Spam Classification using Machine Learning

Authors – Chaithanya Pramodh Kasula and Aishwarya Varala

Course Professor's Name – Dr. Liao DuoDuo

University Name – George Mason University

Course Number and Name – AIT-614, Big Data Essentials

Date – 14th, May 2020

Spam Classification using Machine Learning

1. Abstract

Junk email sent out in bulk to a random list of users is known as a spam email. In commercial advertising, spam emails are generated by bots and infected computers (Cisco, 2020). 95 percent of the emails sent worldwide are presumed to be spam indicating the importance of efficient spam email identification tools (Runbox, 2019). The classification of an email into 'spam' or 'not spam' can be made better by leveraging the power of ML/AI. In this paper, Machine Learning models like Random Forests, Support Vector Machines, and Neural Networks are used to perform a classification task on a dataset to predict whether the input email is a spam or not. Data is streamed from a Kafka producer to a Kafka consumer from a data source and is integrated into an analytics program. The paper describes data pre-processing steps, training related information, hyperparameter tuning, cross-validation, and performance evaluation of the classifiers on the test set. New techniques involving dimensionality reduction and ensemble techniques have been implemented.

2. Introduction

Technically spam is an unsolicited commercial email being sent without the recipient's permission. Sender's do not pay anything for sending a spam email whereas recipients loose time and carrier bandwidth. Hence, commercial advertising uses spam to advertise their products. Spam fills the inbox with a large number of absurd emails. They try to obtain your contact list and alter one's search results. Sometimes, people may receive hundreds of spam emails which a huge waste of time and resources. By preventing the sharing of personal information like credit card details, email, and not responding to unwanted emails can prevent spam mails. BCC can be used while sending bulk emails that prevent the privacy of others' email ids (Security-faqs, n.d.). Even after following precautions, we may still receive spam emails. This can be resolved by using an advanced spam filter which automatically labels the email as spam even before it reaches the inbox. Knowledge Engineering and Machine Learning are the two common approaches used in email filtering (Awad, 2011). The paper aims to train several Machine learning models like Support Vector Machines, Random Forest and Neural Network to classify the email as spam or not.

2.1 Backgrounds and/or related work

(Rusland et. al., 2017) use WEKA and analyze the performance of the Naïve Bayes model on the task [6] (Lee et. al., 2010) describe how feature selection and parameter optimization can be performed to scale the application. (Alistair et. al., 2007) make use of the Fuzzy Support Vector to perform the classification task. (Priyaa et al, 2010) propose a feature reduction method for Spam Classification. (Aakanksha et al., 2016) studied the effect of J48, Naïve Bayes, Random Forest, and Lazy IBK algorithm over this dataset.

3. Objectives

The objective of the project is to learn how data can be directed to the Machine Learning model using data engineering. To apply Kafka to stream data from Producer to Consumer and integrate it with the analytics program. To make use of visualization software to effectively convey analysis and results.

4. The Dataset

4.1 Selection

The name of the dataset is 'Spambase Data Set' (UCI Machine Learning Repository, 1999). It was created by Hewlett-Packard Labs in California.

4.2 Description

This data was internally used in the organization to determine if the email is spam or not. There was an approximate 7 percent misclassification error in data. The total number of instances is 4601, of which spam instances are 1813, which constitutes 39.4% of the entire dataset and non-spam instances are 2788 that makes up 60.6% of the data. There are 58 attributes where 57 are continuous attributes and one is the nominal class label. The last column determines if the mail is spam (1) or not (0).

4.3 Dataset schema

Table I represents the list of attributes present in the dataset.

Feature name	Feature Type	Sample Values
word_freq_make:	continuous.	0
word_freq_address:	continuous.	0
word_freq_all:	continuous.	0.64
word_freq_3d:	continuous.	0
word_freq_our:	continuous.	0.32
word_freq_over:	continuous.	0
word_freq_remove:	continuous.	0
word_freq_internet:	continuous.	0
word_freq_order:	continuous.	0
word_freq_mail:	continuous.	0
word_freq_receive:	continuous.	0
word_freq_will:	continuous.	0.64
word_freq_people:	continuous.	0
word_freq_report:	continuous.	0
word_freq_addresses:	continuous.	0
word_freq_free:	continuous.	0.32
word_freq_business:	continuous.	0
word_freq_email:	continuous.	1.29
word_freq_you:	continuous.	1.93
word_freq_credit:	continuous.	0
word_freq_your:	continuous.	0.96
word_freq_font:	continuous.	0
word_freq_000:	continuous.	0
word_freq_money:	continuous.	0
word_freq_hp:	continuous.	0

word_freq_hpl:	continuous.	0
word_freq_george:	Continuous.	0
word_freq_650:	continuous.	0
word_freq_lab:	continuous.	0
word_freq_labs:	continuous.	0
word_freq_telnet:	continuous.	0
word_freq_857:	continuous.	0
word_freq_data:	continuous.	0
word_freq_415:	continuous.	0
word_freq_85:	continuous.	0
word_freq_technology:	continuous.	0
word_freq_1999:	continuous.	0
word_freq_parts:	continuous.	0
word_freq_pm:	continuous.	0
word_freq_direct:	continuous.	0
word_freq_cs:	continuous.	0
word_freq_meeting:	continuous.	0
word_freq_original:	continuous.	0
word_freq_project:	continuous.	0
word_freq_re:	continuous.	0
word_freq_edu:	continuous.	0
word_freq_table:	continuous.	0
word_freq_conference:	continuous.	0
char_freq_;:	continuous.	0
char_freq_(:	continuous.	0
char_freq_[:	continuous.	0
char_freq_!:	continuous.	0.778
char_freq_$:	continuous.	0
char_freq_#:	continuous.	0
capital_run_length_average:	continuous.	3.756
capital_run_length_longest:	continuous.	61
capital_run_length_total:	continuous	278

Table I. Dataset Schema with Sample Values

The following is the description of the attributes of the dataset.

- There are 48 continuous real attributes of the type word_freq_*WORD*. This is equal to the percentage of words present in the email that match the *WORD*. That is ((number of times the *WORD* appears in the email)/(total number of words in the email))* 100.

- There are 6 continuous real attributes of the type char_freq_CHAR. This is equal to the percentage of characters present in the email that match the *CHAR*. That is ((number of times the *CHAR* appears in the email)/(total number of *Chars* in the email))* 100.
- There is 1 continuous real attribute of the type capital_run_length_average. That is equal to the average length of uninterrupted sequences of capital letters
- There is 1 continuous real attribute of the type capital_run_length_longest. That is equal to the longest length of uninterrupted sequences of capital letters
- There is 1 continuous real attribute of the type capital_run_length_total. That is the sum of the length of the uninterrupted sequences of capital letters.
- 1 column is nominal. It has 0 or 1 as its values The class attribute of type spam implies if the e-mail was considered spam (1) or not (0).

word_freq_make	word_freq_address	word_freq_all	word_freq_3d	word_freq_our	word_freq_over	word_freq_remove	word_freq_internet	word_freq_order	word_freq_mail	word_freq_receive
0	0.64	0.64	0	0.32	0	0	0	0	0	0
0.21	0.28	0.5	0	0.14	0.28	0.21	0.07	0	0.94	0.21
0.06	0	0.71	0	1.23	0.19	0.19	0.12	0.64	0.25	0.38
0	0	0	0	0.63	0	0.31	0.63	0.31	0.63	0.31
0	0	0	0	0.63	0	0.31	0.63	0.31	0.63	0.31
0	0	0	0	1.85	0	0	1.85	0	0	0
0	0	0	0	1.92	0	0	0	0	0.64	0.96
0	0	0	0	1.88	0	0	1.88	0	0	0
0.15	0	0.46	0	0.61	0	0.3	0	0.92	0.76	0.76
0.06	0.12	0.77	0	0.19	0.32	0.38	0	0.06	0	0
0	0	0	0	0	0	0.96	0	0	1.92	0.96
0	0	0.25	0	0.38	0.25	0.25	0	0	0	0.12
0	0.69	0.34	0	0.34	0	0	0	0	0	0
0	0	0	0	0.9	0	0.9	0	0	0.9	0.9
0	0	1.42	0	0.71	0.35	0	0.35	0	0.71	0
0	0.42	0.42	0	1.27	0	0.42	0	0	1.27	0
0	0	0	0	0.94	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0.55	0	1.11	0	0.18	0	0	0	0

Fig. 1 Sample dataset with few features and values

Figure 1 shows a few attributes of the 57 attributes present in the dataset.

4.4 Data Exploration

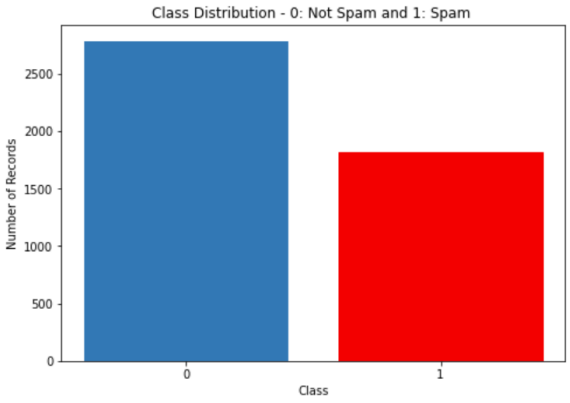

Fig. 2 Class Distribution - 0: Not Spam and 1: Spam

The class distribution of the 'Spambase data set' is represented in figure 2. The number of records in class 0 (non-spam) and class 1 (spam) can be observed as approximately 2900 and 1700, respectively. There is not much difference in class distribution hence oversampling or under-sampling data is not required. The total number of records is 3500.

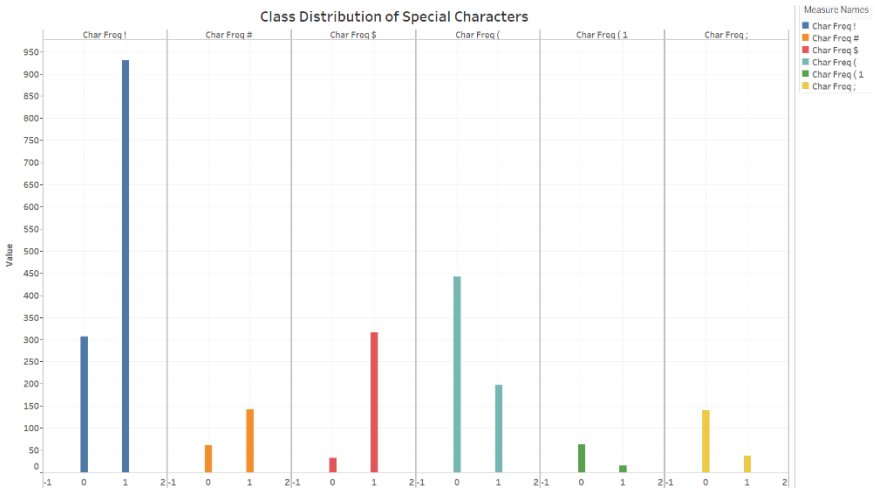

Fig. 3 Class Distribution of Special Characters

The class distribution of the special characters (*!, #, $, (, (1, ;*) can be seen in figure 3. Maximum number of records of special characters in class 1 belong to 'Char Freq !' and in class 0 belong to 'Char Freq ('. Minimum number of records of special characters is class 1 belong to 'Char Freq (1' and in class 0 to 'Char Freq $'. The total number of records in class 0 and class 1 is maximum in 'Char Freq !' and minimum in 'Char Freq (1'.

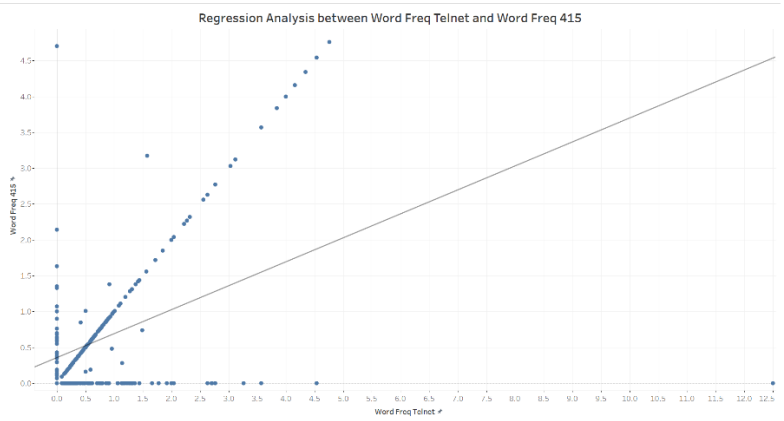

Fig. 4 Regression Analysis between Word Freq Telnet and Word Freq 415

Regression analysis between the columns 'Word Freq 415' and 'Word Freq Telnet' is shown in figure 4. A trend line has been fitted between the features to understand the relation between them. It can be noticed that as the frequency of the word telnet is increasing with the increase in the frequency of 415. Hence, it can be concluded from the figure that the two features are positively correlated.

5. Kafka

Kafka is streaming Hadoop architecture used to replicate a real-time streaming application. Kafka has a producer, consumer, and a topic which are managed by a Zookeeper. Zookeeper also maintains synchronization between the servers and manages cluster nodes. There is an element is Kafka similar to a server known as a broker. A broker helps the producer publish information to the subscribers through topics. A consumer to receive the published information has to subscribe to the topic.

To create a simulation of the real-time environment, the data present in the data source is streamed by the producer to the topic. The consumer registered to this topic extracts the data and uses the data to train and test the Machine Learning models. The architecture of Kafka can be seen in figure 5.

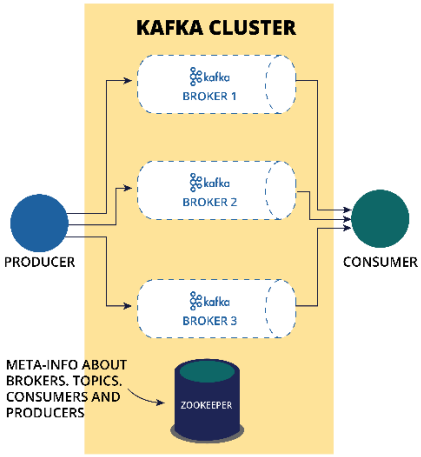

Fig. 5 (Extracted from Pega, 2019)

6. Data Preprocessing

The data is checked for null values if any. 'Spambase data set' **has no null values**. The summary statistics of all the columns is generated and can be seen in figure 6. The figure has only a few columns. A description of the remaining columns can be found in a separate attachment. The feature 'capital_run_length_total' has the maximum mean and standard deviation.

Data Description of a few columns

	count	mean	std	min	25%	50%	75%	max
capital_run_length_total	4601	**283.2893**	**606.3479**	1	**35**	**95**	**266**	**15841**
capital_run_length_longest	4601	52.17279	194.8913	1	6	15	43	9989
capital_run_length_average	4601	5.191515	31.72945	1	1.588	2.276	3.706	1102.5
word_freq_you	4601	1.6621	1.775481	0	0	1.31	2.64	18.75
word_freq_your	4601	0.809761	1.20081	0	0	0.22	1.27	11.11
word_freq_george	4601	0.767305	3.367292	0	0	0	0	33.33
word_freq_hp	4601	0.549504	1.671349	0	0	0	0	20.83
word_freq_will	4601	0.541702	0.861698	0	0	0.1	0.8	9.67
spam_or_not_spam	4601	0.394045	0.488698	0	0	0	1	1
word_freq_our	4601	0.312223	0.672513	0	0	0	0.38	10
word_freq_re	4601	0.301224	1.011687	0	0	0	0.11	21.42
word_freq_all	4601	0.280656	0.504143	0	0	0	0.42	5.1
char_freq_!	4601	0.269071	0.815672	0	0	0	0.315	32.478
word_freq_hpl	4601	0.265384	0.886955	0	0	0	0	16.66
word_freq_free	4601	0.248848	0.825792	0	0	0	0.1	20
word_freq_mail	4601	0.239413	0.644755	0	0	0	0.16	18.18

Fig. 6

6.1 Correlation for feature reduction:

Correlation coefficients among variables can are represented using a correlation matrix. Advanced analysis is employed to summarize the correlation matrix (Displayr, 2020). A correlation plot for the 57 features is plotted to identify the correlation among the features. Figure 7 shows the correlation plot. Highly correlated features (whose Pearson's correlation coefficient value is greater than 0.845) are

removed as they do not add value to the data. The features ‘word_freq_415’ and ‘word_freq_857’ are removed from the dataset as they have a correlation value of 0.99 with the feature ‘word_freq_direct’.

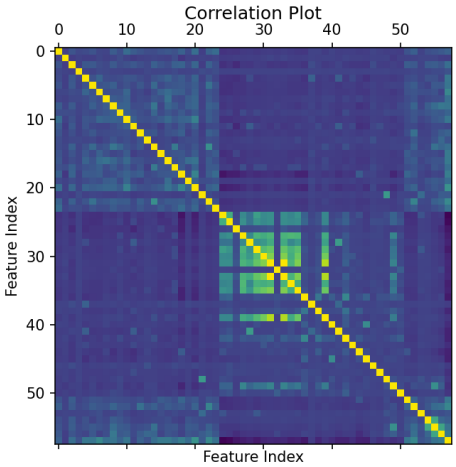

Fig. 7. Correlation Plot of Features in the Dataset

6.2 Normalization

Data normalization is a process where the data is organized in a way that can be used for queries and analysis. The dataset has columns that are not in the same magnitude or format. For example, the columns ‘capital_run_length_average’, ‘capital_run_length_longest’,’ capital_run_length_total’ are more than 1 whereas all the remaining columns are between 0 and 1. To balance the weight or importance given in making a prediction each column is scaled from 0 to 1 such that the minimum value is mapped to 0 and the maximum value to 1 and all other values are mapped to 0 to 1. The formula to normalize the features can be seen in fig. 8.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Fig. 8 Mathematical formula for Normalization

7. Split data into Train and Test Set

By using the train_test_split function the entire dataset is split into a training set and a testing set. 80% of the records make the training set and rest 20% constitute the test set.

8. Architecture

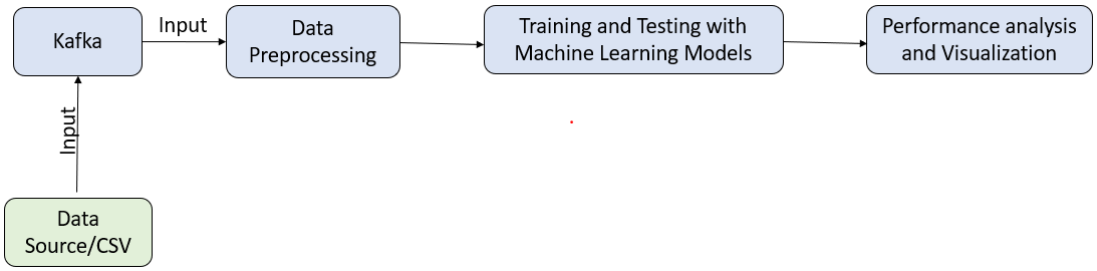

Fig. 9 Project Architectute

8.1 Data Source/CSV - The file or database or stream where the relevant data is stored is known as the data source. Here, the data source is a CSV file containing the frequencies of occurrence of various words in a spam email. This is given as input to Kafka.

8.2 Kafka - Kafka is used for streaming the data for real-time analysis. Kafka producer feeds the data from the data source (here CSV file) to the Kafka consumer through a unique topic. The Kafka server, Kafka producer and consumer are managed by the zookeeper. The Kafka consumer is integrated with the analytics software.

8.3 Data Preprocessing - As part of data preprocessing null values in the dataset have been removed. The maximum values of the mean and standard deviation of the features are identified. A correlation plot has been plotted to detect highly correlated features. Features with correlation values greater than 0.845 are removed. Data has been normalized to balance the weights by mapping the maximum value to 1 and minimum value to 0, for a column.

8.4 Training and Testing with Machine Learning Models - The models Random Forest, Support Vector Machines, and Neural Networks are trained and tested on data. An ensembling technique was developed that uses the outputs to maximize the classification performance.

8.5 Performance Analysis and Visualization - The performance metrics discussed in the paper were represented using the appropriate visualization tools.

9. Data Analytics Algorithms

9.1 Random Forest - Random forest is a supervised learning algorithm that can be used for classification and regression. In random forests, decision trees are created on datasets that are randomly sampled. These decision trees provide outputs and the best solution is selected through voting (DataCamp, n.d.).

9.2 Support Vector Machine - A hyperplane is used to define a Support Vector Machine. When the model is trained on supervised data, an optimal hyperplane is produced as the algorithm's output. The hyperplane is used to categorize/predict the inputs into their respective classes (Geeksforgeeks, 2020).

9.3 T-SNE + Support Vector Machines - T-SNE stands for t-distributed stochastic neighbor embedding. t-SNE is used to embed high dimensional data into a low dimensional data for visualization. This is done to effectively represent the high dimensional data. The high dimensional data is transformed into a two or three-dimensional data by modeling similar objects to close by points and dissimilar objects to distant

points (Wikipedia, 2020). Here, the reduced dimensions were given as input to the Support Vector Machine for classification.

9.4 Neural Network - A series of algorithms that try to imitate the activity of a human brain to identify relationships among data is known as a Neural Network. It has several layers of interconnected nodes where every node is a perceptron. The output generated by these nodes is given as input to an activation function which may or may not be linear (Chen, 2020).

9.5 Ensemble Learning - The results of Machine Learning models can be improved significantly by combining several models using Ensemble technique. As several models are combined the results of Ensembled technique are better than the results of any individual model. They are used to decrease variance, bias, and improve predictions (Smolyakov, 2019).

10. Descriptive Analytics

Summarizing the central tendency, dispersion, and shape of a dataset is known as Descriptive analytics. The description of each column is obtained by using the data.describe() function. Each column's mean, standard deviation, minimum value, the median, and the maximum value is obtained. The column capital_run_length_total has the highest mean, standard deviation, and maximum value. The data was tested and confirmed that it contains no null values.

11. Inferential Analytics

Statistical models are used to compare the sample data to the previous research or to the other samples. It is majorly used to test the logic of the hypothesis. Chi-square test of independence is performed between a feature and the class namely word_freq_money and spam_or_not_spam. The null hypothesis is that the feature and class are independent, and the alternate hypothesis is that they are dependent. After performing the Chi-square test of independence, the p-value is obtained is less 0.05. Hence, the null hypothesis is rejected, and the alternate hypothesis is accepted. This indicates that the feature and class are dependent. Similarly, the same test was also performed for word_freq_technology and spam_or_not_spam. The p-value obtained is less than 0.05, hence the null hypothesis was rejected, and the alternate hypothesis is accepted concluding that they are dependent.

12. Description of the application of Machine Learning algorithms for Data Analysis:

12.1 Random Forest

Random Forest is usually used for classification or regression tasks. Here, Random Forest is used in a classification task, that is, to predict if an email is a spam or not. Random Forest has been applied because of its powerful ensemble technique applied among the generated decision trees.

K-fold Cross-Validation

The performance of a model on a new dataset can be measured using k-fold cross-validation. Here, K-fold validation of 4-splits is performed on the training data.

Hyper-parameter tuning

In Machine Learning models, we always try to obtain the most optimal solution, which can be achieved by exploring various possibilities also called the parameters. The parameters that define the architecture of a model are known as hyper-parameters. The most optimal solution is obtained only when highly suitable hyper-parameters are used. This process of obtaining the best model is known as hyper-parameter tuning.

An operation known as Grid search is performed to obtain the best-suited hyper-parameters that gives the optimal result for the training set. The hyper-parameters for the Random Forest obtained from Grid Search is shown in figure 10.

```
Model Summary
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',
                       max_depth=8, max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=450,
                       n_jobs=None, oob_score=False, random_state=42, verbose=0,
                       warm_start=False)
```

Fig. 10 Definition of Random Forest Classifier

These hyper-parameters are given as inputs to the Random Forest classifier. Cross validation is performed in which the model is trained on k-1 folds and it is tested on 1-fold. By using these hyper-parameters average accuracy, average precision, average recall, and average F1 score can be obtained as shown in figure 11.

```
Avearge accuracy_RF 93.5054347826087
Average_precision_RF 0.9362759779413182
Average_recall_RF 0.9273520460513138
Average F-score 0.9311959829549048
```

Fig. 11 Results of Random Forest on K-fold cross-validation

After cross-validation, this classifier is trained on the complete training set and the model is saved. The saved model is used to make predictions based on the given input test values. The accuracy of the classifier can be obtained by using these predictions. The classification_report function can be used to obtain the classification report of the model depicted in figure 12.

```
Classification Report for Random Forest
              precision    recall  f1-score   support

         0.0       0.93      0.97      0.95       556
         1.0       0.96      0.88      0.92       365

    accuracy                           0.94       921
   macro avg       0.94      0.93      0.93       921
weighted avg       0.94      0.94      0.94       921
```

Fig. 12 Classification Report for Random Forest

12.2 Support Vector Machine

Support Vector Machines can be used to perform either classification or regression. Here, SVM is used to perform a classification task by predicting if an email is a spam or not spam. SVM is considered as one of the best classification techniques needed to model non-linear data and hence was applied to the current data.

Hyper-parameter tuning and K-fold cross-validation

Grid search is performed to identify the best-suited hyper-parameters for the model. The hyper-parameters obtained for Support Vector Machines are shown in figure 13.

```
SVC(C=10, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=1, kernel='rbf', max_iter=-1,
    probability=False, random_state=None, shrinking=True, tol=0.001,
    verbose=False)
```

Fig. 13 Definition of Support Vector Machines

K-fold cross-validation is performed on the training data with k=4 splits. The obtained hyper-parameters are given as input to the SVM classifier to obtain the average accuracy, average precision, average recall, and average F1 score. Can be seen in figure 14.

```
Average accuracy SVM 93.5054347826087
Average_precision_SVM 0.9362759779413182
Average_recall_SVM 0.9273520460513138
Average f1 score SVM 0.9311959829549048
```

Fig. 14 Results of SVM on K-fold cross-validation

After cross validation, the complete training set is used to train this classifier and the model is saved. The test data set is given as inputs and the predictions are generated as output. The classification metrics of the model can be obtained by using the classification_report function. Figure 15 depicts the classification metrics of SVM.

Classification Report for Support Vector Machine

	precision	recall	f1-score	support
0.0	0.93	0.96	0.95	541
1.0	0.94	0.90	0.92	380
accuracy			0.94	921
macro avg	0.94	0.93	0.93	921
weighted avg	0.94	0.94	0.94	921

Fig. 15 Classification Report for Support Vector Machine

12.3 T-SNE + Support Vector Machines

T-SNE was applied over the dataset and its dimensions from 55 have been reduced to three. This was performed to check whether the models would be able to perform better or on par with models trained and tested on the original data. If successful, the dimensions of the data can be reduced from 55 to just 3 features and still preserve its value.

Additionally, t-SNE helps us visualize the dataset by bringing the dimensions from 55 to three. Figure 16 depicts the dataset when visualized in three dimensions.

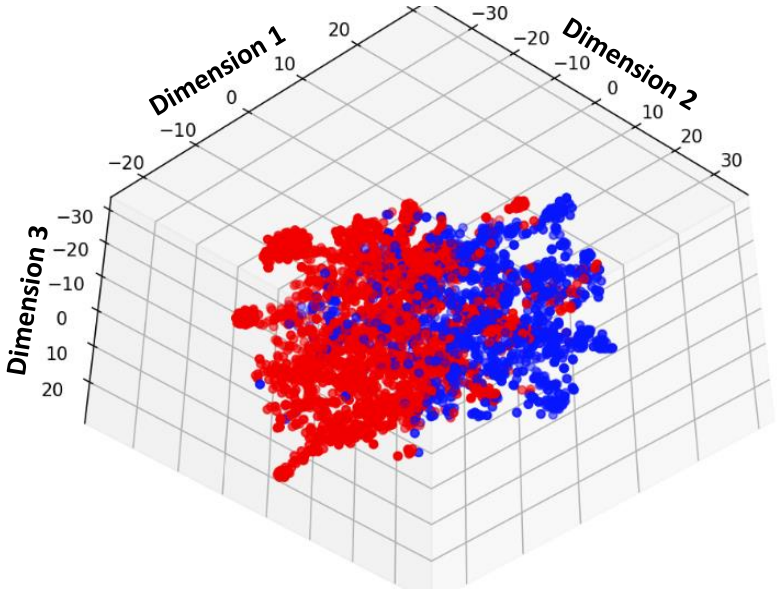

Fig. 16 Scatter plot of data in Reduced Dimensions

The red markers represent the 'spam' class whereas the blue markers represent the 'non-spam' class. A distinction between the two classes is not very direct but is still present. Hence, an SVM is trained on the newly obtained three-dimensional data to check the classifiers performance. Figure 17 details the classification metrics for this approach.

Classification Report for T-SNE + Support Vector Machine approach

	precision	recall	f1-score	support
0.0	0.61	0.80	0.69	556
1.0	0.41	0.21	0.28	365
accuracy			0.57	921
macro avg	0.51	0.51	0.49	921
weighted avg	0.53	0.57	0.53	921

Fig. 17 Classification Report for T-SNE + Support Vector Machine approach

Further comments on the classifier's performance will be made in the Results and Discussion section.

12.4 Neural Network

Neural networks have revolutionized many domains since their inception. A neural network with five layers was applied over the dataset. The hyper-parameter tuning was performed over 5 cross validations and the best parameters are selected. The first, second, third and fourth layer consists of 150, 100, 100 and 100 neurons, respectively. The last layer consists of one neuron with a sigmoid activation function since this is a binary classification task. The loss function is the 'binary crossentropy', the optimization function is 'adam', the activation function for all the layers except the last layer is 'relu'. The number of epochs are 100 and the batch size is 8. The summary of the model is shown in figure 18.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 150)	8400
dense_2 (Dense)	(None, 100)	15100
dense_3 (Dense)	(None, 100)	10100
dense_4 (Dense)	(None, 100)	10100
dense_5 (Dense)	(None, 1)	101

Total params: 43,801
Trainable params: 43,801
Non-trainable params: 0

Fig. 18 Neural Network model summary

The model was tested on the test set. The classification report is shown in Fig 19.

Classification Report for Neural Network

	precision	recall	f1-score	support
0.0	0.96	0.96	0.96	552
1.0	0.95	0.95	0.95	369
accuracy			0.96	921
macro avg	0.95	0.95	0.95	921
weighted avg	0.96	0.96	0.96	921

Fig. 19 Classification Report Neural Network

12.5 Ensemble

It can be observed that the models performed on par with each other. In order to maximize the prediction accuracy and other classification metrics, an ensemble approach has been designed among the trained Random Forest, SVM and Neural Network models. A voting mechanism is established between the respective predictions of the models and the class that was predicted the highest number of times between the three is assigned as the final label for the classifier. The classification report of the ensemble approach is shown in figure 20.

Classification Report for Ensemble

	precision	recall	f1-score	support
0.0	0.95	0.98	0.96	563
1.0	0.96	0.93	0.94	358
accuracy			0.96	921
macro avg	0.96	0.95	0.95	921
weighted avg	0.96	0.96	0.96	921

Fig. 20 Classification metrics for Neural Network Model

13. Results and Discussion

The results of various algorithms applied on the data is shown in the Table II. The classification results of SVM trained over reduced dimensions using t-SNE has not performed well and has ranked lower than all the models. This is explainable as data has been lost when reducing the number of features from 55 to 3. So, t-SNE was not able to capture all the properties of data points in the reduced dimensions.

With respect to other models, it can be noted that the accuracies of the models Random Forest, SVM and Neural Network are 95%, 95% and 96%. The F1-Score, which is the harmonic means of precision and recall is highest for the neural network model and the ensemble model. The precision value for the class 1 is highest for the Random Forest classifier. The recall value for the class 0 is also the highest for the Random Forest model. So, the best model with the given metrics is highly subjective and depends upon the user requirements. For example, if the select model must have a high precision in detecting the spam emails then Random Forest is favored. But if the selected model needs to have a high F1-score, then the neural model or the ensemble model is preferred. However, it must be noted that the performance of the classifiers is very close, and these metrics are very sensitive to the test split. Hence, the best technique would be to rely over the ensemble technique and take advantage of all the trained models instead of relying over a single model.

14. Conclusions

The current project focuses on the application of Machine Learning algorithms for spam email classification. Various existing approaches such as Random Forest, Support Vector Machines along with new approaches such as Neural Networks and SVM on dimensionally reduced data. All the relevant details associated with data pre-processing, data schema, system architecture, data analytics algorithms, data visualization, data analysis, advanced ML analysis, software/hardware details have been described in detail. The results were analyzed, and comments were provided regarding model selection.

The objectives of the project include,

- Learning how data can be directed to the Machine Learning model using data engineering.
- Application of Kafka to stream data in Producer to Consumer paradigm.
- Integration of Kafka with analytics program.
- Application of Machine Learning algorithms for classification techniques.

Models	Accuracy	Precision		Recall		F1-score	
Classes		**0**	**1**	**0**	**1**	**0**	**1**
Random Forest	0.94	0.92	**0.97**	**0.98**	0.88	0.95	0.92
SVM	0.94	0.93	0.94	0.96	0.90	0.95	0.92
SVM+T-SNE	0.57	0.61	0.41	0.80	0.21	0.69	0.28
Neural Network	**0.96**	**0.95**	0.95	0.97	**0.93**	**0.96**	**0.94**
Ensemble	**0.96**	0.94	0.96	0.97	0.92	**0.96**	**0.94**

Table II. Performance Metrics for the developed models

- Using visualization software like Matplotlib and Tableau to effectively convey analysis and results.
- Effectively understand and apply Data Engineering and Data Analysis methods.

From the course, we learned about

- Big Data
- Data Science
- Entity Relationship Models
- SQL and Advanced SQL
- NoSQL databases
- Descriptive Analytics
- Inferential Analytics
- Data Pre-processing
- Hadoop and its elements
- Exploratory Data Analysis
- Machine Learning
- Big Data Security
- Cloud and Internet of Things

15. Hardware and Software used:

Software: Windows Operating System, Kafka, Python, Tensorflow, Keras, Sklearn, Pandas, Numpy, Matplotlib, Jupyter Lab, Anaconda, Tableau.
Hardware: Intel Core i7 Processor, 8GB DDR4 RAM

References

Awad, W. (2011). Machine Learning Methods for Spam E-Mail Classification. International Journal of Computer Science and Information Technology, 3(1), 173-184. doi:10.5121/ijcsit.2011.3112

Chen, J. (2020, January 29). Neural Network Definition. Retrieved May 14, 2020, from https://www.investopedia.com/terms/n/neuralnetwork.asp

Cisco, (2020, April 22). What is Spam Email? Retrieved May 14, 2020, from https://www.cisco.com/c/en/us/products/security/email-security/what-is-spam.html

DataCamp. (n.d.). Random Forests Classifiers in Python. (n.d.). Retrieved May 14, 2020, from https://www.datacamp.com/community/tutorials/random-forests-classifier-python

Displayr. (2020, April 23). What is a Correlation Matrix? Retrieved May 14, 2020, from https://www.displayr.com/what-is-a-correlation-matrix/

Geeksforgeeks .(2020, April 04).Classifying data using Support Vector Machines(SVMs) in Python. Retrieved May 14, 2020, from https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/

Jayalakshmi, A., & Kishore, K. K. (2018). Performance evaluation of DNN with other machine learning techniques in a cluster using Apache Spark and MLlib. Journal of King Saud University - Computer and Information Sciences. doi:10.1016/j.jksuci.2018.09.022

Lee, S. M., Kim, D. S., Kim, J. H., & Park, J. S. (2010). Spam Detection Using Feature Selection and Parameters Optimization. 2010 International Conference on Complex, Intelligent and Software Intensive Systems. doi:10.1109/cisis.2010.116

Pega. (2019, February 21).Kafka standard deployment. Retrieved May 14, 2020, from https://community.pega.com/knowledgebase/articles/decision-management-overview/kafka-standard-deployment

Priyaa, D. S., Kumar, R. N., & Banuroopa, K. (2010). Improvising BayesNet Classifier Using Various Feature Reduction Method for Spam Classification 1.

Runbox, (2019, June 18). What is spam, and how to avoid it. Retrieved May 14, 2020, from https://runbox.com/email-school/what-is-spam-and-how-to-avoid-it/

Rusland, N. F., Wahid, N., Kasim, S., & Hafit, H. (2017). Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets. IOP Conference Series: Materials Science and Engineering, 226, 012091. doi:10.1088/1757-899x/226/1/012091

Security-faqs. (n.d.). What Is Spam, What Are It's Effects And How Do I Prevent It? Retrieved May 14, 2020, from http://www.security-faqs.com/what-is-spam-what-are-its-effects-and-how-do-you-prevent-it.html

Sharaff, A., Nagwani, N. K., & Dhadse, A. (2015). Comparative Study of Classification Algorithms for Spam Email Detection. Emerging Research in Computing, Information, Communication and Applications, 237-244. doi:10.1007/978-81-322-2553-9_23

Shilton, A., & Lai, D. T. (2007). Iterative Fuzzy Support Vector Machine Classification. 2007 IEEE International Fuzzy Systems Conference. doi:10.1109/fuzzy.2007.4295570

Smolyakov, V. (2019, March 07). Ensemble Learning to Improve Machine Learning Results. Retrieved May 14, 2020, from https://blog.statsbot.co/ensemble-learning-d1dcd548e936

UCI Machine Learning Repository. (1999). Spambase Data Set [Data file]. Retrieved from http://archive.ics.uci.edu/ml/datasets/Spambase/

Wikipedia, (2020, May 13)T-distributed stochastic neighbor embedding. Retrieved May 14, 2020, from https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding