Capstone Project Report

A Thesis

Submitted by

Samarth Sarin (R005)
(SAP ID: 70281018005)

Under the Guidance Of
Prof. Siba Panda

in partial fulfillment for the award of the degree of

MTECH
IN
ARTIFICIAL INTELLIGENCE
At

DEPARTMENT OF DATA SCIENCE
MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT AND ENGINEERING, MUMBAI
NOVEMBER, 2019

CERTIFICATE

This is to certify that the thesis entitled “Capstone Project Report” is the bonafide work carried out by Samarth Sarin (SAP ID: 70281018005) M.Tech (Artificial Intelligence), MPSTME (NMIMS), Mumbai, during the III semester of the academic year 2019-20, in partial fulfillment of the requirements for the award of the Degree of Masters of Artificial Intelligence as per the norms prescribed by NMIMS. The project work has been assessed and found to be satisfactory.

Prof. Sarada Samantaray

Associate Dean

NMIMS University

DECLARATION

I, Samarth Sarin, Roll No. R005, M.Tech (Artificial Intelligence), III semester understand that plagiarism is defined as anyone or combination of the following:

1. Uncredited verbatim copying of individual sentences, paragraphs or illustration (such as graphs, diagrams, etc.) from any source, published or unpublished, including the internet.

2. Un-credited improper paraphrasing of pages paragraphs (changing a few words phrases, or rearranging the original sentence order)

3. Credited verbatim copying of a major portion of a paper (or thesis chapter) without clear delineation of who did wrote what. (Source: IEEE, The institute, Dec. 2004)

4. We have made sure that all the ideas, expressions, graphs, diagrams, etc., that are not a result of my work, are properly credited. Long phrases or sentences that had to be used verbatim from published literature have been clearly identified using quotation marks.

5. We affirm that no portion of my work can be considered as plagiarism and we take full responsibility if such a complaint occurs. I understand fully well that the guide of the seminar/ project report may not be in a position to check for the possibility of such incidences of plagiarism in this body of work.

Signature

Name:

Roll No.

Place:

Date:

ACKNOWLEDGEMENT

I owe a great thanks to a great many people who helped and supported me during the making of this project. I thank my guide Prof. Siba Panda who devoted his valuable time for supporting me throughout the project and giving me the opportunity to work on this Project. His wisdom, knowledge, commitment, insight, support and energy motivated me to bring forth my best.

I would also extend my gratefulness to Prof. Sarada Samantaray, Associate Dean, for guiding and correcting various documentations of mine with attention and care. I would also like to thank Mukesh Patel School of Technology, Engineering and Management for this wonderful opportunity.

The project was not possible without the consolidated efforts of my team. It gives me immense pleasure in presenting a part of this project.

Table of Contents

ABSTRACT

Python is a powerful, flexible, open source language that is easy to learn, easy to use and has powerful libraries for data manipulation, analysis and building web interfaces. Every sector of business is being transformed by the modern deluge of data. These spells doom for some, and creates massive opportunity for others. Those who thrive in this environment will do so only by quickly converting data into meaningful business insights and competitive advantage. Now a days' data has increased enormously and opened up ways of analyzing patterns and understanding different things.

All in One Machine Learning app is built with an easy web interface where data and machine learning come together for businesses that are looking to get instant data, instant insights and instant decisions.

Traditionally companies take around 2-3 days to perform basic algorithms and to get insights of the data. It is important for businesses to derive meaningful insights from customer data to make key business decisions and improve customer relationship management.

CHAPTER 1
INTRODUCTION

Machine Learning is one of the trending topics in the industry. Businesses are improving their command over the data with the use of Artificial Intelligence. Machine Learning, Deep Learning and Natural Language Processing are helping businesses to get the insights of the data which might be difficult for a human to retrieve. Most of the companies want to use Machine Learning in their business but lack in talent to actually sit and code the algorithms. Here this tool comes into use to help the businesses in adapting Machine Learning to their problem statement. You can simple upload a CSV dataset and select any algorithm you wish to perform with just a few clicks without actually opening any other tool to code down your problem statement. The tool involves various tasks like all classification algorithms, all Regression algorithms, PCA for dimensionality reduction, feature importance techniques and basic visualizations of the data. It all provides with a report in a word document for all the results and gives few suggestions which the system things might help in improving the model.

1.1 About the Product

1.1.1) More Data More Control

The product takes in the data of the customer in order to make the 360-degree view of the customer, i.e. it analyses the data in order to provide the useful deduction that helps in making useful assumptions about the customer.

1.1.2) More Insights per minute

The purpose of analyzing the customer data is also that it can provide useful insights about the customers that helps in the targeted marketing for the specific customer.

1.1.3) Instant Data, Instant Results

The data is used to make timely decisions as the value of time is indispensable in the modern time analysis.

1.2 Features

1.2.1) All Regression Algorithms

All the regression algorithms like Linear Regression are available for the user to perform on their dataset. Hyperparameter tuning is also available for the user if he wants to tune the algorithm.

1.2.2) All Classification Algorithms

All the classification algorithms like Logistic Regression are available for the user to analyze his dataset along with Hyperparameter tuning of the model.

1.2.3) PCA for Dimensionality Reduction

PCA algorithm is provided as an option for the user if he wants to reduce the dimensionality of the dataset uploaded.

1.2.4) Feature Selection Techniques

Finding out the important features of the dataset and using them for analysis.

1.2.5) Visualization

Basic visualization plot options are given to the user to get the insights of the data.

1.2.6) Report Generation

Detailed report in word document comprising of results and model improvement suggestions.

1.3 Advantages

1.3.1) Faster time to analyze

User doesn't have to code for the particular problem and can perform as much number of algorithms with only few clicks.

1.3.2) Reduced operational Cost

No coding has to be done by the user to perform all the algorithms manually. Now it's all automated.

1.3.3) No Hardware Requirements

There is no need for any heavy hardware requirements as all the computation will be done on the cloud servers with high GPU configurations.

CHAPTER 2
DEVELOPMENT WORKFLOW

2.1 Tool Workflow Steps:

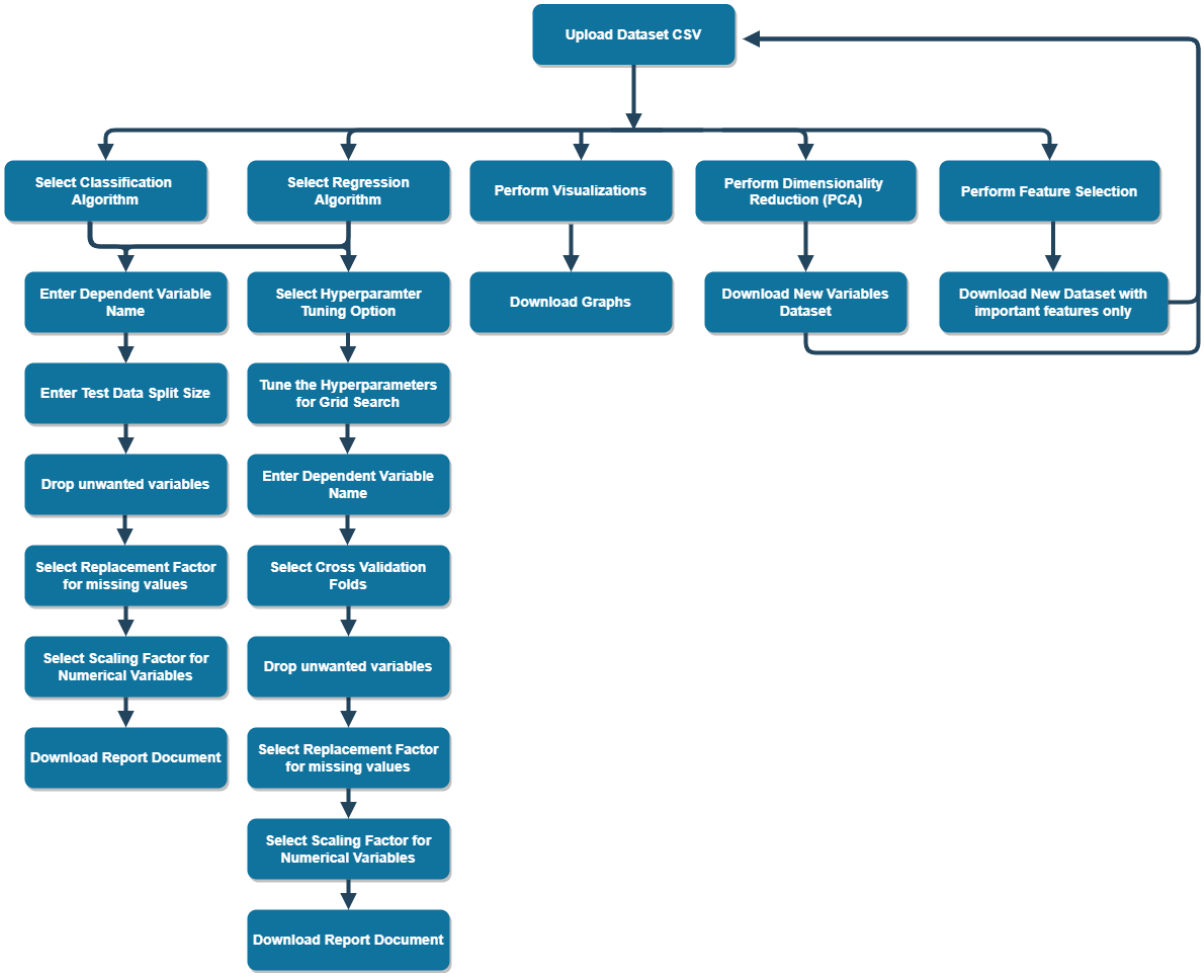


Fig 2.1 Tool workflow

Steps to use the tool

2.1.1) The user has to upload the dataset on the website in CSV format.

2.1.2) All options for different algorithms are given on the home page for user to select.

2.1.3) There is an option given to user in which he can perform all the algorithms available in the tool at once without doing it individually.

2.1.4) After selecting a particular algorithm, user has to enter the name of the dependent variable, test size split, replacement factor for the missing values, and scaling factor for the numerical variables.

2.1.5) After submission of this the results are displayed on the UI and report is generated at the backend which can be downloaded by the user.

2.1.6) If the user is not happen with the results and wants to tune the algorithm, then that can be done by selecting the tuning option for all the algorithms.

2.1.7) The user can perform dimensionality reduction or feature selection as well.

2.1.8) Visualization options are given to the user.

2.2 Requirements for Tool Development:

The requirements for the development of the web tool are:

2.2.1 Pandas

It is a Python package providing fast, flexible, and expressive data structures designed to make working with relational or labelled data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language.

2.2.2 Flask

Flask is an API of Python which allows the users to build up web- applications. A web application is the collection of modules and libraries which helps the Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It helps to run the web applications on the local host itself without the need of any additional server to run locally.

2.2.3 Sklearn

Scikit- Learn is a free machine learning library for the Python programming Language. It features various Classification, Regression and clustering algorithms including linear regression, decision trees, random forest and SVM. It also provides with a lot of data preprocessing techniques.

2.2.4 Matplotlib and Seaborn

There are many visualization libraries in Python available to see the insights of the Data. This helps the user to understand the data well. Both help in providing high level and easily understandable plots related to the data and which can be saved as PNG or JPG image.

2.2.5 Docx

Python helps in creating word documents in which we can add different types of Text formats, images and tables. This is can be used for report generation.

2.2.6 HTML

HTML (Hypertext Markup Language) is a text-based approach to describing how content contained within an HTML file is structured. This markup tells a web browser how to display text, images and other forms of multimedia on a webpage.

2.2.7 CSS

CSS (Cascading style sheets) are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

2.2.8 Heroku

Heroku is a service provided for hosting web applications online. It provides a command line interface to build a communication between our local machine and Heroku servers. Heroku provides high and computationally powerful servers which are paid.

CHAPTER 3
METHODOLOGY

The whole project is based on machine learning and different algorithms used for regression and classification tasks. Almost all the algorithms are provided on the tool for the user to use on their dataset.

This chapter is divided into different parts which comprise of All Algorithm definations, Scaling Techniques, Hyperparameter tuning, Issues in Machine Learning along with their solutions.

3.1 Data Scaling Techniques

There are many types of scaling techniques available in Machine Learning which helps in bringing all the variables on the same scale which helps in increasing the processing speed.

3.1.1) MinMax Scaler

$$\frac{x_i - min(x)}{max(x) - min(x)}$$

It essentially shrinks the range such that the range is now between 0 and 1 (or -1 to 1 if there are negative values). It is sensitive to outliers.

3.1.2) Standard Scaler

$$\frac{x_i - mean(x)}{stdev(x)}$$

The StandardScaler assumes your data is normally distributed within each feature and will scale them such that the distribution is now centered around 0, with a standard deviation of 1.

3.1.3) Robust Scaler

$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)}$$

The RobustScaler uses a similar method to the Min-Max scaler but it instead uses the interquartile range, rather than the min-max, so that it is robust to outliers.

3.2 Regression Algorithms

3.2.1) Linear Regression

Linear Regression is one of the basic regression algorithms where we try to fit a straight line on the data in order to model the linear relationship between in dependent variable and independent variables. Model with one independent variable is called Simple Linear Regression whereas model with more than one independent variable is called Multiple Linear Regression. Linear regression model is often fitted using the least squares approach where we try to minimize the error by evaluating different lines fitted to the data.

3.2.2) Lasso Regression (L1)

Lasso stands for Least Absolute Shrinkage and Selection operator. Lasso Regression is a type of Linear Regression which uses shrinkage. Shrinkage is where data values are shrunk towards the central point such as mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models when you want to automate certain parts of model selection, like variable selection/parameter elimination. Lasso performs regularization which adds a penalty equal to the absolute value of the magnitude of coefficients. This might results in a spare coefficients model making coefficients of some variables as zero and eliminating them from the model hence helping in feature selection.

3.2.3) Ridge Regression (L2)

Ridge is similar to Lasso Regression. The only difference is that Ridge performs regularization which adds a penalty equal to the square value of the magnitude of coefficients. Ridge doesn't result in elimination of coefficients or sparse models. Hence it doesn't help in feature elimination. This makes Lasso far easier to interpret than the Ridge.

3.2.4) Decision Tree Regression

Decision Tree models in the form of tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf

nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. In Decision Tree Regression the output of the node is the average of the data points falling under that section.

3.2.5) Random Forest Regression

Random Forest is a type of Ensemble method in which we create multiple decision trees selecting features and observations randomly making every tree different from each other. When a new point comes in for prediction it is passed through all the trees and for regression task the average of all the trees output is taken as the final prediction.

3.2.6) K Nearest Neighbors Regression

It is called a lazy learner algorithm as there is no training done when we fit the training data to the model. It simply plots the data points in a n-dimensional space. The algorithm uses feature similarity to predict values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set. In KNN Regression the prediction value for the new observation is the average value of all its neighbors.

3.2.7) Support Vector Regression

In simple regression we try to minimize the error rate. While in SVR we try to fit the error within a certain threshold. In SVR we basically consider the points that are within the boundary line. Our best fit line is the line hyperplane that has maximum number of points.

3.3 Classification Algorithms

3.3.1) Logistic Regression

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is binary. Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. It uses Sigmoid function to bring down the regression output in a range of 0 and 1.

3.3.2) Decision Tree Classification

Here we try to split the tree using the variable which provides us the maximum Purity. Gini Index and Information Gain are two different techniques used for evaluation of node purity.

3.3.3) Random Forest Classification

Building Multiple Decision Tree Classifiers help in evaluation the model better. We now take the majority vote of all the tree outcomes. This helps us to be more confident with our results. In most of the cases Random Forest tends to perform better than a single decision tree.

3.3.4) K Nearest Neighbors Classification

Similar to KNN Regression we plot the training data in a N-Dimensional Hyperplane and for prediction of new observation we plot the data point in the same hyperplane and assign it to the majority class of its neighbors.

3.3.5) Support Vector Classification

Support Vector classifier constructs a hyperplane for classes separability, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class, since in general the larger the margin, the lower the generalization error of the classifier. The point lying on the hyperplane are called the Support vectors. Prediction of new point is made by checking on which side of the hyperplane the point lies.

3.4 Different Hyperparameters in Algorithms

3.4.1) Logistic Regression

a) 'C': A control variable that retains strength modification of Regularization by being inversely positioned to the Lambda regulator. It is defined as C = 1/λ. (Default = 1.0)

b) Penalty: Used for Regularization of the model 'L1' or 'L2' for Lasso and Ridge respectively. (Default = l2)

3.4.2) Decision Trees and Random Forest

a) Max Depth - Controls the maximum depth of the tree that will be created. It can also be described as the length of the longest path from the tree root to a leaf. The root node is considered to have a depth of tree. (Default = None)

b) Min Samples Split - The minimum number of samples required to split an internal node. (Default = 2)

c) Min Sample Leaf - The minimum number of samples required to be at a leaf node. (Default = 1)

d) Min Weight Fraction Leaf - The minimum weighted fraction of the sum total of weights required to be at a leaf node. (Default = 0.0)

e) Max Leaf Nodes – Grow a tree with max leaf nodes only. (Default = None)

3.4.3) K Nearest Neighbors

a) Number of Neighbors - Number of neighbors to use for evaluation. (Default = 5)

b) Metric – Distance metric used to calculate distance between 2 points. (Default = minkowski)

3.4.4) Support Vector Machines

a) C: Penalty Parameter which tells how many data points are allowed to be on the incorrect side of the hyperplane. (Default = 1.0)

b) Kernel: Specifies the kernel type to be used in the algorithm (Default: rbf)

c) Degree: Degree of the polynomial kernel function ('poly'). Ignored by all other kernels. (Default = 3)

3.5 Model Problems

3.5.1) Underfitting

Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough. Specifically, underfitting occurs if the model or algorithm shows low variance but high bias. We can check for underfitting if the testing accuracy is much greater than the training accuracy. This might also happened when we are trying to fit a simple model on a really complex dataset, like using a Linear Regression on a non-linear dataset where we might know that ensemble methods or more complex algorithms might perform better than Linear Regression.

3.5.2) Overfitting

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Overfitting is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns. Specifically, overfitting occurs if the model or algorithm shows high variance and low bias. We can also check for overfitting when we see that training accuracy is much greater than the testing accuracy. This can also happen when we try to fit a complex algorithm on a simple dataset, like using Ensemble methods on Linear Dataset. Linear Regression might outperform ensemble or really complex algorithms if the dataset is linear and is satisfying all the assumptions of linear regression. Using complex algorithm might on a simple dataset might lead to overfitting of the model.

3.5.3) Data Imbalance

Imbalance Dataset is primarily in context with datasets involving two or more than two classes. Dataset is said to be imbalanced if number of observations belonging to a particular class are significantly lower than number of observations for the other classes. For eg – In an email spam classification dataset we might have only 1% observations belonging to spam and 99% observations belonging to ham. It is very important to have a balance dataset in order to have a good model.

3.6 Effects of Model Problems

3.6.1) Effect of Underfitting

Underfitting will destroy the accuracy of the model as it will not be able to capture all the underlining trends of the data. Capturing the insights of the data should be done properly for the model to perform well on the testing or unseen data. Testing accuracy will be much higher than the training accuracy on the data.

3.6.2) Effect of Overfitting

In overfitting the model starts learning from the noise which makes the model perform well on the training data but not on the testing data. This means that the model is not able to generalize well on the unseen data. The model should perform well on the unseen that that's why we are training the model in order to predict the future values for a particular observation. Training Accuracy is really high as compared to testing accuracy.

3.6.3) Effect of Data Imbalance

If the data is imbalanced that is the number of observations belonging to a particular class are significantly higher than the other classes then the model might not be able to capture all the classes. It will learn more about the class which is having majority of observations and less about the class which is having minority number of observations. This will reduce recall and precision for the minority class hence affecting the performance of the model.

3.7 Solutions to Model Problems

3.7.1) Underfitting

3.7.1.1) **Add more features:** Occasionally our model is under-fitting on the grounds that the feature items are insufficient. You can add other feature items to unfold it well. This part of machine learning is called a feature engineering in which the user can create new variables that might help the model to get better insights of the data

3.7.1.2) **Add polynomial features:** which are usually utilized as a part of machine learning algorithm. For example, the linear model is more generalized by adding quadratic or cubic terms.

3.7.1.3) **Reduce the regularization parameters:** The motivation behind regularization is to prevent over-fitting, yet now the model has an under-fitting, you have to diminish the regularization parameters. Early stopping and pruning in Decision Trees is also a type of regularization.

3.7.2) Overfitting

3.7.2.1) **Re-cleaning the data**, one cause of over-fitting may also be caused by impure data. If over-fitting occurs, we need to clean the data again.

3.7.2.2) **Increasing the amount of data training**, there is also a reason that the amount of data we use for training is too small, and the proportion of training data to total data is too small.

3.7.2.3) **Adopt the regularization method**. The regularization method includes the L0 regular, the L1 regular, and the L2 regular, while the regularity is generally followed by the norm for the objective function. However, the L2 regularity is generally used in machine learning.

3.7.2.4) **Cross Validation** might also help in reducing overfitting as now the model training is not dependent on a particular training dataset selected from the whole dataset based on the training and testing split.

3.7.3) Data Imbalance

3.7.3.1) **Under sampling Techniques** If a class of data is the overrepresented majority class, under sampling may be used to balance it with the minority class. Under sampling is used when the amount of collected data is sufficient. We reduce the number of observations of the majority class and bring down the number to match the number of minority class observations.

3.7.3.2) **Random oversampling:** Random Oversampling involves supplementing the training data with multiple copies of some of the minority classes. Oversampling can be done more than once (2x, 3x, 5x, 10x, etc.) Instead of duplicating every sample in the minority class, some of them may be randomly chosen with replacement.

3.7.3.3) **SMOTE:** Synthetic Minority Over-sampling Technique. In this technique we try to create artificial observations which are belonging to the minority class so that we can match the number of observations for all the classes. This helps the model to give equal attention to all the different classes. To illustrate the working on SMOTE we can assume an imbalanced dataset with 2 classes. To oversample, take a sample from the dataset, and consider its k nearest neighbors. To create a synthetic data point, take the vector between one of those k neighbors, and the current data point. Multiply this vector by a random number x which lies between 0, and 1. Add this to the current data point to create the new, synthetic data point.

3.7.3.4) **ADASYN:** The adaptive synthetic sampling approach, or ADASYN algorithm, builds on the methodology of SMOTE, by shifting the importance of the classification boundary to those minority classes which are difficult. ADASYN uses a weighted distribution for different minority class examples according to their level of difficulty in learning, where more synthetic data is generated for minority class examples that are harder to learn.

3.7.3.5) Using Class Weights: In all Machine Learning taken from Sci-kit Learn Python Library we have a parameter which we can control called as class weights. This parameter helps us to tell the model if we want it to focus more on a particular class. We can add a dictionary of numbers with the key represents the class name and value represents any decimal number. More the value more is the weight assigned to that class and more focus is given to that class by the model. Hence we assign more weight to the minority class and less weight to the majority class. By default is set to uniform that is equal weightage is given to all the classes.

CHAPTER 4
RESULTS AND EVALUATION

It is very important to evaluate the model in order to judge its performance on the unseen dataset. There might be different methods to evaluate the model as we might be dealing with different problems in our dataset. Few of the Techniques are listed below

4.1 Model Evaluation Techniques for Regression

4.1.1) Mean Square Error (MSE)

Mean Squared Error (MSE) of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value. The MSE is a measure of the quality of an estimator and it is always non-negative, and values closer to zero are better.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i-1}^{n}(Y_i - \hat{Y}_i)^2.$$

4.1.2) Root Mean Square Error (RMSE)

The Root Mean Square error (RMSE) is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the square root of mean squared error (MSE). These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors (or prediction errors. RMSE is always non-negative, and a value of 0 would indicate a perfect fit to the data.

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

4.1.3) Mean Absolute Error (MAE)

Mean Absolute error is an average of the absolute errors |e| = |yi– xi| where yi is the predicted value and xi is the actual value. Absolute value gives only positive output. Taking the average of all the errors that is dividing it with the number of observations will result in Mean Absolute Error.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

4.1.4) R Square (R2)

R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Whereas correlation explains the strength of the relationship between an independent and dependent variable, R-squared explains to what extent the variance of one variable explains the variance of the second variable.

$$\mathrm{R}^2 = 1 - \frac{\text{Explained Variation}}{\text{Total Variation}}$$

4.1.5) Adjusted R Square (Adj R2)

The adjusted R-squared compares the descriptive power of regression models that include diverse numbers of predictors. Every predictor added to a model increases R-squared and never decreases it. Thus, a model with more terms may seem to have a better fit just for the fact that it has more terms, while the adjusted R-squared compensates for the addition of variables and only increases if the new term enhances the model above what would be obtained by probability and decreases when a predictor enhances the model less than what is predicted by chance. In an overfitting condition, an incorrectly high value of R-squared, which leads to a decreased ability to predict, is obtained. This is not the case with the adjusted R-squared.

$$R^2_{adj} = 1 - \left[\frac{\left(1 - R^2\right)(n-1)}{n-k-1} \right]$$

4.2 Model Evaluation Techniques for Classification

4.2.1) Accuracy

It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

It works well only if there are equal number of samples belonging to each class.

For example, consider that there are 98% samples of class A and 2% samples of class B in our training set. Then our model can easily get 98% training accuracy by simply predicting every training sample belonging to class A.

When the same model is tested on a test set with 60% samples of class A and 40% samples of class B, then the test accuracy would drop down to 60%. Classification Accuracy is great, but gives us the false sense of achieving high accuracy.

The real problem arises, when the cost of misclassification of the minor class samples are very high. If we deal with a rare but fatal disease, the cost of failing to diagnose the disease of a sick person is much higher than the cost of sending a healthy person to more tests.

4.2.2) Confusion Matrix

Confusion Matrix as the name suggests gives us a matrix as output and describes the complete performance of the model.

Let's assume we have a binary classification problem. We have some samples belonging to two classes: YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 165 samples, we get the following result.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

There are 4 important terms:

a) **True Positives:** The cases in which we predicted YES and the actual output was also YES.
b) **True Negatives:** The cases in which we predicted NO and the actual output was NO.
c) **False Positives:** The cases in which we predicted YES and the actual output was NO.
d) **False Negatives:** The cases in which we predicted NO and the actual output was YES.

Accuracy for the matrix can be calculated by taking average of the values lying across the "main diagonal" i.e

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

Confusion Matrix forms the basis for the other types of metrics.

4.2.3) Recall

It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

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

4.2.4) Precision

It is the number of correct positive results divided by the number of positive results predicted by the classifier.

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

4.2.5) F1 Score

F1 Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as:

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

F1 Score tries to find the balance between precision and recall.

4.2.6) ROC AUC Score

Area Under Curve(AUC) is one of the most widely used metrics for evaluation. It is used for binary classification problem. AUC of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. Before defining AUC, let us understand two basic terms:

True Positive Rate (Sensitivity): True Positive Rate is defined as TP/ (FN+TP). True Positive Rate corresponds to the proportion of positive data points that are correctly considered as positive, with respect to all positive data points.

$$TruePositiveRate = \frac{TruePositive}{FalseNegative + TruePositive}$$

False Positive Rate (Specificity): False Positive Rate is defined as FP / (FP+TN). False Positive Rate corresponds to the proportion of negative data points that are mistakenly considered as positive, with respect to all negative data points.

$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

False Positive Rate and True Positive Rate both have values in the range [0, 1]. FPR and TPR both are computed at threshold values such as (0.00, 0.02, 0.04, …., 1.00) and a graph is drawn. AUC is the area under the curve of plot False Positive Rate vs True Positive Rate at different points in [0, 1]. As evident, AUC has a range of [0, 1]. The greater the value, the better is the performance of our model.

4.3 Feature Selection and Dimensionality Reduction

4.3.1) Principal Component Analysis (PCA)

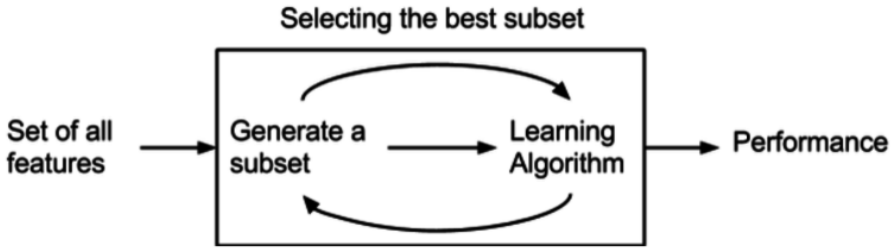
Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors (each being a linear combination of the variables and containing n observations) are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables.

4.3.2) Feature Selection

We want a way to create a model that only includes the most important features. This has three benefits. First, we make our model more simple to interpret. Second, we can reduce the variance of the model, and therefore overfitting. Finally, we can reduce the computational cost (and time) of training a model. The process of identifying only the most relevant features is called “feature selection.” The reason is because the tree-based strategies used by random forests naturally ranks by how well they improve the purity of the node. This mean decrease in impurity over all trees (called Gini impurity). Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of trees. Thus, by pruning trees below a particular node, we can create a subset of the most important features. Tree based algorithms help in feature selection based on the criteria they use to select the node for splitting of the variables.

4.3.3) Step Forward Feature Selection

Step forward feature selection starts with the evaluation of each individual feature, and selects that which results in the best performing selected algorithm model. That depends entirely on the defined evaluation criteria (AUC, prediction accuracy, RMSE, etc.). Next, all possible combinations of that selected feature and a subsequent feature are evaluated, and a second feature is selected, and so on, until the required predefined number of features is selected.



4.3.4) Backward Feature Selection

It starts with the entire set of features and works backward from there, removing features to find the optimal subset of a predefined size. The performance of the model can be evaluated using any of the evaluation metrics.

4.3.5) Hybrid Feature Selection

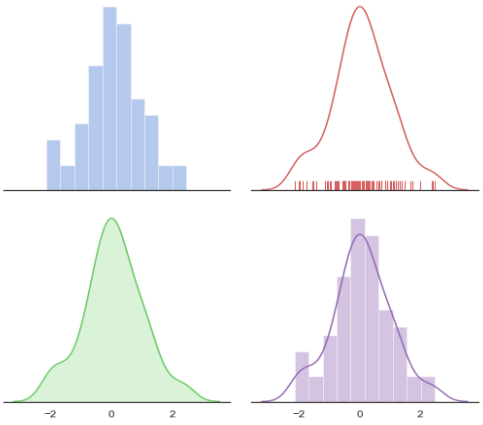
It is a combination of both Step Forward and Backward Feature Selection. It starts with one variable and keeps on adding and removing the variables depending on the improvement of the model performance.

CHAPTER 5
VISUALIZAITIONS

5.1 Visualizations

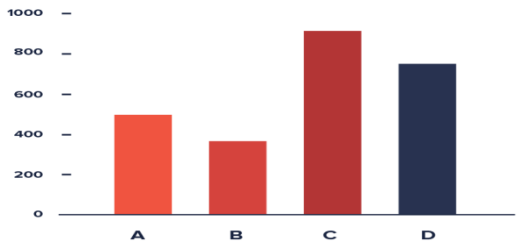
5.1.1) Distribution Plot

A distribution plot displays a distribution and range of a set of numeric values plotted against a dimension.



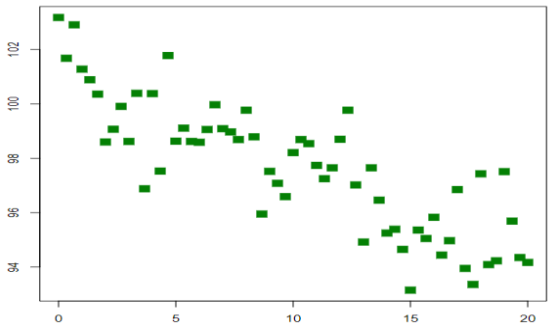
5.1.2) Bar Plot

A bar graph is a chart that uses bars to show comparisons between categories of data. The bars can be either horizontal or vertical. Bar graphs with vertical bars are sometimes called vertical bar graphs. A bar graph will have two axes. One axis will describe the types of categories being compared, and the other will have numerical values that represent the values of the data. It does not matter which axis is which, but it will determine what bar graph is shown. If the descriptions are on the horizontal axis, the bars will be oriented vertically, and if the values are along the horizontal axis, the bars will be oriented horizontally.



5.1.3) Scatter Plot

Scatter plot is a type of plot drawn using Cartesian coordinates to display values for typically two variables for a set of data. The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.



5.1.4) Heatmap

A heatmap is a graphical representation of data where the individual values contained in a matrix are represented as colors. We use heatmap to depict the correlation between the independent variables. Color Scheme is available to clearly visualize the value of the variable. Different colors are assigned to different values of the map in order to see the intensity of the correlation.

5.1.5) Box Plot

A boxplot is a graph that gives you a good indication of how the values in the data are spread out.

Boxplots are a standardized way of displaying the distribution of data based on a five number summary (“minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”).

a) **Median (Q2/50th Percentile)**: the middle value of the dataset.

b) **First quartile (Q1/25th Percentile)**: the middle number between the smallest number (not the "minimum") and the median of the dataset.

c) **Third quartile (Q3/75th Percentile)**: the middle value between the median and the highest value (not the "maximum") of the dataset.

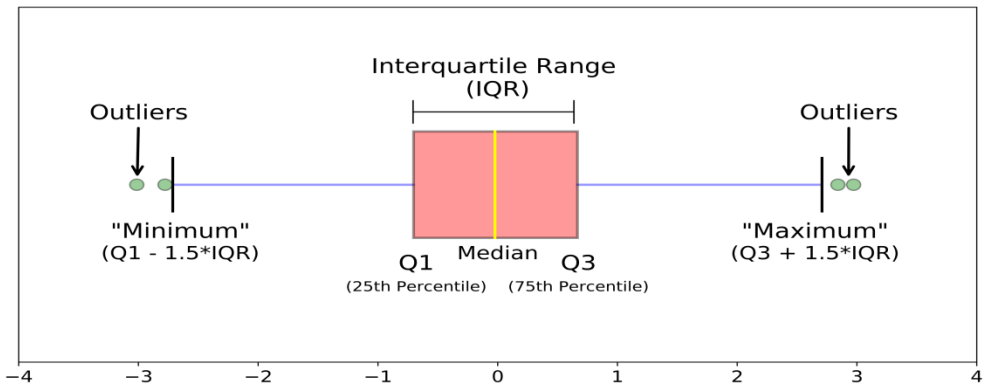
d) **Interquartile range (IQR)**: 25th to the 75th percentile.

e) **Whiskers (shown in blue)**

f) **Outliers (shown as green circles)**

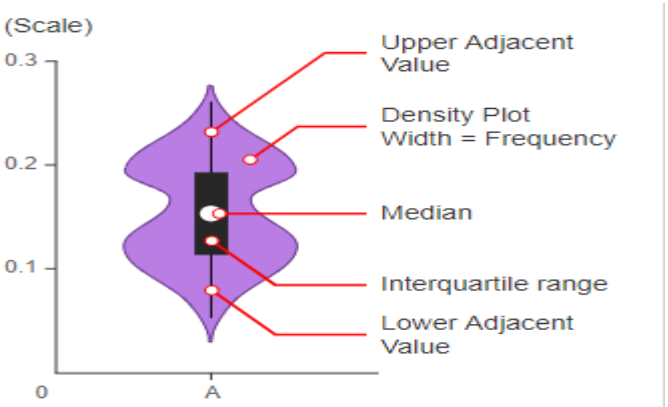
g) **Maximum**: Q3 + 1.5*IQR

h) **Minimum**: Q1 -1.5*IQR



5.1.6) Violin Plot

A Violin Plot is used to visualize the distribution of the data and its probability density.

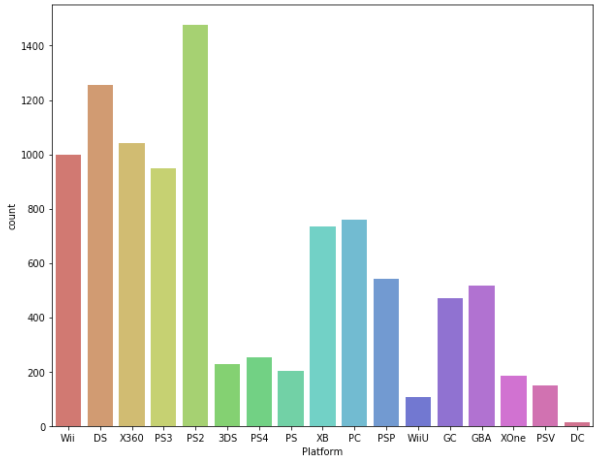


The white dot in the middle is the median value and the thick black bar in the center represents the interquartile range. The thin black line extended from it represents the upper (max) and lower (min) adjacent values in the data.

A violin plot is more informative than a plain box plot. While a box plot only shows summary statistics such as mean/median and interquartile ranges, the violin plot shows the full distribution of the data. The difference is particularly useful when the data distribution is multimodal (more than one peak). In this case a violin plot shows the presence of different peaks, their position and relative amplitude.

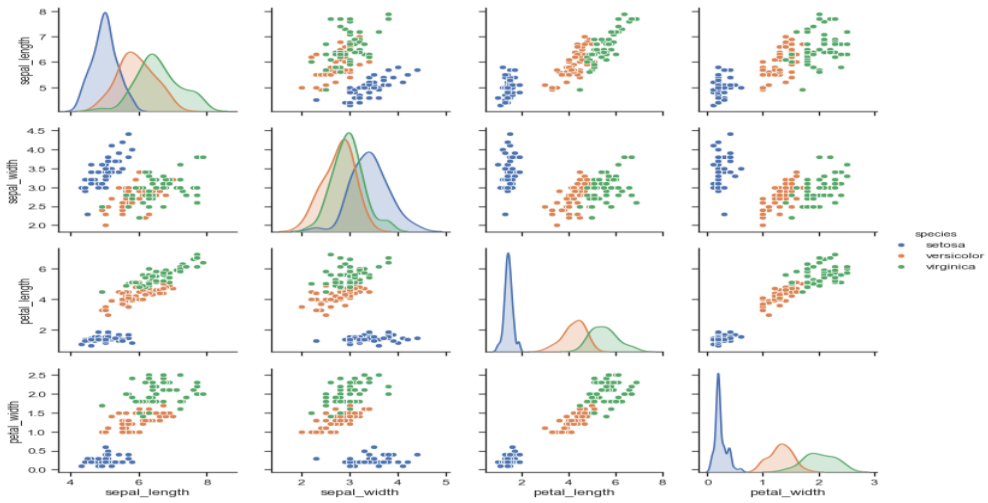
5.1.7) Count Plot

Show the counts of observations in each categorical bin using bars. Different bars are shown with different colors each representing the class on the x-axis and count on y-axis.



5.1.8) Pair Plot

It represents pairwise relation across the entire Dataframe and supports an additional argument called hue for categorical separation. What it does basically is create a plot between every possible numerical column.



CHAPTER 6
FUTURE SCOPE

This system will be helpful to all the companies trying to implement machine learning in their business. Starting with evaluation and insights of the all the systems will help them know more about the data and also help them to improve the performance of the data. There are few improvements that can be done in this tool

1) Adding Unsupervised Machine Learning Algorithms
2) Adding more complex algorithms like AdaBoost, XGBoost
3) Adding more Visualization support.
4) Hosting the tool on high configuration machine that supports GPUs and are faster in computation

CHAPTER 7
REFERENCES

1) https://www.kdnuggets.com/2018/06/step-forward-feature-selection-python.html

2) https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51

3) https://seaborn.pydata.org/examples/index.html

4) https://en.wikipedia.org/wiki/Principal_component_analysis

5) https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

6) https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis