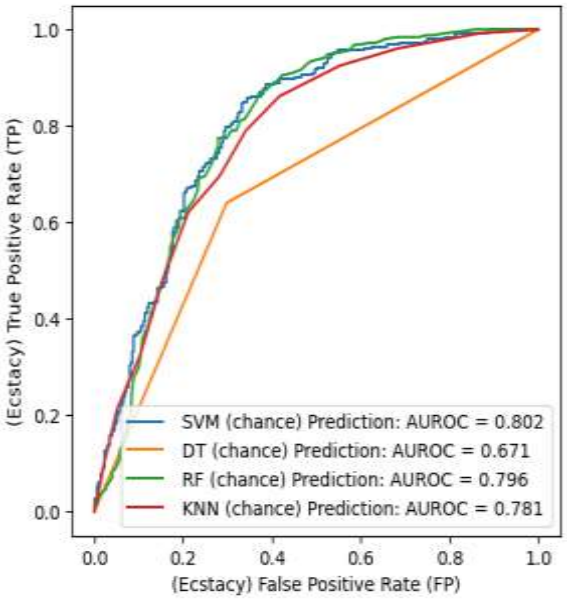












Lessons learnt:

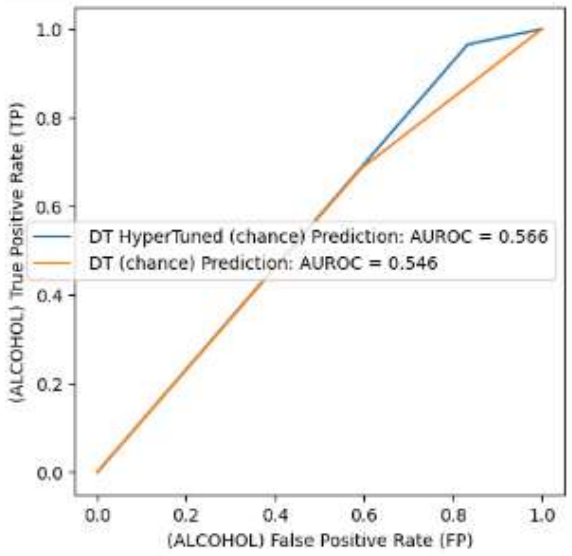
Parallel study with Research Paper:

1) **Feature Selection:** The Research paper has a specific feature selection for every drug to bring out the significant correlation and effect of the input features with the specific drug consumption. For example,
    a. For Drug ecstasy consumption the best classifier is DT based on features age, SS, gender and has sensitivity 76.17% and specificity 77.16% using just 3 features.
    b. For our Model we have considered 8 features selected using ANOVA F measure for the drugs

2) **Hyper Parameter Tuning:** In the research paper for the DT Gini gain, information gain and DKM gain are considered while developing the decision tree. For our activity, Gini and Entropy were used to see is the hyper parameter tuning using Grid search helped to improve the score for Alcohol. The performance was improved slightly for the prediction as denoted in the ROC curve below. (Refer Figure 1.0)

Conclusion:

With this binary classification it is evident that the respondents can be clearly classified as users and non-users and hence we reject the null hypothesis (H_0).




```
Accuracy score (Decision Tree Hyper Tuned):-  0.9421221864951769

Confusion Matrix (Decision Tree Hyper Tuned):-

[[  3  15]
 [ 21 583]]
Decision Tree Hyper Tuned
              precision    recall  f1-score   support

           0       0.12      0.17      0.14        18
           1       0.97      0.97      0.97       604

    accuracy                           0.94       622
   macro avg       0.55      0.57      0.56       622
weighted avg       0.95      0.94      0.95       622
```

Figure 1.0

GitHub Link:

https://github.com/SharuGitHubSpace/CSI-5155-Machine-Learning-Assignment-2.git

Assumptions and Evaluation Methodology:

- The programming is optimized to handle the Cross validation, over sampling and under sampling in one loop with the function Cross validation with parameter configuration *Cross_Validation (X, y, ToDoOversample = 0, ToDoUnderSample = 0, datasetName = '')*
- For efficient programming all models are looped, and Hyper parameters are configured
- The cross_val_score taken in the score of the Accuracy metrics
- The cross validation is done at two level outer and inner folds
- While printing the result the' Drug Dataset CV 'got appended as an error. Kindly apologize for the same and ignore the same.

Summary:

The Best performing drug VSA with an accuracy of 86% is selected for further analysis. The VSA drug has a class imbalance with Non-Users (1654) and Users (230). The bar chart below shows the distribution.

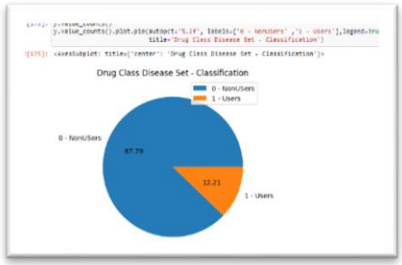
```
]: DataSet_Algorithms_AccuracyTable
```

]:

	SVM	DT	RF	KNN	MLP	GB
Drug Dataset	0.894	0.894	0.884	0.894	0.894	0.894
Drug Dataset - OS	0.73	0.788	0.8677	0.772	0.772	0.894
Drug Dataset - US	0.656	0.751	0.735	0.661	0.719	0.894
Labor CV DS	0.833	0.833	0.833	0.833	0.833	0.833
Heart CV DS	0.766	0.733	0.733	0.7	0.833	0.833

The Drug Classification dataset study

Data Set 1: Dataset D (Base line)

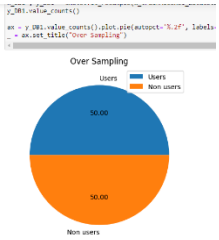


- The VSA drug data is **feature scaled** using **Min Max Normalizer** to normalize the data to improve the model performance
- The VSA drug has a data imbalance
- The data is cross validated with **Simple k – Fold Cross Validation** (k=10). Cross validation improves the over all accuracy of the model by performing 10 folds of validation. With 9 folds as Training and 1 set for Testing.
- **The 9 folds data is again split into 2 folds 6 for training and 3 for validation in the inner fold.** This way the model can prevent overfitting the data.
- **Hyper parameter tuning i**s done within the Cross-validation loop using Gridsearchcv with different parameters for the different classifiers (Refer Picture 1.0)

Observations and Lessons Learnt from Cross Validation:

- Once the cross validation was done the mean accuracy of the model improved from 86 % to 89 % for most of the Models
- Cross validation gave an idea of generalization of the model
- Training time increased with Cross validation taking higher computational time
- Hyper parameter tuning is extensive time-consuming process, and the accuracy was slightly improved with hyper tuning on the models which is a tradeoff

Data Set 2: Dataset DB 1 (Cross validation + SMOTE Over Sampling)



- The VSA drug data is feature scaled using Min Max Normalizer to normalize the data to improve the model performance
- The VSA drug has a data imbalance
- The data is cross validated with simple k – Fold Cross Validation (k=10).
- After the Cross validation the Oversampling is done for the Minority class to balance the data in dataset
- The Users are the Minority class, and they are oversampled using **SMOTE (Synthetic Minority Oversampling Technique).** SMOTE technique introduced synthetic samples of the minority class into the dataset for over sampling the minor class.

- [KNN, Decision Tree, Random Forest, SVM]4 Algorithms are applied to this Data set and the results are as below
- **Method First Cross Validate Then Over sampling:** The whole dataset is split into 10 folds with 9 folds used for training and 1-fold for testing and this is repeated for 10 times. Inside the loop of cross validation after splitting the folds the over sampling is applied only to the training set. This prevents data leakage and better prediction as the model

Best performing Model: Gradient Boost with 89 % Accuracy and Hyper parameters learning rate, max_depth and max_features

Observations from Cross Validation + Over Sampling:

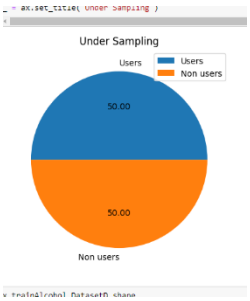
- Over sampling is a non-cost sensitive to learn and implement all learning algorithms
- To achieve the best accuracies Hyper Parameter tuning is done for all the Models (as shown in the picture 1.0). After the cross validation and Over sampling is done the mean accuracy of the models decreased. This is because puts more weight to the small class, makes the model bias to it. *The model will now predict the small class with higher accuracy, but the overall accuracy will decrease*
- **Analysis:** The Drug dataset over sampling rates for SVM dropped significantly from 89% to 73%, while the DT, KNN, MLP dropped noticeably. Random forest performance was not much affected with the accuracy.

Lessons Learnt from Cross Validation + Over Sampling:

- The synthetic generated samples are not representative of the minority class
- The Model is overfitted to the data and over sampling is not suitable for all imbalanced classes

- Hyper parameter tuning was an extensive time-consuming process, and the accuracy was slightly improved with hyper tuning on the models

Data Set 3: Dataset DB 2 (Cross validation + Random Under Sampling)



- The VSA drug data is feature scaled using Min Max Normalizer to normalize the data to improve the model performance
- The VSA drug has a data imbalance
- The data is cross validated with simple k – Fold Cross Validation (k=10).
- After the Cross validation the Under sampling is done for the NON-USERS class which is the Majority class to balance the data in dataset
- The nonusers are the Majority class and they under sampled to match with the user's class

- The Sampler used if the random sampler and the model is trained with hyper parameter tuning for every classifier as mentioned in the picture 1.0

Best performing Model: Gradient Boost with 89 % Accuracy and Hyper parameters learning rate, max_depth and max_features

Observations from Cross Validation + Under Sampling:

- After the cross validation and under sampling was done the mean accuracy of the model further decreased with the Drug class data set. The accuracy of the model denotes the prediction of the positive and negative class, the Recall / precision will give a better clarity to the prediction of the user and non-user class.
- The performance of SVM and KNN reduced drastically during under sampling. This is possible because both are distance-based algorithm and with under sampling the majority class is deleted that may be useful, important, or perhaps critical to fitting a robust decision boundary.

Lessons Learnt from Cross Validation + Under Sampling:

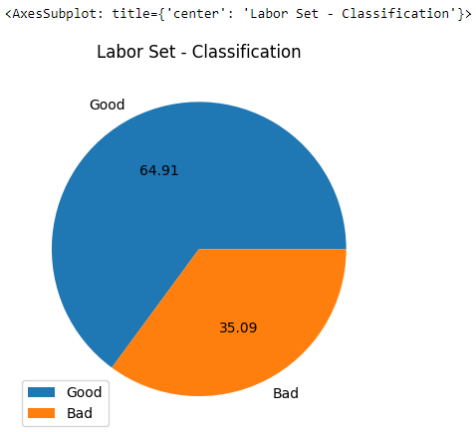
- Under sampling has not improved the overall accuracy of the models however the accuracies are deprecated, and this could be the overall accuracy of the model is the measure of the correct prediction out of all the predictions made. Like over sampling the recall / precision score will give a picture of the prediction of the imbalance class.

Labor Data set Study

Data set Pre-Processing

- The labor data set taken was not cleaned.
- Data preprocessing steps like cleaning Nan values, Null values were done.
- To handling missing values Data imputation techniques like one hot encoding is done using Simple Imputer.
- Columns with more than 20 missing values were dropped due to less significance. The Final data set cleaned and appeared as below. The feature scaling and feature selection is done as a part of binary Classification.

- The target field is Good or bad. The Data is not very imbalanced as shown in the graph below
- Following steps are done to the data set
    - ***Preprocessing***
        - Data cleaning
        - Data Imputation
    - ***Feature Engineering*** – One hot encoding
    - ***Feature Scaling*** – Min Max Normalizer
    - ***Feature Selection*** - The features selected for the analysis are Duration of employment, Wage, Hours of work, Holiday, vacation using ANOVA




```
In [55]: X,y = Get_Labordataset()
Out[55]:
```

	dur	wage1	wage2	cola	hours	holidays	vacation	dntl_ins	empl_hplan
0	1.000000	5.000000	3.971739	0.0	40.000000	11.00000	0.0	1.0	0.0
1	2.000000	4.500000	5.800000	0.0	35.000000	11.00000	1.0	0.0	0.0
2	2.160714	3.803571	3.971739	0.0	38.000000	11.00000	2.0	1.0	1.0
3	3.000000	3.700000	4.000000	1.0	38.039216	11.09434	1.0	1.0	0.0
4	3.000000	4.500000	4.500000	0.0	40.000000	12.00000	0.0	1.0	1.0
5	2.000000	2.000000	2.500000	0.0	35.000000	12.00000	0.0	1.0	0.0
6	3.000000	4.000000	5.000000	1.0	38.039216	12.00000	2.0	2.0	1.0
7	3.000000	6.900000	4.800000	0.0	40.000000	12.00000	1.0	1.0	0.0
8	2.000000	3.000000	7.000000	0.0	38.000000	11.00000	1.0	1.0	0.0
9	1.000000	5.700000	3.971739	0.0	40.000000	11.00000	2.0	0.0	0.0

Best performing Model: All the models predicted the accuracy of 83%

The Assignment expects to produce highest accuracy for the 6 algorithms for this dataset. Since the Data set is Mildly skewed Cross validation + Over sampling and cross validation + under sampling techniques were applied to the labor data set and the predictions were made to the accuracy of the model. It is observed that only cross validation works well this data set. So, the Labor data set is cross validation and taken for observation.

Observations and Lessons Learnt:

- The labor data set is not skewed so the over sampling or under sampling methods were not effective
- This data set is having many missing values and heavy preprocessing is done to the dataset which might have resulted in data loss
- All the models have the same accuracy
- The hyper tuning took a lot of time and the there is not much difference in the accuracies which is an overhead. The hyper parameters of the models were configured to produce the best score and execution time was heavily increased in this process.
- Hyper tuned Gradient boosting took the maximum time for execution

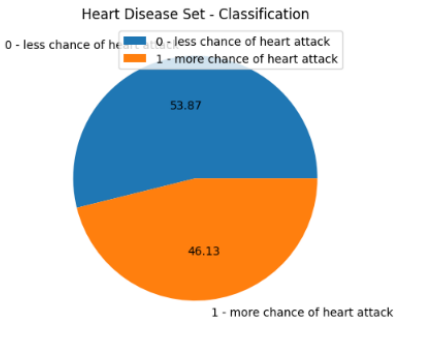
Heart Disease Data set study

The heart disease data set is not having any significant skew and the following steps were done to pre-process the data set,

- Following steps are done to the data set
    - ***Preprocessing***
        - Data cleaning
        - Data Imputation
    - ***Feature Engineering*** – One hot encoding
    - ***Feature Scaling*** – Min Max Normalizer
    - ***Feature Selection*** - The features selected for the analysis are The Features selected for the heart attack set are as below using the **SelectKBest** are the Age, Sex of Patient, RestEcg, thalach, exang, oldpeak, slope and Ca as shown in the figure below. using ANOVA

```
In [152]: y = pd.DataFrame(y)
          y.value_counts().plot.pie(autopct='%.2f', labels=['0 - less chance o
                                    title='Heart Disease Set - Classification'
```

```
Out[152]: <AxesSubplot: title={'center': 'Heart Disease Set - Classification'}
```




```
[153]: X,y = Get_HeartDataset()
       print(X)
```

```
      1    2         7    8         9    10        11   12
0    1.0  0.0  0.458015  0.0  0.016129  0.5  0.333333  0.0
1    0.0  0.0  0.610687  0.0  0.290323  0.0  0.666667  0.0
2    0.0  0.0  0.328244  0.0  0.419355  1.0  0.000000  0.0
3    1.0  0.0  0.786260  0.0  0.225806  0.5  0.333333  0.0
4    1.0  0.0  0.557252  1.0  0.290323  0.5  0.000000  0.0
..   ...  ...       ...  ...       ...  ...       ...  ...
292  1.0  1.0  0.839695  0.0  0.000000  0.0  0.000000  1.0
293  1.0  1.0  0.526718  0.0  0.193548  0.5  0.000000  1.0
294  1.0  1.0  0.450382  1.0  0.258065  0.5  0.000000  1.0
295  0.0  1.0  0.847328  0.0  0.225806  0.0  0.000000  0.0
296  1.0  1.0  0.648855  1.0  0.000000  0.0  0.000000  1.0

[297 rows x 8 columns]
```

Best performing Model: All the models predicted the accuracy of MLP and GB 83%

The Assignment expects to produce highest accuracy for the 6 algorithms for this dataset. Since the Data set is Mildly skewed Cross validation + Over sampling and cross validation + under sampling techniques were applied to the labor data set and the predictions were made to the accuracy of the model. It is observed that only cross validation works well this data set. So, the Labor data set is cross validation and taken for observation.

Observations and Lessons Learnt:

- The heart data set is also not skewed significantly so the over sampling or under sampling methods were not effective
- MLP and Gradient boosting algorithm gave the best prediction with the hyper parameters [activation, alpha, solver, learning rate] and [loss, learning rate, max_depth, max_features] respectively
- The hyper tuning took a lot of time and the there was not much difference in the accuracies which is an overhead. To experiment with the hyper parameters, the execution time was a big hurdle

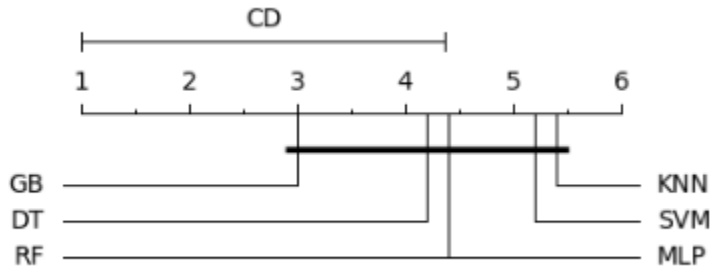
Friedman's Test and Critical Distance calculation:

The critical value for k = 6 (Algorithms) and n = 5 (Datasets) at the α = 0.05 has a critical value = 12.592. The null hypothesis being all the algorithms are performing equally. With the Friedmans statistic value being 5.58 we accept the Null Hypothesis. And all algorithms are performing equally.

The Friedman's Statistic p value obtained: 5.581728799672265

The Average ranks from Friedman's Test: The mean ranking of the algorithms are SVM (5.2), DT (4.2), RF (4.4), KNN (5.4), MLP (4.4), GB (3.0).

Critical difference value 3.3718



Annexures:

Hyper Parameter Tuning for the Models

```
DT__params = {"max_depth": [3, None],
              "max_features": [1, 3, 10],
              "min_samples_split": [1, 3, 10],
              "min_samples_leaf": [1, 3, 10],
              # "bootstrap": [True, False],
              "criterion": ["gini", "entropy"]}

SVC__params = {'C': [0.1, 1, 10, 100, 1000],
               'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
               'kernel': ['rbf','poly', 'sigmoid']}

RF__params = {
               'n_estimators': [200, 700],
               'max_features': ['auto', 'sqrt', 'log2']
                }
k_range = list(range(1, 31))
KNN__Params = dict(n_neighbors=k_range)

MLP__params = {
            'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],
            'activation': ['tanh', 'relu'],
            'solver': ['sgd', 'adam'],
            'alpha': [0.0001, 0.05],
            'learning_rate': ['constant','adaptive'],
                }
GB__params = {
"loss":["deviance"],
"learning_rate": [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2],
#"min_samples_split": np.linspace(0.1, 0.5, 12),
#"min_samples_leaf": np.linspace(0.1, 0.5, 12),
"max_depth":[3,5,8],
"max_features":["log2","sqrt"],
#"criterion": ["friedman_mse",  "mae"],
#"subsample":[0.5, 0.618, 0.8, 0.85, 0.9, 0.95, 1.0],
#"n_estimators":[10]
}
```

Picture 1