Report For Partial completion for the Coursework of Statistical Machine Learning (AM41ML)

Submitted by:

Name: Karan

Student ID: 230293944

ABSTRACT

This report outlines the comprehensive process undertaken to build a predictive model with the primary aim of achieving less than 5% errors in predicting labels on test data. The dataset provided, named "Coursework dataset.csv," underwent a systematic and thorough analysis involving data cleaning, exploration, and visualization, followed by model selection and testing. The data cleaning phase involved meticulous handling of missing values, addressing outliers, and ensuring data integrity. Subsequently, data exploration and visualization techniques were employed to gain insights into the dataset's characteristics, distributions, and potential patterns. This exploratory analysis laid the foundation for informed decisions in later steps. Throughout the process, clear justifications for each decision were provided, ensuring transparency and accountability. The report is structured in a coherent manner, presenting logical steps that lead from data cleaning to model selection. Plots and visualizations are strategically incorporated to enhance clarity and aid in the interpretation of findings.

TABLE OF CONTENT

1. INTRODUCTION

In the field of data science, the importance of thorough dataset preprocessing, and the use of proficient classification algorithms cannot be emphasized enough. This assessment delves into the meticulous process of cleaning and analyzing a dataset given in CSV format. The dataset includes an index with two numerical headers named “Variable 1” and “Variable 2” of float type and a column named "Label," having two unique categorical values - "Red" and "Blue". The first steps involve cleaning the dataset to ensure accuracy and uniformity. Subsequently, the dataset is visualized using scattered plot techniques, offering insights into the distribution and patterns of the data. The SVM algorithm is implemented, incorporating a Radial Basis Function (RBF) kernel, and varying the gamma values to examine their impact on the results. The focus shifts towards minimizing validation error through hyperparameter tuning and cross-validation techniques. This critical phase ensures that the SVM model is fine-tuned for best performance, balancing bias and variance. Finally, the dissertation culminates in the testing phase, where the SVM model is applied to the testing dataset, and the results are interpreted. The overarching goal is to present a comprehensive exploration of data cleaning methodologies, SVM classification with varying hyperparameters, and the significance of cross-validation in enhancing model robustness.

2. CLEANING PROCESS

As the databases grew in popularity in the 1970s, ETL was introduced as a process for integrating and loading data for computation and analysis, eventually becoming the primary method to process data for data warehousing projects.

Data cleaning supplies the foundation for data analytics and machine learning workstreams. Through a series of rules, data cleaning helps to cleanse and organize data in a way which addresses specific intelligence needs, but it can also tackle more advanced analytics, which can improve back-end processes or end user experiences. ETL is often used by an organization to:

EXTRACTION – Dataset is extracted from the source location. We can extract data from a variety of data sources, which can either be structured or unstructured.

TRANSFORM - In the staging area, the raw data undergoes data processing. Here, the data is transformed and combined for its intended analytical use case.

LOAD – Once the transformation is completed, data is ready to upload to the destination location.

2.1. DATA CLEANING

The raw data undergoes data processing. Here, the data is transformed and merged for its intended analytical use case. This Phase involves following operations on the dataset:

- Removal of unnecessary column named “INDEX”.
- Removal of N/A or NAN from Label column. Eliminating 5 rows from total dataset.
- Uniformity in datatype – change object to float using coerce attribute for “Variable 1” and “Variable 2” and change coerce values from NAN to 0.
- Removal of outliers - To remove outliers, boxplots have been used for search of outliers. There are 10 outliers, all part of Variable 1 column. Using their index values, we have dropped those rows. Resulting in 4985 rows and 3 columns. There is no outlier in Variable 2 column.

2.2. DATA EXPLORATION

Exploring data encompasses a spectrum of methods, ranging from hands-on manual analysis to the use of automated software solutions designed for data exploration. These tools visually navigate through datasets, uncovering and highlighting relationships among various data variables, the structure of the dataset, the presence of outliers, and the distribution of data values to reveal patterns and points of interest, enabling data analysts to gain greater insight into the raw data.

2.3. VISUALIZATION

Data visualization involves presenting data using familiar graphics like charts, plots, infographics, and animations. These visual representations effectively convey intricate data relationships and insights in a format that is easily comprehensible.

For this assessment, scattered plot representation is used. As there are only 2 categories in the Label and the dataset is not large so visualization is quite better using this technique and as a result, we can easily find extreme values, outlier values (if there is any) and detect patterns between the categories. Use of seaborn library has been used for eloquent representation with distinct color for different Label value.

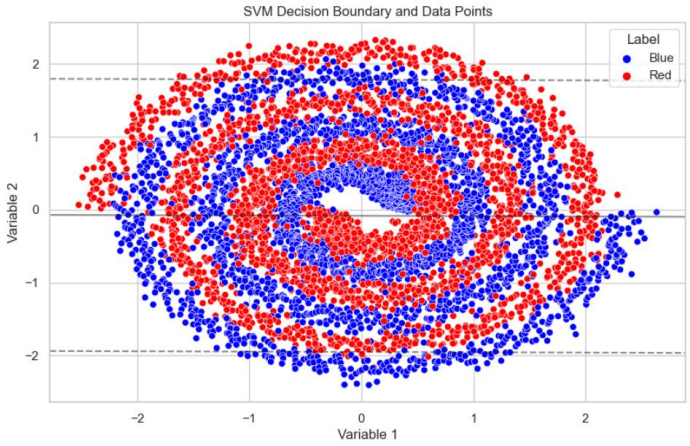

Fig 1

3. SVM MODEL

3.1. Model description

A support vector machine (SVM) falls under the supervised learning models to solve complex classification, regression, and outlier detection problems by performing best data transformations that determine boundaries between data points based on predefined classes, labels, or outputs. The plotting is done in an n-dimensional space where n is the number of features of a particular data. Then, classification is carried out by finding the most suitable hyperplane that separates the two (or more) classes effectively.

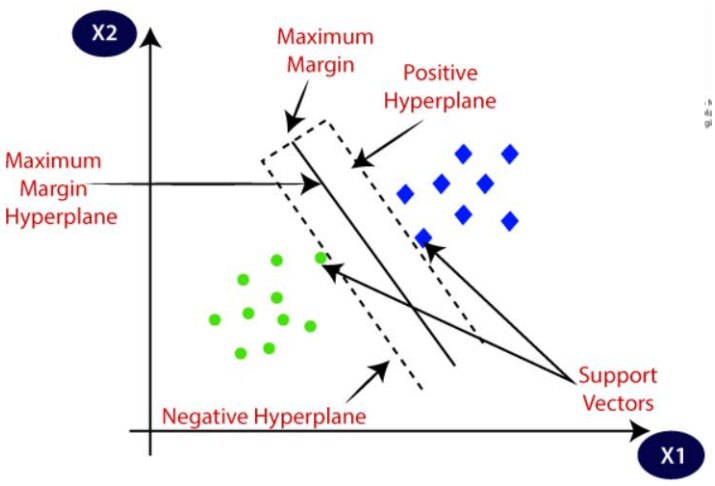

Fig 2

3.1.1. Types of SVM

Linear SVM: Linear SVM is used for data that are linearly separable i.e., for a dataset that can be categorized into two categories by using a single straight line. Such data points are termed as linearly separable data, and the classifier is used described as a Linear SVM classifier.

Non-linear SVM: Non-Linear SVM is used for data that are non-linearly separable data i.e., a straight line cannot be used to classify the dataset. For this, we use something known as a kernel trick that sets data points in a higher dimension where they can be separated using planes or other mathematical functions. Such data points are termed as non-linear data, and the classifier used is termed as a Non-linear SVM classifier.

3.1.2. Hyperplane

To classify data in n-dimensional space, multiple lines or decision boundaries can be drawn. However, to pinpoint the best decision boundary that effectively categorizes the data points. This best boundary is termed the hyperplane in Support Vector Machines (SVM).

The way the hyperplane looks depends on the features in the data. If there are two features, the hyperplane is like a straight line. But if there are three features, it becomes a flat, two-dimensional plane.

In SVM, our goal is to make this hyperplane with the biggest gap between data points. This big gap makes our classification stronger and more correct.

3.2. SVM Linear Kernel

The Dataset is split into 2 parts which are called training dataset and testing dataset. The ratio we have chosen is 80-20 percent. 80 percent of data which is equal to 3988 rows of dataset will be used as training dataset and rest of the set that is equal to 977 rows will be used as testing dataset. Generally, the dataset split into 80-20 ratio of training and testing. But to get better accuracy and lesser error, we can adjust this ratio accordingly. The more you train your dataset, the higher accuracy will be achieved on the testing dataset.

$$X\,train, X\,test, y\,train, y\,test = train\,test\,split\,(X, y, test\,size = 0.2, random\,state = 42)$$

These are typically used to denote the features (input) and labels (output) of a dataset, respectively. 'X' is a matrix or data frame holding the features, and **y** is an array or series holding the corresponding labels.

- X = X train will hold training set of features while X test will hold testing set of features.
- y = y train will hold label of training set and y test will hold label of testing set.
- Test size = 0.2 is said as well. The rest (0.8) is a training set.
- Random state = 42. Ensure reproducibility. The split will be the same every single time we run the code.

$$svm\ model=SVC\ (kernel=\ 'linear',\ C=1)$$

$$svm\ model.fit(X\ train,y\ train)$$

SVC - Support Vector Classification:

SVC is a class in scikit-learn specifically designed for support vector machine classification.

kernel='linear': This parameter specifies the type of kernel function to be used in the SVM. In this case, the kernel is set to 'linear' showing a linear kernel. A linear kernel implies that the decision boundary between classes will be a straight line.

C=1:

The C parameter controls the regularization strength in the SVM model. A smaller C value leads to a wider margin but may misclassify some data points. A larger C value allows for a narrower margin but aims to classify all training points correctly. In this case, C is set to 1, suggesting a balanced approach.

The training accuracy with this approach is .507 or 50.7 % showing that at least half of the time, prediction is correct. A training accuracy of 50.7% showing that the model's shows that it has some limitation. The model could be underfitting the training data, meaning it is too simplistic to capture the complexities in the dataset. It cannot show the dominant trend within the data, resulting in training errors and deficient performance of the model. If a model cannot predict new data well, then it cannot be used for classification or prediction tasks. High bias and low variance show that the model is underfit.

This underfitting can be reduced by following techniques:

- Introduce more features and complexity in the data.
- Increase the duration of training for better results.
- Include more data in the training set.
- Removing noise so it will allow to focus on the key features.

3.2.1. Linear Kernel Advantages

For better understanding, the linear kernel model is here compared with non-linear kernel for this specific dataset.

The reason for selecting this model is as follows:

- Data is linearly separated that is 80-20% in testing and training dataset, meaning that the classes can be effectively separated by a straight line (or hyperplane in higher dimensions), a linear kernel SVM considered as better choice over the non-linear kernel.
- Linear SVM are less prone to overfitting, especially when dealing with limited training data like in our case where dataset is limited (4985 rows). Whereas Non-linear kernels may be more susceptible to overfitting in such scenarios.
- Linear kernels are faster to train in comparison with non-linear kernels. Especially if the dataset is extremely large then and linear kernel is being apt depending on the complexity of the data. The computation time difference can be extreme in some cases. The linear kernel does not perform any mapping, it is generally faster to train your classifier than with other kernels.
- Linear kernels are best to apply on linearly separable data just like in our case (the data is linearly separable). Here in our dataset, there are only 2 features, that is red and blue. If you plot your dataset samples in a chart or scatter plot using the 2 features red and blue, you'll be able to see how samples from different classes position in relation to each other. The visualization is very constructed as compared to other.
- The number of features (dimensionality) is relatively low, and the data shows a linear separation, a linear SVM can perform well without the need for complex non-linear transformations.
- When dealing with a dataset having only two features, the decision to choose a linear kernel SVM is often based on the simplicity of the data distribution and the desire for a straightforward, interpretable model. However, it's essential to consider the nature of the problem, explore the data, and potentially experiment with both linear and non-linear kernels to figure out the best-performing approach.

3.3. SVM- Non-Linear

The basic idea is to transform the input features into a higher-dimensional space where a linear decision boundary can be found. This transformation is achieved through a kernel function. The original feature can always be transformed to some higher-dimensional feature where the training set is separable.

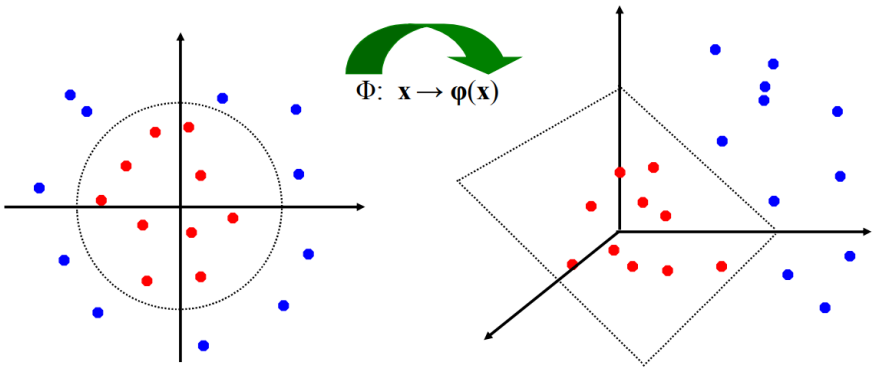

Fig 3

Let's denote the input features as 'x' and the transformed feature space as $\varphi(x)$. The decision function for a linear SVM is given by:

3.3.1. SVM with RBF Kernel

The Gaussian kernel, commonly known as the Radial Basis Function (RBF) kernel, holds prominence in Support Vector Machines (SVMs) for its utility in handling non-linear classification and regression challenges. This kernel is particularly adept at addressing intricate, non-linear relationships within datasets. It is a robust machine learning algorithm suitable for both classification and regression tasks. Operating as a non-parametric model, it excels in scenarios involving non-linear and high-dimensional data, making it a versatile tool in data analysis and predictive modeling.

$$K(X_1, X_2) = exp(-\gamma \|X_1 - X_2\|^2)$$

where, $\|X_1 - X_2\|$ = distance between 2 points.

```
svm_model_rbf = SVC (kernel='rbf', C=1, gamma=gamma)
svm_model_rbf.fit(X_train_scaled, y_train)
```

The general code for using the rbf kernel in SVM is given in the above statement. Here, rbf with cost (C) equal to 1.

3.4. CROSS VALIDATION

In machine learning, to assess the efficacy of a model when confronted with new, unseen data. This technique entails partitioning the available dataset into distinct folds or subsets, choosing one of these

folds as a validation set, and training the model on the remaining folds. This process is repeated multiple times, with each iteration employing a different fold as the validation set. Through this systematic approach, cross-validation supplies a robust evaluation of a model's generalization performance and helps mitigate potential biases introduced by a specific data split. Finally, the results from each validation step are averaged to produce a more robust estimate of the model's performance. The main purpose of cross validation is to prevent overfitting, indicate high variance and low biasness. By evaluating the model on multiple validation sets, cross validation supplies a more realistic estimate of the model's generalization performance, i.e., its ability to perform well on new, unseen data.

grid_search = GridSearchCV(SVC (kernel='rbf'), param_grid, cv=5, scoring='accuracy')

CV: The cv parameter figures out the cross-validation splitting strategy. In this case, it's set to 5, implying that 5-fold of cv. Each iteration of the cross-validation involves training the model on 4 folds and validating on the remaining 1-fold. This process is iterated 5 times, with each fold serving as the validation set exactly once.

scoring= 'accuracy': The scoring parameter specifies the evaluation metric to be used for selecting the hyperparameters. In this case, accuracy is used as the scoring metric.

Validation error rate: Validation error rate is a metric that measures the error or misclassification rate of a machine learning model on a validation dataset. It is essentially the proportion of incorrectly classified instances in the validation set.

$$Validation\ Error\ Rate = Total\ Number\ of\ Instances\ in\ Validation\ Set \div Number\ of\ Misclassifications\ in\ Validation\ Set$$

4. INTERPRETATION

4.1. SVM with RBF Kernel:

In our project, we specifically use 3 different gamma values [1, 10, 1000] and the output is as follows:

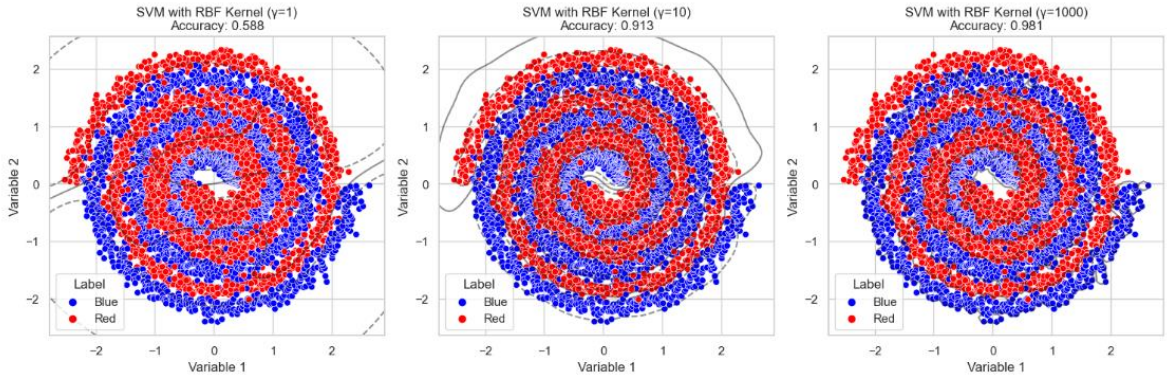

Fig 4

The RBF kernel has a hyperparameter called γ that determines the shape of the decision boundary. High values of γ lead to a more complex decision boundary, potentially leading to overfitting. Proper tuning of γ is crucial for model performance.

In the above image, we can see three different scattered plot diagrams. Each diagram is different because of its different gamma value decision boundary, shown by line. On the X-axis, we plotted the variable 1 and Y-axis is for Variable 2. The range of value for all the data (variable 1 and variable 2) is between range –3 to 3. The category can be seen here with Label (red and blue).

- When γ=1, we have the accuracy of 58.8%, the model might be underfitting and the decision boundary also very distorted.
- When γ=10, we have the accuracy of 91.3%, the model is giving particularly good accuracy but there might be possibility of overfitting as we have not use cross-validation yet to mitigate the overfitting.
- When γ=1000, we have the accuracy of 98.1%, the model producing almost exact accuracy which is too good to be true. There is a high chance that the model is overfitted.

4.2. CROSS-VALIDATION

To overcome the overfitting and produce the best accuracy for the model with the given dataset, use cross-validation method. Instead of relying on a single split, cross-validation uses multiple folds of the data, training the model on different subsets and validating on others. This gives a more representative estimate of how well the model generalizes. Cross-validation helps find overfitting by revealing if a model's performance significantly drops on validation sets compared to the training set. It allows us to assess how well different hyperparameter configurations perform across multiple folds, helping you choose values that result in good generalization rather than overfitting to the training data.

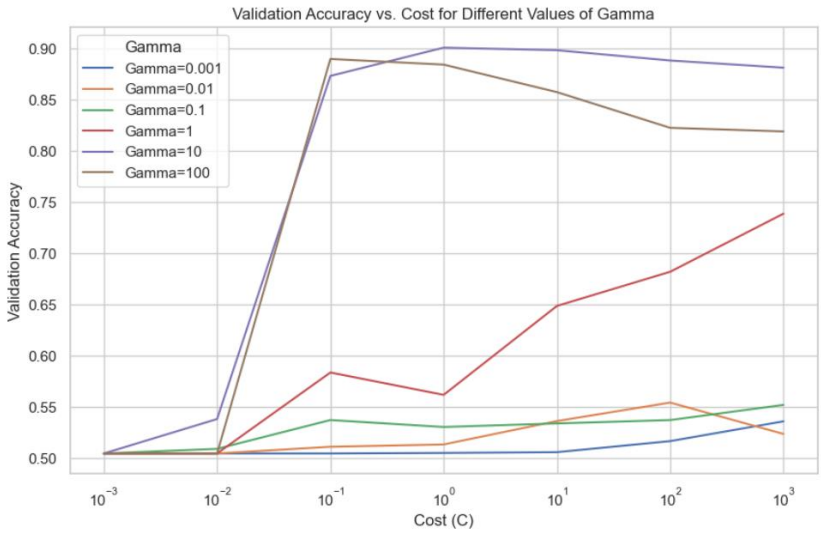

Fig 5

In figure 5, we can interpret that it is producing variant of validation accuracy for different values of gamma(γ) shown by different colored line. The highest accuracy we can achieve when γ is 10. The value is achieved when the cost is 1.

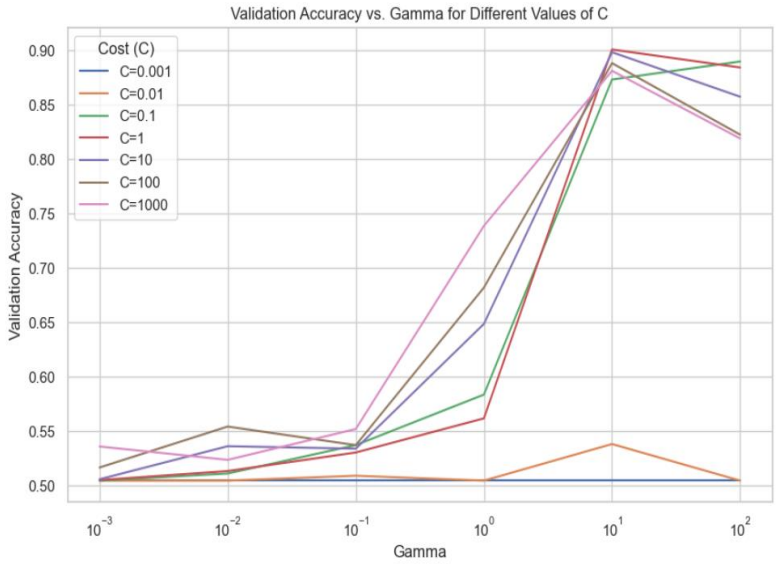

Fig 6

In figure 6, different validation accuracy is achieved for different value of cost (C). The best-case scenario occurs when the cost is either 1 or 10. If we look closely, we can see that there is a very minuscule difference and validation accuracy is bit higher when C = 1. And the value is achieved when gamma is 10.

5. MODEL TESTING

With the help of Support vector machines, we have now trained out data using RBF kernel and use cross validation to drop biasness. We first need to compute the best case of γ and cost, which will give us the best accuracy. In our case, the best case is when the γ **= 10** and **Cost = 1.** The testing dataset consists of 997 rows.

Using our best-case values, we will perform the testing on the testing dataset which is unseen to the machine and provide us with the view that how much it is trained and the accuracy of prediction on testing dataset.

Accuracy is 90.17 % on the testing dataset. The model is neither underfit nor overfit. The use of cross-validation during the model training phase helps mitigate overfitting.

6. CONCLUSION/ RESULT

The test accuracy on the testing dataset is 90.17%. SVM model with linear kernel, the accuracy is only 50.17% on the testing dataset. With the help of RBF kernel method, which is a part of non-linear SVM, we were able to find the accuracy for 3 different cases of γ (1,10,1000). In case of γ=1, the accuracy is around 58% which implies that it might be underfit while in case of γ=10, accuracy is 91%. In last case where γ = 1000, the accuracy is 98.1 %, which is most probably overfitted model. With the help cross-validation method, we came to conclusion that γ= 10 and cost C= 1 is our best-case where we were able to achieve the highest accuracy. This will help to overcome overfitting in the model. After applying cross-validation technique called grid-search, we were finally able to achieve the result of 90.17% accuracy. But as the project guideline mentioned, we cannot achieve the required accuracy of 95%.

7. OTHER APPROACH

For better result in accuracy, we can use other approaches instead of SVM algorithm.

Ensemble Method:
Ensemble methods are machine learning techniques that combine the predictions of multiple individual models to achieve highly correct model. The idea behind ensemble methods is to use the diversity among the individual models to improve overall performance, making the ensemble more accurate and robust than its individual components.

Common Ensemble classifiers:

- **Bagging -** Build many bootstrap replicates of the data, train many classifiers, vote the results.
- **Stacking** - Involves training multiple diverse base models on the original data.
- **Boosting -** Builds a sequence of weak learners (models that perform slightly better than random chance).
- **Forest Model -** A specific type of ensemble method that combines bagging and random feature selection for decision trees. Multiple decision trees are trained on different subsets of the data and different subsets of features. Ensemble methods often perform better than individual models, reducing overfitting and improving generalization to unseen data.

With forest model implementation with multiple decision tree, we can achieve more than 95% of accuracy with no-overfitting. With this specific model, our project can produce **accuracy of 96.25%.**

Neural Network: This approach involves training and combining the outputs of several neural networks to create a more robust and accurate model. Ensemble neural networks aim to use the diversity of individual models to improve overall performance and generalization. Neural networks, especially deep neural networks, have a high ability to learn complex and non-linear relationships in data. They can automatically extract hierarchical features from raw input.

8. BIBLIOGRAPHY

- https://www.ibm.com/topics/etl
- https://www.ibm.com/topics/data-visualization
- https://www.geeksforgeeks.org/cross-validation-machine-learning/
- https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://wiki.pathmind.com/neural-network
-