Uncertainty Estimation in Machine Learning Based Classification Models

By

Shilpi Majumder

Registration No - 154158 of 2020-2021

Examination Roll No - M4CSE22034

Thesis submitted in partial fulfilment of the requirement for the Degree of M.E. Computer Science and Engineering

In the Department of Computer Science and Engineering

Under the supervision of

Dr.Sanjoy Kumar Saha

Jadavpur University

Kolkata- 700032

2020-22

Certificate from the Supervisor

This is to certify that the thesis entittled "Uncertainty Estimation in Machine Learning Based Classification Models" has been satisfactorily completed by Shilpi Majumder(Registration No - 154158 of 2020-2021; Examination Roll No- M4CSE22034). It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for the awarding of the Master of Engineering in Computer Science and Engineering degree of the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2021-22.

Dr.Sanjoy Kumar Saha

Department of Computer Science and Engineering

Jadavpur University

(Supervisor)

Prof. Anupam Sinha

Head of the Department

Department of Computer Science and Engineering

Jadavpur University

Prof. Chandan Mazumdar

Dean, Faculty of Engineering and Technology

Jadavpur University

Certificate of Approval

This is to certify that the thesis entittled "Uncertainty Estimation in Machine Learning Based Classification Models" has been satisfactorily completed by Shilpi Majumder (Registration No - 154158 of 2020-2021; Examination Roll No- M4CSE22034) in partial fulfilment of the requirements for the award of the degree of Master of Engineering in Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made,opinion expressed or conclusion drawn therein but approve the thesis only for the purpose of which it has been submitted.

_____

(Signature of The Examiner)

_____

(Signature of The Supervisor)

Declaration of Authorship

I hereby declare that the thesis entitled "Uncertainty Estimation in Machine Learning Based Classification Models" contains the work presented in it are my own research in my degree of Master of Engineering in Computer Science and Technology in the Department of Computer Science and Engineering, Jadavpur University. All information have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

_____

Signature of the candidate

Acknowledgements

I would like to extend my heartfelt gratitude to people who helped to bring this thesis work to complete. First, I would express my deep and sincere gratitude and appreciation to my supervisor Dr. Sanjoy Kumar Saha for providing me all the facilities and support to the activities of this research without whose continuous support and encouragement this work would not have been possible. His assistance, valuable suggestions and personal guidance throughout the duration of the project has played a pivotal role. Without his enthusiasm, encouragement, support and continuous optimism this thesis would hardly have been continued.

I am thankful to my parents who have always been my constant source of support and inspiration. No words for them to express gratitude, love and respect towards them. I would also like to thank all my teachers for sharing their knowledge and valuable thoughts. Last, but not the least, I would like to thank all my friends for their valuable suggestions and helpful discussions.

Regards,

Shilpi Majumder

University Registration No.: 154158 of 2020-2021

Department of Computer Science and Engineering

Jadavpur University

Contents

Abstract

Machine learning has demonstrated excellence at medical image analysis tasks like segmentation and classification for diagnosis of autonomous driving, cyber bullying detection, Twitter's sentiment analysis, from music generation to augmenting or replacing radiologists to identify cancer in Computed Tomography scans. Deep learning is a component of many industrial and clinical solutions. Despite their success, these methods do not consider the output quality and are simply concerned with improving point forecast accuracy and for that it is essential to understand how trustworthy a forecast is. Deep learning algorithms are becoming simpler to employ as new ones are developed, but uncertainty estimates still poses a serious challenge for safety-critical applications.

The ability to estimate uncertainty not only enables us to assess the dependability of a system's decision, but it also enables us to take on nondeterministic tasks with multiple potential outcomes, such as future prediction, which, when fully characterised, is a crucial component of human intelligence.

This thesis starts with a general presentation of the uncertainty principle and apply them in Twitter US Airline Sentiment data to analyze how travelers in February 2015 expressed their feelings on Twitter. First, I used different text classification models with predicting uncertainty in respective to every single model . Then I compare these classification techniques based on the results, to find which machine learning algorithms appears to be superior to the other algorithms for this Twitter aircraft sentiment analysis data-set provided by the United States.

Finally, I choose the best models (models with lowest uncertainty) and ensemble them to see if we can improved our model accuracy of prediction using uncertainty as a parameter.

Chapter 1

Introduction

Recently, Machine learning has excelled in practical sciences including biology, physics, chemistry, engineering, medical diagnosis and text classification. They are still largely underutilised in mission- and safety-critical real-world applications as they are typically brought on by preexisting ambiguity in the data or a lack of neural network expertise. Uncertain forecasts must be avoided or given to human specialists in order to distinguish between in-domain and out-of-domain samples, occurrence of overconfident predictions and lack of expressiveness and transparency in the inference model.

For neural networks, estimating uncertainty is more feasible to determining how sure a neural network is ,in its predictions and the aim of uncertainty estimate in the learning models. Individuals see the hazard in their choices, and they utilise this knowledge to better themselves.Estimating uncertainty is important not only in high-risk industries like remote sensing but also in areas where data sources are very irregular and labelled data is in short supply [6].

In computer-based healthcare treatments, a poor choice might have devastating repercussions, especially in life-threatening situations. It's crucial to be able to tell whether a model is confident in its results when analysing data, and it's also necessary to be able to decide whether you want to utilise more varied data. or ought the model to be modified? or should I be cautious while choosing a course of action? Since the majority of machine learning models are thought of as deterministic operations, they operate in a completely different setting than probabilistic models that also take uncertainty into account. When choosing between two likely outcomes without any prior knowledge, a categorization model is often needed.

It is crucially wanted to represent uncertainty in any AI-based system in a trustworthy way. In AI situations like AL and concrete learning algorithms, uncertainty principles play a crucial role.

It has long been difficult to get reliable uncertainty estimates for predictions made by learning models.

In this study, we examine the effectiveness of Logistic Regression, Decision tree, Support Vector Machine, KNN, Gaussian Naive Bayes and Random Forest as a classifier in classification model of Twitter US Airline Sentiment Analysis where it assesses whether the emotion expressed in the set of tweets for six US airlines was good, neutral, or negative. By assessing the uncertainty associated with each classifier's predictions, we compare the outputs of several classifiers and study how uncertainty is used to improved the prediction accuracy [1].

Chapter 2

Uncertainty Principle

In machine learning, noise in the data, inadequate domain coverage, and flawed models are the three primary sources of uncertainty.
Uncertainty sources appear when test and training data are unequal, whereas data uncertainty appears as a result of class overlap or data noise; nonetheless, assessing knowledge uncertainty is far more challenging than computing data uncertainty [11].

Like people, a machine learning model may indicate how confident it is in its predictions. It is customary to make a distinction between epistemic and aleatoric uncertainty when addressing the uncertainty principle. In order to further develop deep learning networks, especially to accomplish continuous learning using deep learning, it is thus important to employ uncertainty estimations.

Aleatoric and epistemic uncertainties are the two basic categories of uncertainty.

Aleatoric uncertainty is an unavoidable flaw in data that makes predictions uncertain (also known as data uncertainty). This refers to the data's inherent uncertainty like consider this noise to be statistical or sensory as there is always have some underlying noise in the data collecting process, regardless of how much data you get. It's built into the data itself. This kind of uncertainty cannot be reduced since it is a characteristic of the data distribution and not a feature of the model. Noise in the observations provides a description of the input-dependent uncertainty. This type of uncertainty is caused by hidden factors or measurement errors, and it cannot be eliminated by accumulating additional data. When our data is noisy, it is greatest. Changing your sensor and obtaining more precise data is the only method to lower aleatoric uncertainty.

Model uncertainty or epistemic uncertainty are terms used to describe the uncertainty in model parameters. We may utilise these estimates to understand when the model cannot give a trustworthy response by

referring to epistemic uncertainty, also known as systematic uncertainty, which can characterise the models confidence in the forecast. It happens because there isn't enough data for the model to be sufficiently trained to infer the underlying data-generating function. Because of this, the amount of training instances is inversely correlated with epistemic uncertainty, which may be decreased by collecting and training on more data.

Probability distributions over model inputs or parameters are used to represent uncertainties. The prior distribution of the model's weights W, which depicts how much they fluctuate given a certain set of data, is supplied, and epistemic uncertainty is simulated.

These methods used for evaluating uncertainty in a variety of models, from basic Bayesian models to neural networks. Numerous methods for estimating uncertainty have been developed for networks and are based on them [10].

Most deep learning networks generate probability distributions that can generally capture the data.

Given training inputs $x = x_1, x_2, ..., x_n$ and their corresponding output labels $y = y_1, y_2, ..., y_n$ derived from an undefined probability distribution,using Bayesian neural networks provide a principled mathematical framework for this kind of uncertainty by estimate the weight distribution over the space of parameters (θ). Here $d = (x_1, y_1), (x_2, y_2), ...(x_n, y_n)$. The posterior distribution can represent as, $p(\theta|d) = \frac{p(d|\theta)p(\theta)}{p(d)}$
$p(d|\theta)$ is the likelihood of the training data given a parameter θ.Assuming each training sample is independent to each other,we need to find the product of likelihood of each individual training sample.

$$p(d|\theta) = \sum_{n=1}^{N} P(y^n|x^n, \theta) \tag{2.1}$$

This likelihood value will calculated differently for both classification and regression problem.

2.1 For classification(in discrete domain)

For classification problem ,we need to use a special activation function softmax() that will convert all the score values into normalize probability distribution. Using softmax we can convert any arbitrary real numbers into probability scores and we get two constraints on our output.
Using Softmax(),we get-

$$P(d|\theta) = \frac{e(f_d{}^{\theta})}{\sum_d^D e(f_d{}^{\theta})} \tag{2.2}$$

1) Every output of probabilities has to be bigger than zero.
2)We normalised the sum of all of our class probabilities to 1.
The likelihood distribution $p(d|\theta)$—the probabilistic model by which the inputs generate the outputs given

some parameter setting θ.

We could then define this unique loss, known as the cross-entropy loss, that allowed us to optimise our distribution learning by minimising the negative log-likelihood of the predicted distribution to match the ground truth category distribution after receiving an output of class probabilities from this softmax() activation function.

$$cross - entropy = \sum_{i=1}^{n} y_i * logP_i \tag{2.3}$$

Our target class label y is taken from some likelihood function, in this case a categorical distribution specified by distributional parameter P, which truly defines our distribution and likelihood over that anticipated level. Specifically the probability of the answer being in the ith class is exactly equal to the ith probability parameter. $y = categorical(P)$, Here, labels are derived from a categorical distribution's underlying likelihood function. Additionally, a set of distributional parameters define each of these probability functions, $P = P_1, P_2, ...P_i$ as these probabilities define the categorical distribution, where $\sum_{i=1}^{n} P_i = 1$ and $P_i > 0$.

2.2 For continuous class target(Regression)

Here instead of using class probability we are using a probability distribution over the entire real number like the classification domain. Here we can output the parameters of our distribution namely mean(μ) and the standard deviation or variance of the distribution(σ^2), since the support of this output is continuous and infinite . Constrain: $\mu \in R$ and $\sigma > 0$.
As mean is unbounded so no need to constraint it and standard deviation must be strictly positive, so that we can use an exponential activation function to enforce that constraint.
$y \approx Normal(\mu, \sigma^2)$
$Negativeloglikelihood = -log(N(y|\mu, \sigma^2))$

Use this loss function to learn not a point estimate of the label y but a full distribution or gaussian around describing that data likelihood.

Given a dataset $d = (x_1, y_1), (x_2, y_2), ...(x_n, y_n)$, the posterior distribution over the space of parameters(θ) by using Bayes'theorem:

$$p(\theta|d) = \frac{P(d|\theta) * P(\theta)}{P(d)} \tag{2.4}$$

In the case of infinitely many weights, a Gaussian process may be constructed given a neural network, however with a fixed number of weights, Bayesian neural networks can still be used to provide model uncertainty [4].

Epistemic uncertainty, which represents the uncertainty inherent in the modelling forecasting process itself, is far more difficult to assess.

2.3 Uncertainty calculation

Epistemic uncertainty is modelled and the weights of the model are given a prior distribution that depicts how much they fluctuate given specific data. Bayesian neural networks offer a methodical mathematical framework for dealing with this sort of uncertainty. The output of the network uses Bayesian inference to construct a posterior over the weights for supplied training data. This posterior is used to calculate the predictive distribution of a test sample and to express the prediction distribution over that particular model.
$p(y|x, \theta) = \int p(y|x, \theta)q(\theta)d\theta$
We use this predictive distribution as a input of softmax() activation function to get the result. The cross entropy value of this result is the epistemic uncertainty in the prediction of output in the model. Epistemic uncertainty indicate to the uncertainty in the model parameter.This can be view as a spread of the posterior probability distribution, in which the flatter posterior distribution reflects higher epistemic uncertainty and peaked posterior distribution reflects lower epistemic uncertainty. And aleatoric uncertainty refers to uncertainty in the input itself.

So, Epistemic Uncertainty(in classification model) = cross-entropy(softmax(predictive distribution of that particular model)).

Total Uncertainty = Epistemic Uncertainty + Aleatoric Uncertainty .

2.4 Importance of uncertainty principle in learning models

Techniques that assess the accuracy of machine learning models include confusion matrices, ROC curves, and F-Scores that produces model's effectiveness.However, it is unclear which datasets the model is misidentifying and why without additional investigation. This is the reason, I have been using multiple classifier on the same dataset for see the difference of their output. Because we can't tell if the model is ineffective on data that represent a certain region of the feature space or how well a model works with unlabeled data?
A person may be able to partially validate the effectiveness of the algorithm in some applications. Manual

evaluation is either essentially impossible in many fields where the model is entrusted with categorising enormous amounts of data. Then measuring the degree to which a model's output may be relied upon when categorising specific samples with specific classifier.The performance of a model on samples categorised after deployment may not correspond to the accuracy it displays during testing. So we are using a point estimate representation to the classifiers' prediction, putting the framework of a classifier's uncertainty into the perspective of probability distributions.

While uncertainty can come from a variety of factors, including the data, model choice, the model parameter, and the decision itself, error describes the discrepancies between predictions and actual observations given a fixed model. Numerous different data models with distinct flaws are feasible as a result of the sources of uncertainty.

2.5 Application of uncertainty principle

There are currently few instances of these methods being successfully applied in clinical practise, despite the remarkable acceleration of deep learning research in healthcare, with potential applications proven across a wide variety of topics [12]. The majority of deep learning-based systems provide deterministic results and do not account for or manage prediction uncertainty, which can undermine confidence in automated diagnosis and interpretation mistakes.

In addition to identifying samples that deviate from the data used to train the model, uncertainty may be used to determine which samples are difficult to categorise and require further expert assessment. The network can still produce (random) predictions with a high degree of confidence even when the distributions of the training and test data are different. The out-of-distribution problem, which is a critical issue in medical applications, refers to how to identify when a model is being used on a domain outside than the training domain. We haven't been able to obtain precise uncertainty estimates for neural network predictions in a very long time.

Estimating epistemic and aleatoric uncertainty has also been addressed for classification and segmentation of medical images. Traditional deep learning techniques for regression and classification do not account for model uncertainty. It is common to mistake the predictive probabilities acquired at the pipeline's end (the softmax output) for model confidence in classification. A model's predictions can be erroneous even with a high softmax output. Passing a point estimate of a function via a softmax results in extrapolations with unrealistically high confidence for locations far from the training data.

Chapter 3

Uncertainty on Learning Models

This section explains the suggested uncertainty-related to the learning models and discusses the approaches employed in this study. Then we can apply uncertainty estimation of these models after calculate the predictive distribution of a test data and to express the prediction distribution over that particular model. Constructing an approximation distribution of the mean (μ) and variance (σ^2) of each layer of the trained model is another efficient way in case of regression problem. Any number of these parameters can be drawn from this distribution during test time for several forward passes .

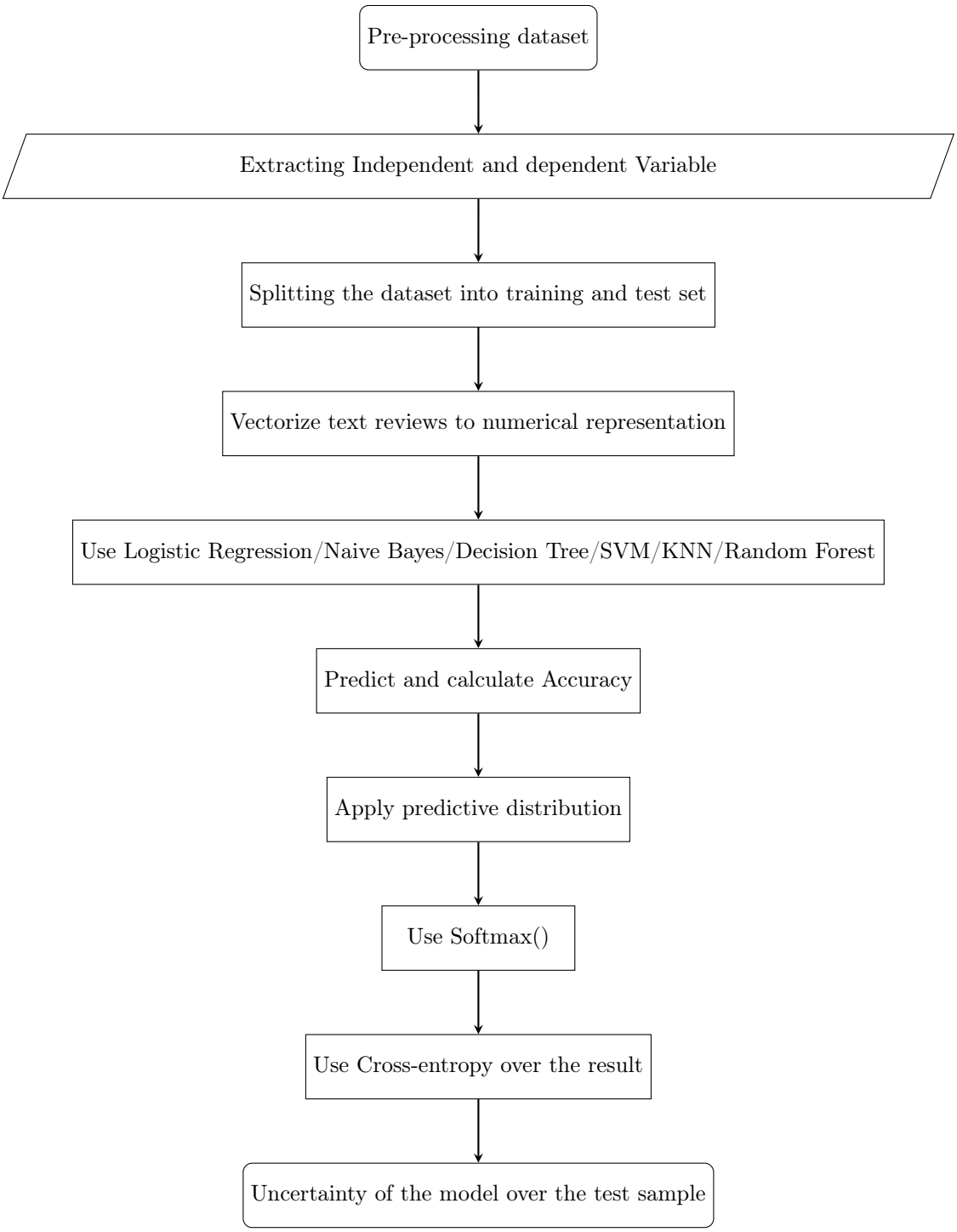
$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.1}$$

and

$$\sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2 \tag{3.2}$$

To approximate the mean and variance, Atanov et al. [4] recommends using a normal and a log normal distribution, respectively. Incase of Classification models we use,
Epistemic Uncertainty = cross-entropy(softmax(predictive distribution of that particular model)).
We go over the specifics pre-trained classifier like Logistic Regression,Naive Bayes, Decision tree, Support Vector Machine, k-nearest neighbors and RandomForest in Sections 3.1, 3.2, 3.3, 3.4, 3.5 and 3.6 before describing how uncertainty is handled in these models.

Flowchart of uncertainty estimation with different classifier



3.1 Logistic Regression

Logistic regression is a member of the supervised machine learning model family in the context of artificial intelligence. It is also regarded as a discriminative model because it makes an effort to distinguish between different classes (or categories).Predictive analytics and categorization frequently employ this kind of statistical model. Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. Given that the result is a probability, the dependent variable's range is 0 to 1. In logistic regression, the probability of success divided by the probability of failure are transformed using the formula -

$$logit(pi) = 1/(1 + \exp(-pi))$$

$$ln(pi/(1 - pi)) = \beta_0 + \beta_1 * x_1 + \ldots + \beta_k * K_k$$

where x is the independent variable, and the dependent variable is logit(pi). And maximum likelihood estimation is used to calculate the beta parameter in this model. This approach evaluates various beta values over a number of iterations to get the best match for the probability value. Logistic regression aims to maximise this function after each of these rounds in order to determine the optimal parameter estimation.It is recommended to assess the model's fit to find out how well it predicts the dependent variable, once the model has been calculated.

3.2 Naive Bayes

Naive Bayes is the most straightforward and quick classification method for large amounts of data where each pair of features being categorised is distinct from the others. Spam filtering, text classification, sentiment analysis, and recommendation systems are just a few of the areas where the Naive Bayes classifier has been successfully used [15]. It makes use of the Bayes probability theory for predictions of unknown classes where it determines the likelihood of an event occurring given the likelihood of an earlier event occurring. The Naive Bayes classification algorithm is a simple yet efficient classification problem in machine learning.Naive Bayes makes the erroneous assumption that each feature or variable belonging to a class contributes equally and independently to the result. The employment of Bayes' theorem with a strong independence assumption between the characteristics forms the basis of naive Bayes classification. Results from the Naive Bayes classification are favourable [7]. We state our issue as follows, given a data matrix A and a goal vector B: $P(B|A) = \frac{P(A|B)*P(B)}{P(A)}$ where n is the sample's total number of features, B is the class variable, and A is a dependent feature vector of dimension n. In order to forecast membership probabilities for each class, such

as the likelihood that a certain record or data point belongs to that class, the Naive Bayes Classifier applies the Bayes theorem. The most likely class is the one with the greatest chance of occurring i.e we only need to identify the result with the greatest likelihood.

$B = argmax_B P(B) * [P(A_1|B) * P(A_2|B) * P(A_3|B) * ... * P(A_n|B)]$

where $A_1, A_2, .., A_n$ are conditionally independent.

$B = argmax_B P(B) \prod_{i=1}^{n} P(A_i|B)$

We are calculating the probability based on the attribute value $A_1, A_2, .., A_n$. The Naive Bayes Classifier presumes that there are no relationships between any of the attributes i.e they are independent. When domain knowledge is there but lack of information, then we can use Naive Bayes.

3.3 Decision Tree

A decision tree, a supervised learning approach, is a tree-structured classifier in which internal nodes stand in for a dataset's characteristics, branches for the decision rules, and individual leaf nodes for the results where the class labels with the data items have been grouped.Tree construction and tree trimming are the two stages of a decision tree. When the tree is recursively partitioned until all the data items have the same class label. Due to the recurrent traversal of the training data set, it is highly laborious and computationally costly. In order to reduce over-fitting and increase the prediction and classification accuracy of the algorithm, tree pruning is carried out from the bottom up manner [9].

The decision tree begins at the root node, the parent node of the tree and the remaining nodes are referred to as the child nodes.. The full dataset is represented, which is then split into two or more homogenous sets. After receiving a leaf node, the tree cannot be further divided; leaf nodes are the ultimate output nodes. In splitting, the decision node or root node is divided into sub-nodes in accordance with the specified requirements.

In decision tree, given a training sample, we try to find the importance of feature by computing entropy (entropy gives the importance of the feature) from the training sample and accordingly the most discriminative capability of the feature is determined. So decision tree is a feature selection process (given a sample with n number of feature, select m number of feature). And to check independence, use Mutual information.

3.4 Support Vector Machine

In this SVM approach, each data point is represented as a point in n-dimensional space (number of features), with each feature's value being the value of a particular coordinate. Then, categorization is achieved by identifying the hyper-plane that clearly separates the two groups [16]. SVM develops the optimum hyperplane repeatedly in order to differentiate between different classes, and then uses it to minimise an error. To divide the dataset into classes as equally as feasible, I must identify a maximum marginal hyperplane (MMH). It establishes a decision boundary between two classes to classify them. Support vectors are the data points that are closest to the hyperplane. These points will help to clarify the separation line by computing margins. A hyperplane is a decision plane that separates a collection of objects into a variety of classes. The distance between the two lines on the class points that are closest to one another is known as a margin. It is calculated how far the line is from the nearest points or support vectors perpendicularly. A larger gap between the classes indicates a big margin; a smaller gap indicates a weaker margin. By translating our data using mathematical operations known as Kernels, such as linear, sigmoid, non-linear, polynomial, and others, SVM creates the hyperplane [2].

Steps to constructing a model-

i) Obtaining the most accurate data for training and testing.

ii) Data vectorization

iii) To train and predict, a Linear SVM Model is created.

A SVM classifier will be built where each unique word in the sentence as well as all consecutive words, will be considered by the classifier.To make this format useful for our SVM classifier, I turn each word into a vector. Each word in the list of all the words found in our training data, which makes up our vocabulary, corresponds to each item in the vector. A word has a value of 1 in the vector if it is present; otherwise, it has a value of 0.

In this method, each data item is plotted as a point in n-dimensional space, and the value of each feature is the value of a specific coordinate (where n is the number of characteristics you have). Then, categorization is achieved by identifying the hyper-plane that clearly separates the two groups.

3.5 k-nearest neighbors

The K-Nearest Neighbor algorithm, which is based on the supervised learning approach, believes that the new data and the existing data are comparable, and it places the new data in the category that the existing categories are most similar to. This implies that it keeps track of all the information that is available and categorises new information based on similarities. As new information emerges, it can then be quickly categorised using the K-NN algorithm into a suitable category [8].

The following stages can be used to describe how the K-NN works: Calculate the Euclidean distance between K neighbours by selecting the Kth neighbour. Pick the K closest neighbours based on the Euclidean distance estimate. Count the number of data points in each category among these k neighbours. Assign the additional data points to the category where the neighbour count is at its highest.

In KNN, K refers how many nearest neighbours you are looking for. How to fix the value of K? If you increase the value of K, the algorithm will take more time to converge. If you use 1NN and if the data point is a noisy one, it will have greate possibility that the classification result be wrong as classification will be done basis of one-neighbours. So 1NN is very risky for noisy dataset, better to use 3NN, 5NN.

3.6 Random Forest

The widely used machine learning method Random Forest is based on the idea of ensemble learning, which is the act of mixing several classifiers to solve a challenging issue and enhance the model's performance. Random Forest is a classifier that uses many decision trees on different subsets of the input dataset and averages the results to increase the dataset's predicted accuracy. Instead than depending on a single decision tree, the random forest uses forecasts from each tree and predicts the result based on the votes of the majority of predictions.All of the trees together will forecast the right output since the random forest mixes many trees to estimate the class of the dataset [3]. When compared to other algorithms, Random Forest can predict results with a high degree of accuracy, function effectively on massive data, and retain accuracy even when a substantial piece of the data is absent.

Chapter 4

Experimental Setup For Twitter US airline sentiment analysis

Twitter is a social media site where users may interact by utilising hashtags to send tweets on topics they've covered in their posts. Data are abundant on Twitter. Understanding what people are talking about, their moods, their opinions on a certain subject or business, and social trends may all be learned through studying tweets. Sentiment analysis is the practise of determining how someone feels or thinks about something based on information like text or pictures. Sentiment analysis aids commercial decision-making.For instance, a company could try to alter a product or stop production entirely if the general public has an unfavourable view of it in order to reduce losses. It's a technique for analysing data and locating sentiment within it. Twitter sentiment analysis is the process of using sentiment analysis to extract user sentiments from data from Twitter (tweets)[14].

This data available at Kaggle is an edited version of the original data. Both a CSV file and a SQLite database are included.

This dataset, which includes tweets categorised as positive, negative, or neutral about US Airlines, is extremely impressive. Based on the justification for the unfavourable feeling, bad tweets are also divided into categories.

4.1 Libraries

For this, I make use of tools from packages like Pandas, Seaborn, Matplotlib, Jupyter Notebook, and NLTK.

4.2 Setup

I'll utilise to look at the tweets' most popular positive and negative terms. An excellent technique to display nlp data is with a wordcloud. The term occurs more frequently in our text data the larger it is in the wordcloud picture. next picture the unfavourable airline-related attitudes [5].

4.2.1 Text Preprocessing

i) Stemming is the process of stripping a word of its suffixes and reducing it to its most basic form, which may represent all of the word's many forms (for example, "read" and "reading" are both reduced to "read").
ii) In English, terms like a, an, the, as, in, on, and so forth are regarded as stop-words, so we can eliminate them in accordance with our needs to minimise vocabulary size.
iii) Change all word capitalization to lower case, reducing the scope of our vocabulary by doing this.
iv) Before analysing text, NLP software separates it into words and phrases.

4.2.2 TF-IDF

A scoring method that is frequently used in information retrieval (IR) or summarization is term frequency — inverse document frequency (TF-IDF). The TF-IDF is intended to demonstrate how significant a phrase is inside a certain document.

A term's weight in a document is simply proportionate to how often it appears.

$$tf(t,d) = \frac{count(t)in(d)}{number(words)in(d)} \tag{4.1}$$

The information density factor (IDF) is the inverse of the document frequency, which assesses the informativeness of word t.

$$IDF(t) = \log \frac{N}{df(t)} \tag{4.2}$$

4.3 Evaluation Metrics Used for learning Model

Evaluation measures such as accuracy, precision,Recall and F1-score were employed to assess the Monte-Carlo DropOut model's performance.

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

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

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

$F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$

The symbols TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively. The mean of the metric scores for each class was utilised to calculate the technique's overall metric score because we applied the same treatment to all classes in this study.

Chapter 5

Result of Using Uncertainty Esimation on Learning Models

These tests aimed to identify which model increases classification accuracy with the least amount of uncertainty.

5.1 Uncertainty Analysis with Evaluation Metrics

Uncertainty Estimation on learning models						
Results	Logistic Regression	Gaussian Naive Bayes	Decision Tree	SVM	KNN	Random Forest
Uncertainty	1.19	1.21	1.19	1.20	1.19	1.21
Accuracy	0.78	0.42	0.67	0.78	0.69	0.76
Precision	0.78	0.66	0.53	0.78	0.70	0.74
Recall	0.79	0.41	0.67	0.78	0.69	0.76
f1-score	0.78	0.44	0.58	0.77	0.70	0.74

From table 5.1 it is evident that using Logistic Regression and SVM model, we get highest accuracy value. However uncertainty present at these models are approx same. Using Logistic Regression, Decision Tree and KNN , we get model uncertainty of 1.19 percent, and using Support vector machine , we get uncertainty of 1.20 percent. These mainly indicating that it offers a more accurate results, whereas Naive Bayes and Random Forest have little higher model uncertainties for the same dataset (1.21 percent, respectively).

According to the observation, more reliable models has low degree of uncertainty give a preferable accuracy value.

Our first round of experiments showed that uncertainty estimates may be used to account for the classifier's confidence. We can observe how the balanced accuracy of the classifier increased when the low uncertainty models were applied. These results suggest that it may be advantageous to use uncertainty measures to pinpoint models in which the classifier is likely to make incorrect predictions.

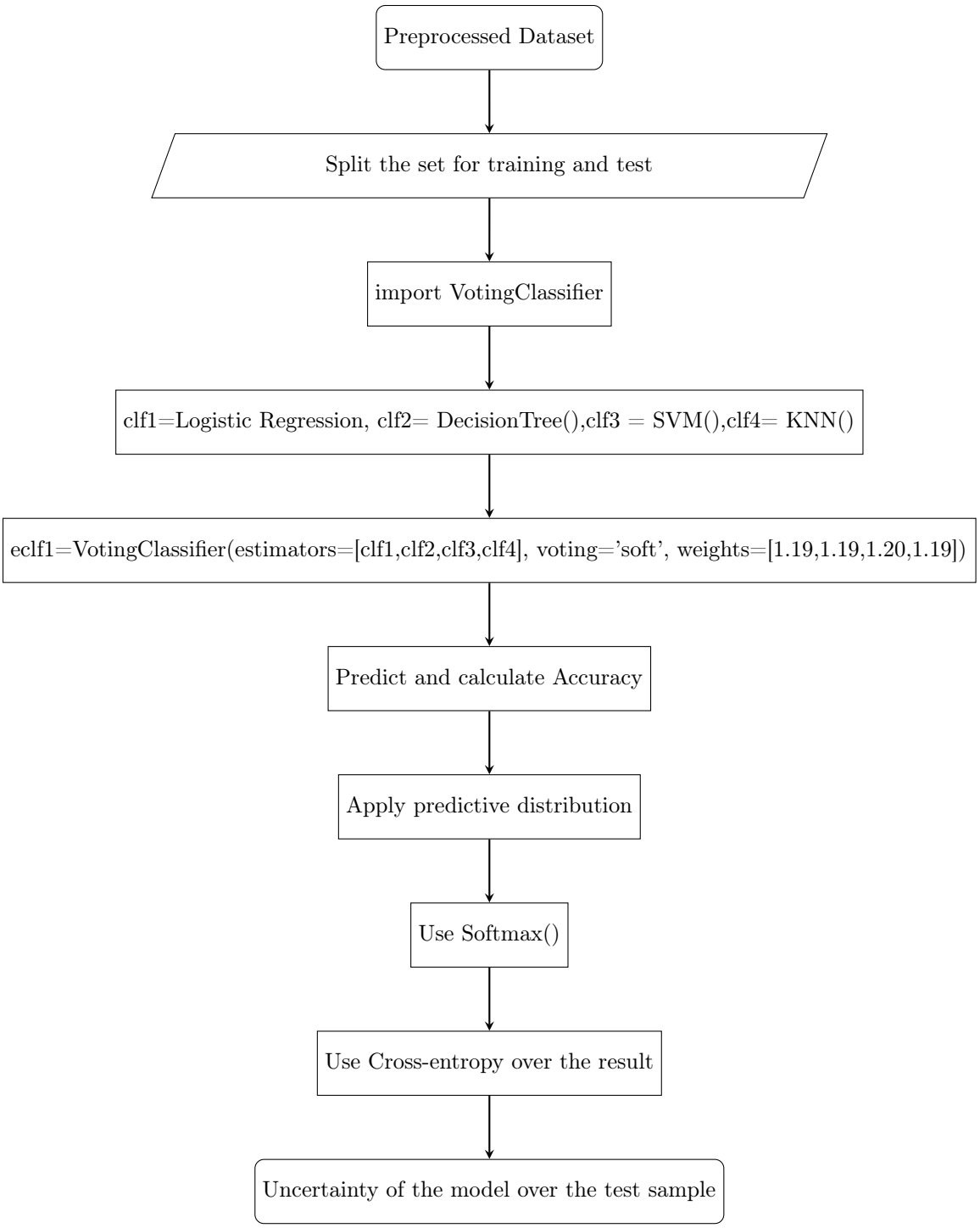
5.1.1 Case 1: Ensemble With Voting Classifier Using Uncertainty as a Parameter

Ensembling is an effective method for enhancing the model's performance by fusing different base models to create an ideal and reliable model. In this thesis, we are implementing voting classifier, from models with lowest uncertainty and see how can it be used to improve the performance of the model.

A voting classifier is a type of machine learning estimator that develops a number of base models or estimators and makes predictions based on averaging their results. Voting for each estimator output can be integrated with the aggregating criterion [13].

There are two categories of voting criteria: Hard voting: Voting is based on the output class that is anticipated. Soft Voting: Voting is based on the output class's anticipated likelihood.

Flowchart of Ensemble With Voting Classifier Using Uncertainty as a Parameter



From this table 5.1, we choose three best estimators in respect to uncertainty value - Logistic Regression (1.19), Decision Tree(1.19),Support Vector Machine(1.20) and K Nearest Neighbor(1.19) with our classification dataset. Voting=‘soft’, and we taking parameter weight as [1.19,1.19,1.20,1.19],i.e uncertainty value of the respective models. The sequence of weights are given as the models are ensemble in this Voting Classifier. Now lets, examine the result of each of the base estimators of the voting classifier.

Table 5.2 : Uncertainty estimation of Voting Classifier

Results	Voting Classifier
Uncertainty	1.00
Certainty	0.99
Accuracy	0.78
Precision	0.74
Recall	0.74
f1-score	0.74

From this table 5.2, we can see that using Voting Classifier with best esimators (Logistic Regression, Decision Tree, Support Vector Classifier, and K Nearest Neighbor) or ensembled models with low uncertainty estimator, we can decrease uncertainty (from 1.99 to 1.00) value with good accuracy. Also we can trust our model with 99 percent certainty value.

5.1.2 Case 2: Ensemble With Voting Classifier Using certainty as a Parameter

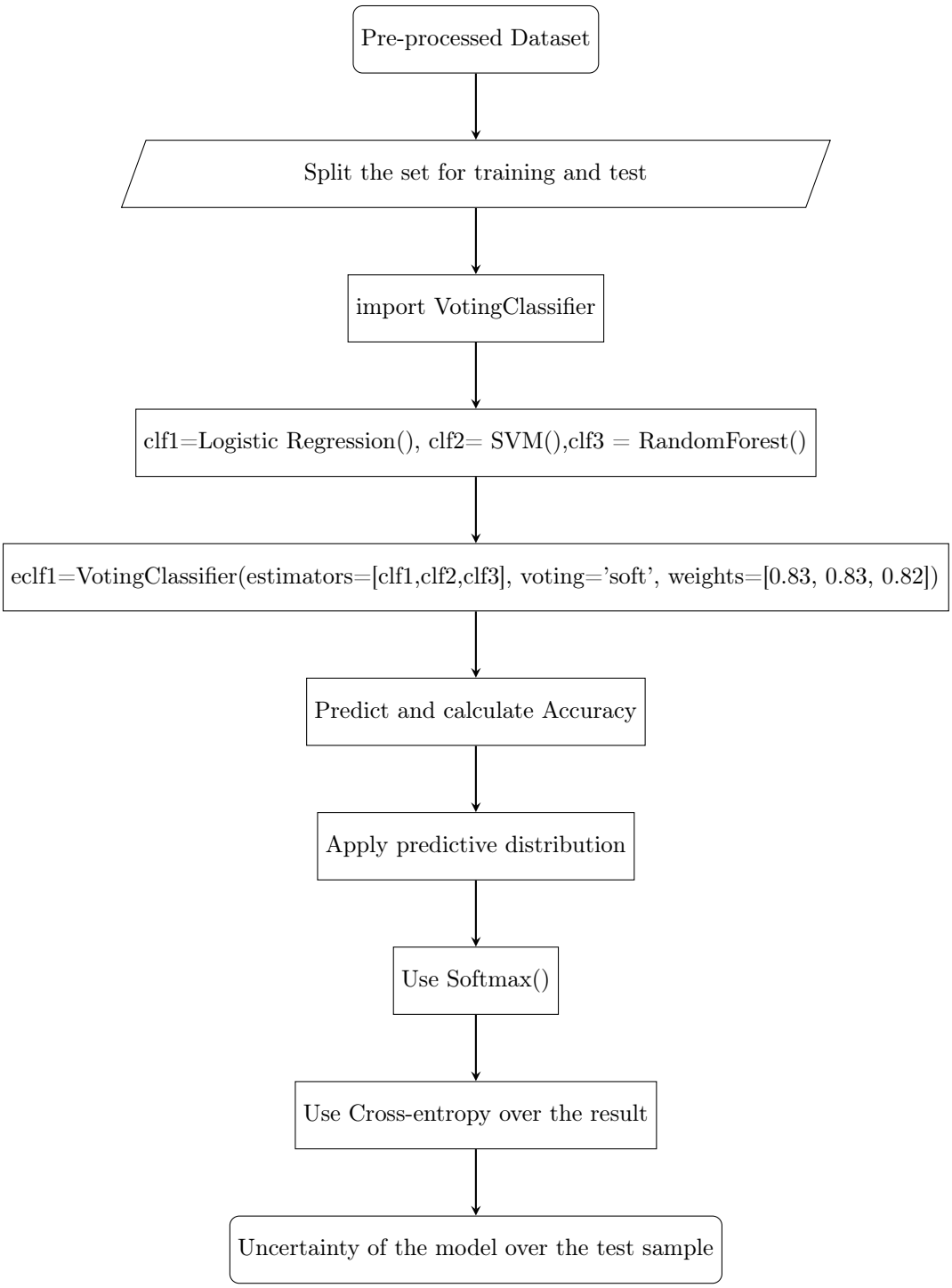
From Table 5.1, we get these Accuracy/Uncertainty values of different learning models.

Table 5.3 : Accuracy/Uncertainty results of different classifier

Results	Logistic Regression	Gaussian NB	Decision Tree	SVM	KNN	Random Forest
Accuracy/Uncertainty	65.54	34.71	56.30	65	57.98	62.80

Now, we are implementing voting classifier, from models with highest results and see how can it be used to improve the performance of the model. Here we choose Logistic Regression (65.54),SVM(65) and Random Forest(62.80) as estimator.Voting=‘soft’, and we taking parameter weight as [0.83,0,83,0.82],i.e certainty value of the respective models. The sequence of weights are given as the models are ensemble in this Voting Classifier.

Flowchart of Ensemble With Voting Classifier Using certainty as a Parameter



Now lets, examine the result of each of the base estimators of the voting classifier.

Table 5.4 : Uncertainty estimation of Voting Classifier Using certainty as a Parameter

Results	Voting Classifier
Uncertainty	1.00
Certainty	0.99
Accuracy	0.78
Precision	0.74
Recall	0.79
f1-score	0.78

From this table 5.4, we can see that using Voting Classifier with best estimators (Logistic Regression, Support Vector Classifier, and Random Forest), we can decrease uncertainty (from 1.99 to 1.00) value with good accuracy. Also we can trust our model with 99 percent certainty value.

Chapter 6

Conclusion and Future Scope

A measure of model believability is provided by an uncertainty analysis, which takes understanding one step further.A model with high uncertainty suggests that there are more legitimate data interpretations available, but that the model is unable to discriminate between them. Uncertainty analysis must be performed by everybody who is going to make a judgement. In many different disciplines, it may be used to compute, recognise, and express emotion. Despite the fact that the effort has produced intriguing findings, we wish to make specific improvements in future work to improve performance and acquire better results. But it's also crucial to make precise decisions and have a model with high dependability.

In first test case, from table 5.1, 5.2 , we can say that we choose the best models(lowest uncertainty) and ensemble them to get reliable model accuracy with only 1 percent pf uncertainty value or(99 percent certainty) of prediction using uncertainty as a parameter.

And in second test case, from table 5.3, 5.4, we can say that we choose the best models(highest result from accuracy/uncertainty) and ensemble them to get reliable model accuracy with only 1.0005 percent pf uncertainty value or(99 percent certainty) of prediction using certainty as a parameter.

This chapter demonstrates how uncertainty analysis enables us to make judgments on the accuracy of classifier predictions on a given dataset. The applications of this knowledge are numerous. When a classification system is utilised in the real world, for instance, when it is expensive to identify the actual label, the uncertainty of a forecast might be used as a gauge of trust.

In future we are using these for regression to check whether uncertainty estimation will effect on their prediction of accuracy or not and we will research uncertainty analysis for issues with various, difficult features.

Bibliography

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

[2] Munir Ahmad, Shabib Aftab, Muhammad Salman Bashir, and Noureen Hameed. Sentiment analysis using svm: A systematic literature review. *International Journal of Advanced Computer Science and Applications*, 9(2), 2018.

[3] Yassine Al Amrani, Mohamed Lazaar, and Kamal Eddine El Kadiri. Random forest and support vector machine based hybrid approach to sentiment analysis. *Procedia Computer Science*, 127:511–520, 2018. PROCEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING IN DATA SCIENCES, ICDS2017.

[4] Charles Blundell. Balaji Lakshminarayanan, and Alexander Pritzel. Simple and scalable predictive uncertainty estimation using deep ensembles. page 6403–6414, 2017.

[5] Deb Dutta Das, Sharan Sharma, Shubham Natani, Neelu Khare, and Brijendra Singh. Sentimental analysis for airline twitter data. *IOP Conference Series: Materials Science and Engineering*, 263:042067, nov 2017.

[6] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

[7] Sonoo Jaiswal. *Javatpoint, Naïve Bayes Classifier Algorithm.* Sonoo Jaiswal, 2011.

[8] Sonoo Jaiswal. *Javatpoint,K-Nearest Neighbor(KNN) Algorithm for Machine Learning.* Sonoo Jaiswal, 2011.

[9] Dr.A.Nisha Jebaseeli and S.Kasthuri. An efficient decision tree algorithm for analyzing the twitter sentiment analysis. *ISSN- 2394-5125*, 7(10):1010–1018, 04,2020.

[10] Iswariya Manivannan. A comparative study of uncertainty estimation methods in deep learning based classification models. Technical report, Fachbereich Informatik, 2020.

[11] Lotta Meijerink, Giovanni Cinà, and Michele Tonutti. *Uncertainty estimation for classification and risk prediction in medical settings.* 04 2020.

[12] Lotta Meijerink, Giovanni Cinà, and Michele Tonutti. Uncertainty estimation for classification and risk prediction on medical tabular data, 2020.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] Ankita Rane and Anand Kumar. Sentiment classification system of twitter data for us airline service analysis. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 01, pages 769–773, 2018.

[15] Nand Kishore Sharma, Surendra Rahamatkar, and Sachin Sharma. Classification of airline tweet using naïve-bayes classifier for sentiment analysis. In *2019 International Conference on Information Technology (ICIT)*, pages 70–75, 2019.

[16] Nurulhuda Zainuddin and Ali Selamat. Sentiment analysis using support vector machine. In *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, pages 333–337, 2014.