Enhancing Customer feedback processing with Machine Learning in Microsoft Azure

Hanna-Maria Sormunen

Master's Thesis

Big Data Analytics

2022

Hanna-Maria Sormunen

MASTER'S THESIS	
Arcada	
Degree Programme:	Big Data Analytics
Identification number:	
Author:	Hanna-Maria Sormunen
Title:	Enhancing Customer feedback processing with Machine Learning in Microsoft Azure
Supervisor (Arcada):	Amin Majd, D.Sc. (Tech.) and PhD
Commissioned by:	Finnish Tax Administration

Abstract:

Text Classification and Natural Language Processing (NLP) is developing fast, and all the applications are rapidly evolving, GPT-3 emerged in the field just last year and there are some open-source options in the field, like GPT-NEO or GPT-J. It is well known that data is the new oil. Especially the analytics side of NLP is yet to unravel.

This thesis introduces a process and Microsoft Azure cloud architecture to preprocess and classify incoming text data, in this specific case, Finnish Tax Administration customer feedback. The thesis will go through all the steps needed to preprocess the text to discover named entities, use Named entity recognition (NER) for pseudonymization, use machine learning (ML) for classifying purposes, and determine the optimal ML model for organizing the customer feedback.

The thesis divides into three main parts; the first part goes through the theoretical concepts of machine learning and state-of-the-art Natural Language Processing.

The second part covers the essential components of Microsoft Azure cloud and the options to use ML models and customer feedback processing. Also, the second section will cover the basic architecture for the application pipeline in Azure.

The third part focuses on developing the model itself and testing different models to find the best performing classification model for the task at hand, the classification of the customer feedback data.

We will be using SVM, Neural Networks, such as TensorFlow, and the coding is in Python.

Keywords:	Machine Learning, ML, Architecture, Microsoft Azure, NLP, Natural Language Processing, Finnish BERT, Finnish NER, Named Entity Recognition, Support Vector Machine, SVM, Neural Networks, TensorFlow, Generative Pre-trained Transformer, GPT, GPT-3, GPT-Neo, GPT-J
Number of pages:	51
Language:	English
Date of acceptance;	

FOREWORD

The field of Artificial Intelligence has fascinated humankind for centuries. In the 1500 century, the inventors of their time designed mechanical animals. In 1600 Blaise Pascal developed the first digital calculation machine, and Sir Samuel Morland designed an arithmetical machine for calculating. In the 1700 century, scientists developed a lot of mechanical devices, along with "Digesting duck" created by Jacques de Vaucanson and Wolfgang von Kempelen's invention, phony automatic chess player, "The Turk" (1769).

The Turk was revealed to be a hoax later, but the idea of creating human or even God-like intelligence has been the hidden goal of scientists and researchers all over the world for centuries.

As Frank Martela, a Finnish philosopher, said in a Customer Relation Unit personnel conference arranged by FTA in October 2021: "Change in our working life is driven by *GAD* (*globalization*, *automation*, *digitalization*)", which is very much true; we perform our everyday chores with different kind of Gadgets, they are here to help us.

These days, we already have cloud services, and we depend heavily on them. Artificial Intelligence is one form of trying to mimic natural creations and create the ultimate intelligence, a mastermind with all the knowledge and ability to utilize it to benefit humankind. One attempt in this task is GPT-3. GPT-3 was developed by OpenAI, a company that specializes in artificial intelligence.

In Taoism, Wu Wei is a concept of effortless action, the art of non-doing. That has been the real goal for human beings for centuries. Wu Wei.

> *When you enlarge your mind and let go of it,*
> *When you relax your [qi 氣] vital breath and expand it,*
> *When your body is calm and unmoving:*
> *And you can maintain the One and discard the myriad disturbances.*
> *You will see profit and not be enticed by it,*
> *You will see harm and not be frightened by it.*
> *Relaxed and unwound, yet acutely sensitive,*
> *In solitude you delight in your own person.*
> *This is called "revolving the vital breath":*
> *Your thoughts and deeds seem heavenly.*
> *(Original Tao: inward training (*nei-yeh*) and the foundations of Taoist mysticism, Harold D. Roth, 1999)*

CONTENTS

Figures

Tables

1 INTRODUCTION

1.1 Motivation

This thesis is commissioned by the Finnish Tax Administration, FTA, by the Customer Relations Unit. The problem to be solved with this thesis is how the customer feedback process could perform more efficiently. Approximately 1500 spontaneous customer feedback comes in from Vero.fi website each month. The input is gathered using the Webropol tool, and the feedback is fetched from Webropol's site manually, preprocessed manually, and classified manually. The customer feedback is essential information for FTA's processes to understand how they have been doing and how they could do better.

Feedback is a gift; someone takes time to give you feedback to get better.

Manual feedback processing is labor-intensive and inhibits the use of all the feedback channels. The processing must be done automatically instead of manual handling to include the other feedback channels in customer feedback handling. Several other channels provide textual material gained from the customer, but we will not be diving into these channels in this thesis. The channels produce from 30 000 up to 2 million rows of feedback a year per channel, so it is evident that the processing needs to be aided with automation.
Regarding analytics infrastructure, FTA has recently moved from on-premises services to cloud services, Microsoft Azure. This thesis covers the basic architecture design for the customer feedback processing in Azure and the machine learning model development using natural language processing.

1.2 Research Aim and Objectives

This thesis's research objectives and the aim is to automate the customer feedback processing in Microsoft Azure. Define the suitable architecture to handle the data pipeline in Microsoft Azure, design the necessary steps to preprocess the feedback, and define a machine learning model for classifying the feedback to given categories. There's quite a handful of research to be done in this area, so we reduced the research a bit. Let's try to answer these three research questions in this thesis:

1. What could be an appropriate method for classifying customer feedback?
2. What would be a proper architecture for customer feedback processing in Microsoft Azure?
3. What could be a practical method or tool for anonymization/pseudonymization of the customer feedback?

2 RESEARCH APPROACH

The research approach is hands-on analysis. Feasibility study, or proof-of-concept more specific, and it will cover the state-of-the-art review in what it comes to machine learning in the field of Natural Language Processing, model selection, and development

2.1 Data Source and Data

There was an organizational change in Finnish Tax Administration, and the organization

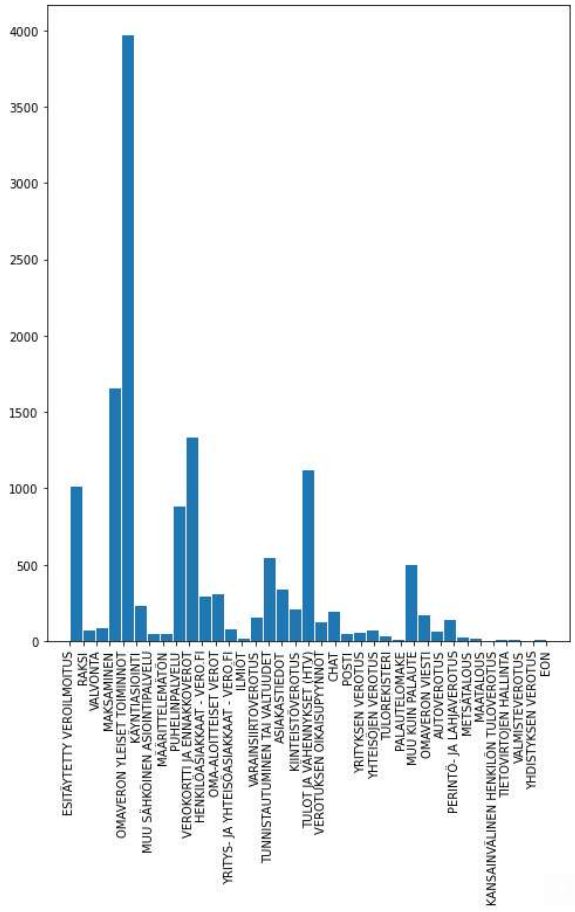
shifted to a product, or service, based organizational structure. The organizational change also led to customer feedback needing to be structured to serve the new product-based organizational structure optimally. The categories for feedback gathering were changed in May 2021.

In December 2021, around 13801 rows of feedback had been classified manually since May 2021.

The dataset contains 37 different categories. The pace at which the customer feedback is coming in is at proximately 1500-3000 feedbacks per month. Some of the categories have just a few feedbacks, and other categories are very popular, making the dataset very unbalanced. (Figure 1)

The feedback texts also vary in length. The median word count is 26 words, and the mean word count is 34.7 words per feedback. The maximum word count of feedback in the dataset is 1213 words.

The feedback texts are raw data; the data has not been processed except for the labeling and manual pseudonymization. There is some metadata available that we can use to get a better performance out of the classification. The metadata contains information that the customer has decided to give in the feedback form. Most of the questions are mandatory on the form.



ASIAKASTIEDOT	340
AUTOVEROTUS	64
CHAT	194
EON	6
ESITÄYTETTY VEROILMOITUS	1010
HENKILÖASIAKKAAT - VERO.FI	291
ILMIÖT	17
KANSAINVÄLINEN HENKILÖN TULOVEROTUS	3
KIINTEISTÖVEROTUS	206
KÄYNTIASIOINTI	228
MAATALOUS	14
MAKSAMINEN	1654
METSÄTALOUS	19
MUU KUIN PALAUTE	500
MUU SÄHKÖINEN ASIOINTIPALVELU	46
MÄÄRITTELEMÄTÖN	42
OMA-ALOITTEISET VEROT	306
OMAVERON VIESTI	170
OMAVERON YLEISET TOIMINNOT	3966
PALAUTELOMAKE	9
PERINTÖ- JA LAHJAVEROTUS	134
POSTI	42
PUHELINPALVELU	880
RAKSI	67
TIETOVIRTOJEN HALLINTA	9
TULOREKISTERI	29
TULOT JA VÄHENNYKSET (HTV)	1116
TUNNISTAUTUMINEN TAI VALTUUDET	541
VALMISTEVEROTUS	10
VALVONTA	80
VARAINSIIRTOVEROTUS	156
VEROKORTTI JA ENNAKKOVEROT	1333
VEROTUKSEN OIKAISUPYYNNÖT	120
YHDISTYKSEN VEROTUS	1
YHTEISÖJEN VEROTUS	68
YRITYKSEN VEROTUS	53
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	77

Figure 1 Customer Feedback distribution in different categories

This data will be used as training and testing data for the machine learning model.

3 RESEARCH BACKGROUND

3.1 Introduction

3.1.1 Architecture

The current situation of the Finnish Tax Administration's Cloud architecture is that there are several subscriptions in Microsoft Azure for different kinds of purposes or functions. There are multiple environments for development, testing, data-development and data-testing, and the production environment. The structure in each sub-environment is the same for any application, but the data content, the access policy, and application architectures differ. The architectures are based on various Microsoft reference architecture models.

3.1.2 Technology stack

A Microsoft technology stack is available; projects can use a selection of Microsoft Azures components. The usage of these components, technologies, and data is regulated heavily; everything goes through tight inspections considering information security and data protection.

3.1.3 Dataflow

The dataflow design is that the data is fetched through Webpropol's API in comma-separated value format. (.csv). Currently, the dataset is fetched manually from the Webropol tool in excel format. The data is then preprocessed to remove any sensitive data the customers might have written in their feedback. After the pseudonymization, the classification is performed with a machine learning model that has been trained with our datasets. This process and the technical details are covered in the coming chapters.

4 THEORETICAL BACKGROUND

4.1 Machine Learning, Artificial Intelligence and Deep Learning

Artificial Intelligence (AI) is referred to as the simulation of human intelligence by machines, especially computer systems. **Machine Learning (ML)** in computer science uses algorithms that can improve automatically through experience and by the use of data. Machine Learning is seen as a part of Artificial Intelligence**.** (Figure 2) Machine Learning models are built with sample data, so-called training data, to make predictions or decisions instead of programming the rules explicitly in code. Machine Learning models are used in various applications today, such as computer vision and data mining. (Haenlein and Kaplan, 2019) (Alzubi et al., 2018)

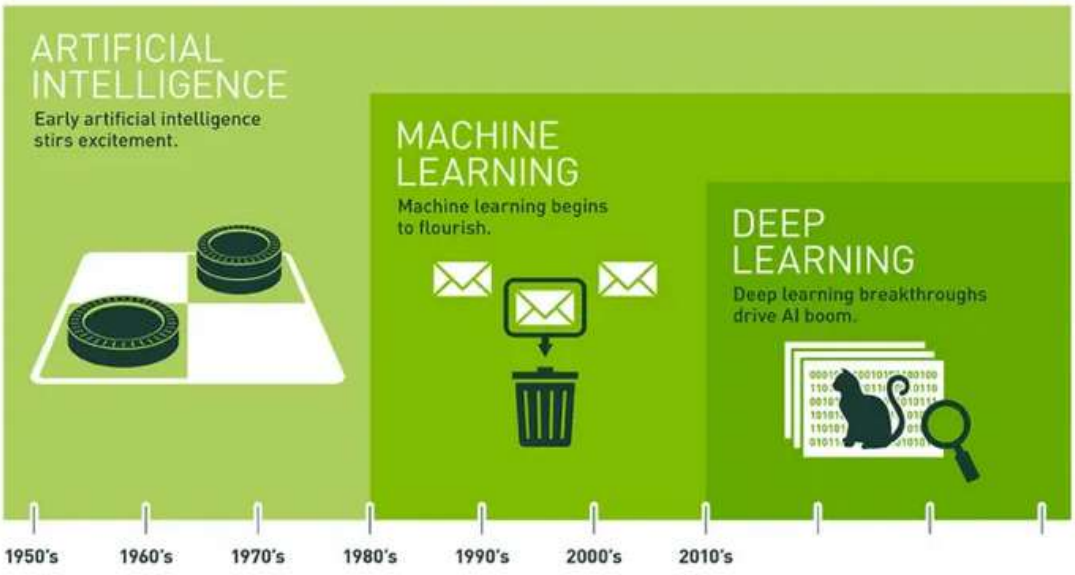


Figure 2 Artificial Intelligence has intrigued mankind for decades. (Copeland, 2016)

Deep learning is a subfield of machine learning. It uses **neural networks** for solving problems. In deep learning, several hidden neural network layers of neural nodes learn weights in each layer to have the desired output. There is a specific step in developing a model called "*feature engineering"* for more traditional machine learning models. Only the key components, or principal components, for the model optimal performance are selected in the feature engineering step. With deep learning, the selection of the most

important information for the model is made by the neural network model. (Figure 3) It automatically sets the weight of the critical information higher from those not that important, essentially eliminating the need for feature engineering. (Oppermann, 2020)

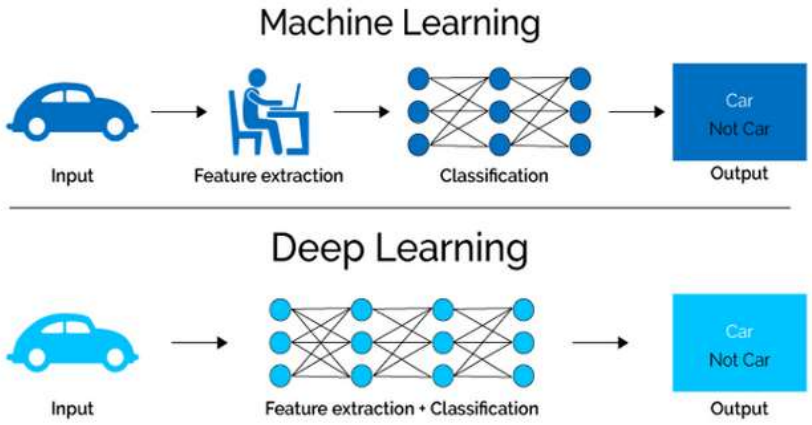


Figure 3 Machine Learning vs Deep Learning. Deep Learning does the feature extraction in the learning process.(Oppermann, 2020)

The machine learning field is divided into two main areas: **unsupervised and supervised learning**. As the name says, unsupervised learning models are not provided with pre-defined labels or targets that the model should be able to reach by learning the training data. **Unsupervised learning** algorithms learn the naturally occurring patterns in the training data. The features found at the learning stage are then used in the future to discover similar items that have similar features. A typical unsupervised learning example is clustering, where the model automatically groups given data to a given number of categories. (Liu, 2011) (Ghahramani, 2004)

Supervised learning, in contrast, has the category labels predefined before the model training begins. **Semi-supervised learning** instead combines both methods of learning. In semi-supervised learning, only a small number of data is pre-labeled, and the other part of the training data is unlabeled. (Kingma et al., 2014)

There are also other traits of training the machine learning models, like **reinforcement learning (RL)**. Reinforcement learning, also called approximate dynamic programming, or neuro-dynamic programming, does not need labeled input data; instead, the reward, the label, is given in after the model has acted. And the model is retraining itself along the process with given timesteps. The model will learn the optimal performance when the user provides reinforcing signals to the learning process. This is remarkably

similar to how the human brain, or animals, works when trying to learn new things. Reinforcement learning has been applied in problems like elevator scheduling and robot control. (Francois-Lavet et al., 2018)

You might also encounter the term **transfer learning (TL)**, which is not a model training or a learning method itself, but a way to get the gained knowledge from a pre-existing already trained model and reuse it in a new model in different related kinds of problem-solving application. Like when a model is trained to predict whether an image is a cat or a dog, the same model, with some modifications, can be used to predict cars from trucks. Transfer learning is commonly used with neural networks models as they need a lot of data to train. (Pilault et al., 2021)

4.2 Neural Networks

Neural networks are machine learning algorithms used to detect complicated patterns in data using multiple hidden layers and non-linear activation functions. A neural network takes an input, hands it through multiple layers of hidden neurons, and outputs a prediction representing the merged input of all the neurons.

Neural networks are trained *iteratively* using optimization techniques like gradient descent. Error metrics are calculated on the difference between prediction and target after each training cycle. The derivates of error metrics are calculated and propagated back through the network using backpropagation. Each neuron's coefficients (weights) are adjusted until network error drops below an acceptable threshold.

(“Concepts — ML Glossary documentation,” n.d.)

The accuracy of a neural network is heavily dependent on the amount of training data. (Figure 4)The traditional machine learning methods perform better than neural network models when the training data size is small. (Oppermann, 2020)

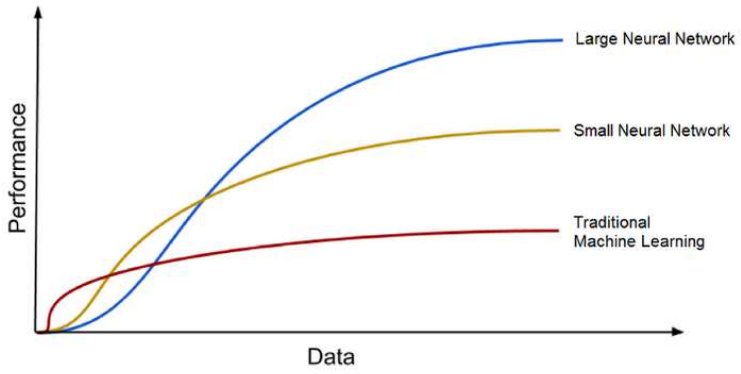


Figure 4 The performance of a Neural Network vs. traditional Machine Learning model depends heavily on the amount of data available for training. (Oppermann, 2020)

4.3 Support Vector Machine SVM

Support vector machines are supervised learning models from the more traditional machine learning spectrum. SVM algorithms analyze data for classification and regression analysis. They are very widely used machine learning models that use support vectors to distinguish between classes the user is trying to classify. (Figure 5) They perform well even with a small amount of data is available. SVM models are easy to use and consistently perform well on text classification tasks. (Joachims, 1998)

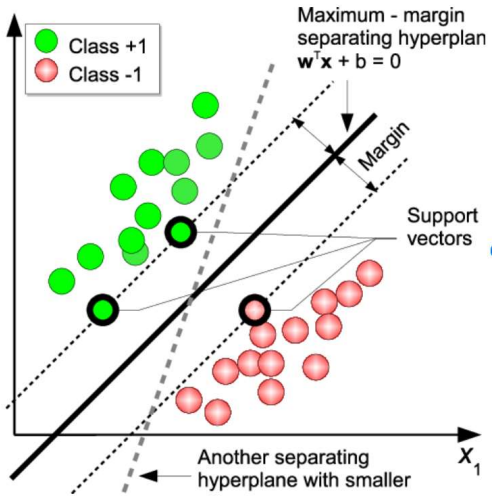


Figure 5 Principle of support vector machine (SVM). A maximum margin hyperplane that separates the positive (+1) and negative (-1) training examples is found by an optimization process. (Kinnunen, n.d.)

4.4 Hyperparameter tuning

Hyperparameters are parameters that control the learning process of the machine learning algorithm. They are not part of the resulted model, but they determine heavily how the model develops in its training phase. They are a set of parameters that are defined before the actual learning process takes place.

For example, the previous support vector machine has a hyperparameter of C which defines the width of the band that divides the two groups with a vector, and kernel and gamma, which determine the kernel type to be used in the algorithm and its coefficient.

Hyperparameters are also used in preprocessing, like in this customer feedback, in case the machine learning pipeline uses Tf-Idf conversion of the texts. The Tf-Idf conversion will also have its hyperparameters that will affect the machine learning model's performance. For example, a ngram hyperparameter can be given for the Tf-Idf converter. The ngram and Tf-Idf conversions are covered in the incoming paragraphs.

For Hyperparameter tuning, it is common to use a systematic approach to find the best-performing model and hyperparameter sets.

In the customer feedback processing case, the processing pipelines used Sklearn packages' ***GridSearchCV*** to find the best performing cross-validated hyperparameters.
(Claesen and De Moor, 2015)

4.5 Evaluating Machine Learning

In a supervised classifying problem, a machine learning model's evaluation is based on having data and labels to compare with the machine learning model's predictions. Commonly this is done by splitting the data into training, validation, and testing datasets. (Figure 6)

The model is trained with a training dataset, and test or/and validation datasets are used to evaluate the true performance of the trained model. The validation dataset is used for assessing the model when for example, hyperparameter tuning has been used in the testing dataset to find the perfect hyperparameters for the model's optimal performance.

The testing dataset is used to evaluate the results of the model's performance, and it must not have any part in the model training.

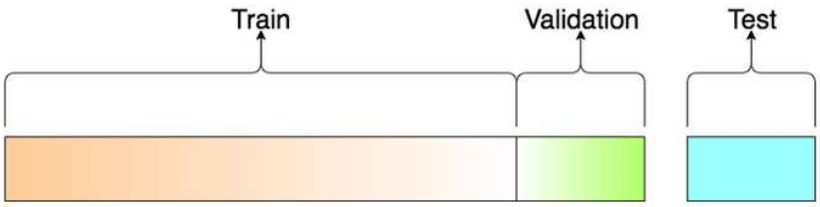


Figure 6 Illustration of training, validation and testing datasets. (Shah, 2020)

The general problem is that there is only limited training and testing data available since gathering the data might take time, and the labeling of the training data is time-consuming and, thus, expensive. To address this issue, model developers typically use ***K-fold Cross-Validation***. It divides the data into K number of splits and folds and uses them sequentially, dividing the splits into separate testing and training sets. In the Figure 7, the fold1, split1 is used as the validation data, and the rest is used as training data. (Kumar, 2021)

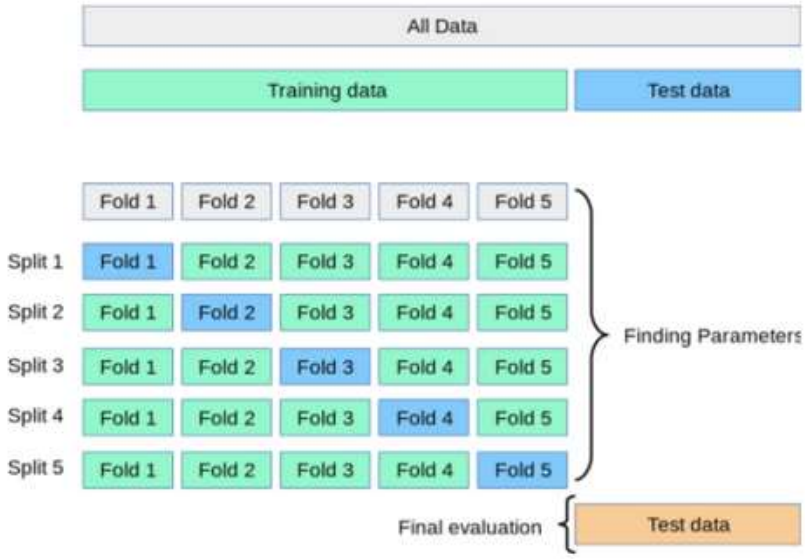


Figure 7 Training, validation and testing datasets with 5 folds with K-fold Cross Validation. (Kumar, 2021)

4.5.1 Confusion Matrix

Understanding the ***confusion matrix*** is first useful to understand the concepts of true positive and false positive, true negative and false positive.

A true positive is when the predicted and actual test values in the dataset are both the same. This is also the case with a true negative; when the predicted false and the actual false are true simultaneously, the prediction is a true negative.

If the predicted value fails to predict the actual value correctly, it is called false positive or false negative. When predicted and actual values are drawn in the same matrix, this is a confusion matrix. (Figure 8) When a model predicts all classes correctly, the values will be aligned to the diagonal of the matrix. (Jayaswal, 2020)

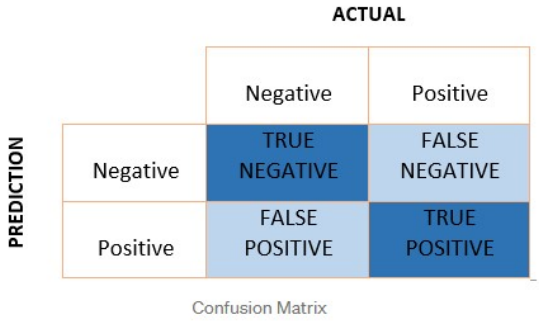


Figure 8 Basic confusion matrix (Jayaswal, 2020)

The confusion matrix for a multilabel classification contains multiple columns and rows instead of this straightforward two-row positive/negative, true/false matrix. (Figure 9)

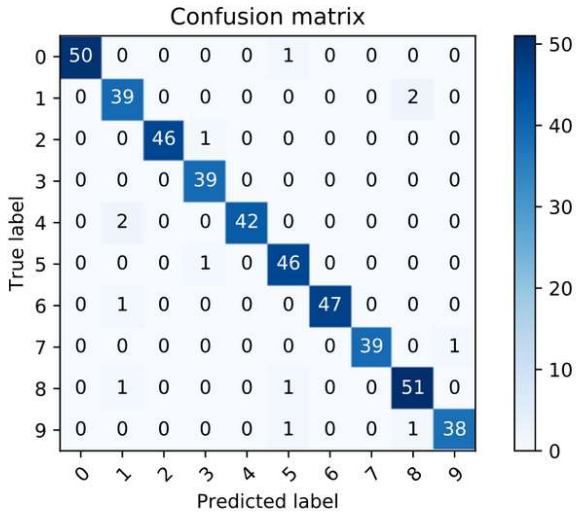


Figure 9 Example of a multilabel confusion matrix. (Khalusova, n.d.)

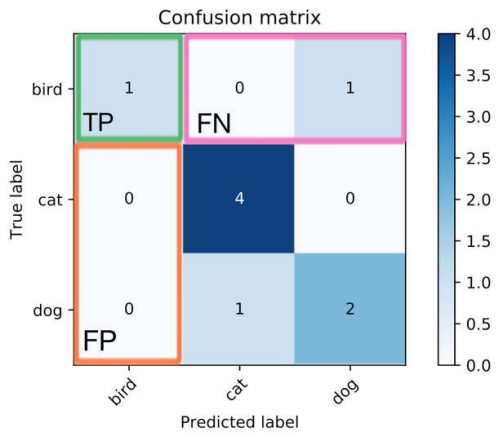


Figure 10 Example of true positive, false positives and false negatives in a confusion matrix. (Khalusova, n.d.)

In multiclass classification, true positives are the same as for binary classification, and also is the same for the true negatives, which are the other correctly classified predictions, but false; both false positives and false negatives; predictions are calculated together from all the values missed to predict the actual value. (Figure 10) (Khalusova, n.d.)

4.5.2 F1-Score, Precision and Recall

To evaluate the performance of a classifier, the model's ***precision, recall, and F1-score*** can be calculated. ***Precision*** calculates what number of predicted positives are truly positive. (Figure 11)

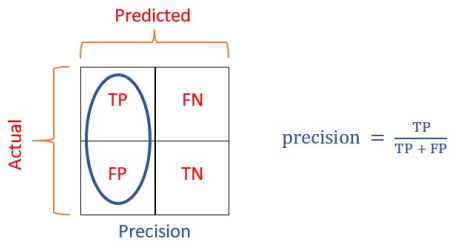


Figure 11 Precision calculates what number of predicted positives that are truly positive. (Bex, 2021)

Recall calculates what proportion of the actual positives are correctly classified. (Figure 12)

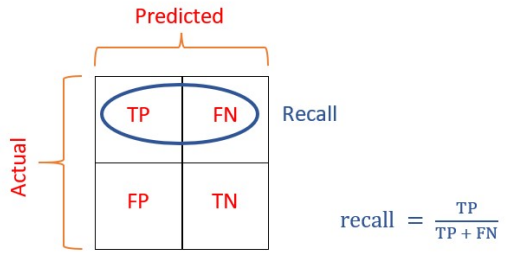


Figure 12 Recall calculates what proportion of the actual positives are correctly classified (Bex, 2021)

F1-score calculates both recall and precision; it tries to minimize both false negatives, and the recall, and false positives, and the precision. (Figure 13)

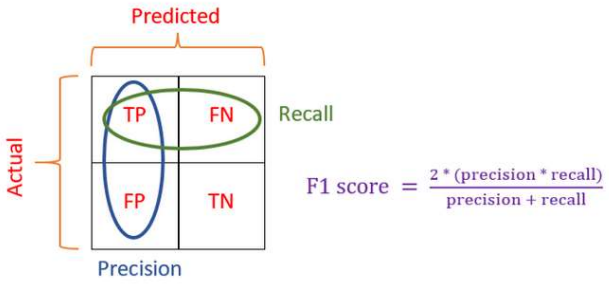


Figure 13 F1 score calculation (Bex, 2021)

4.6 Natural Language Processing (NLP)

4.6.1 Introduction to NLP

Natural language processing (NLP) is a special artificial intelligence (AI) field that makes human language understandable to machines. In short, a computer program can break down and interpret human language as it is spoken and written. NLP is used to comprehend the structure and meaning of human language by analyzing different factors like syntax, semantics, pragmatics, and morphology. Computer science transforms this linguistic knowledge into machine learning algorithms used to solve specific problems and tasks.

NLP is used in multiple applications, like autocorrect, autocomplete, language translators, chatbots and virtual assistants, social media monitoring, survey analysis, customer feedback processing, targeted advertising, recruitment, voice assistants like Siri, Alexa or Google Assistant, grammar checkers, and email filtering.

In May 2020, OpenAI introduced Generative Pre-trained Transformer 3 (GPT-3), an autoregressive language model that uses deep learning to produce human-like text. OpenAI has also presented in August of 2021 Codex, an artificial intelligence model that generates code in response to natural language input. Codex is based on GPT-3 and is fine-tuned with 54 million GitHub repositories to produce the code, and OpenAI claims Codex can function in over a dozen programming languages. (OpenAI, 2021)

4.6.2 Bag-of-Words (BoW)

Bag-of-words is a method of using word counts for classifying tasks. The words in the text are not in relation to each other. Bag-of-words only count the times the word appears in the text. With word2vec, the place where the word appears in the text has meaning. With BoW, the sentence meaning is discarded; only the word count is relevant in classifying tasks.

The features of the text are based on individual word or word pairs. This classic movie review analysis from Purva (2020) demonstrates the behavior of the bag-of-words with an example of movie reviews:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

Review 1 and 2 have contrasting meanings. Review 1 says the movie was very scary, and review 2 says it was not scary. Bag of words does not take this kind of contrast into account very effectively; it simply puts the words into dictionaries and says both review 1 and 2 has the word scary in them. The model using this kind of approach of handling data to perform, for example, classification, might have to rely on some other word or word pair to classify whether the movie was scary or not scary. Also, review 3 says the movie was scary, just with a different term. It depends heavily on the classifying task itself what approach or method should be used.

For the classifying task at hand, the meanings or intentions, or semantics don't matter; it is enough to classify the feedback to the correct area.
(Purva, 2020)

4.6.3 Term Frequency-Inverse Document Frequency (TF-IDF)

First, let's have a formal definition around Tf-Idf, Term Frequency - Inverse Document Frequency. Here's how Wikipedia puts it:

"Term frequency-inverse document frequency is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus." (Wikipedia contributors, 2022)

Let's calculate Tf-idf weights with a straightforward restaurant review example to understand the concept.

Review 1: "The food was terrible, and service was very slow"
Review 2: "The food was good, and location was amazing.
Review 3: "The location was nice, and the food was good"

The way to calculate the term frequency in the document:

$$tf_{t,d} = \frac{n_{t,d}}{Number\ of\ terms\ in\ the\ document}$$

Table 1 Term frequency calculation example

Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3
)The	1	1	1	1/9	1/8	1/9
food	1	1	1	1/9	1/8	1/9
was	2	2	2	2/9	2/8	2/9
terrible	1	0	0	1/9	0/8	0/9
and	1	1	1	1/9	1/8	1/9
service	1	0	0	1/9	0/8	0/9
very	1	0	0	1/9	0/8	0/9
slow	1	0	0	1/9	0/8	0/9
good	0	1	1	0/9	1/8	1/9

amazing	0	1	0	0/9	1/8	0/9
nice	0	0	1	0/9	0/8	1/9
location	0	1	1	0/9	1/8	1/9

To calculate the inverse document frequency:

$$idf_t = log \frac{number\ of\ documents}{number\ of\ document\ with\ term\ 't'}$$

Table 2 Inverse document frequency calculation example

Term	Review 1	Review 2	Review 3	IDF
The	1	1	1	0
food	1	1	1	0
was	2	2	2	0
terrible	1	0	0	0.48
and	1	1	1	0
service	1	0	0	0.48
very	1	0	0	0.48
slow	1	0	0	0.48
good	0	1	1	0.18
amazing	0	1	0	0.48
nice	0	0	1	0.48
location	0	1	1	0.18

And finally, how to calculate the Tf-Idf weight for a word in the corpus:

$$(tf\_idf)_{t,d} = tf_{t,d} * idf_t$$

Table 3 Tf-Idf calculation example

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
The	1	1	1	0	0.000	0.000	0.000
food	1	1	1	0	0.000	0.000	0.000
was	2	2	2	0	0.000	0.000	0.000
terrible	1	0	0	0.48	0.053	0.000	0.000
and	1	1	1	0	0.000	0.000	0.000
service	1	0	0	0.48	0.053	0.000	0.000
very	1	0	0	0.48	0.053	0.000	0.000
slow	1	0	0	0.48	0.053	0.000	0.000
good	0	1	1	0.18	0.000	0.023	0.020
amazing	0	1	0	0.48	0.000	0.060	0.000

nice	0	0	1	0.48	0.000	0.000	0.053
location	0	1	1	0.18	0.000	0.023	0.020

With Tf-Idf weights, weights are given for each word in the corpus and distinguish the essential words from ordinary words. Like in the customer feedback task, the term weight distinguishes the words that belong to different categories and separates feedback from another. These weights make it easier for the model to learn the basis of categorization.

4.6.4 Word2vec embeddings

Word2vec tries to find words with similar meanings by calculating a spatial location for a word in space. The difference is presented by adding embeddings or the location of the word in the vocabulary. The most common example of word2vec embeddings is perhaps the example: "Man is to king what woman is to queen". Words man' and 'woman' are placed near each other in the multidimensional space, and also the distance and direction from the words' man' and 'woman' are approximately the same as for words 'king' and 'queen'.

Also, we can have the countries and their capitals presented as vectors in this word space. In the figure, the capital cities are clustered together, and countries are clustered near each other. Also, the vector lengths and directions between the countries and the capitals are approximately the same. (Figure 14) The vectors can be calculated easily and find the relations of the words. This calculation is called vector arithmetic. For example, when vectors Beijing - China are calculated, we get the same result as Rome - Italy. This way, the result of Rome - Italy + China would be Beijing. (Mikolov et al., 2013)

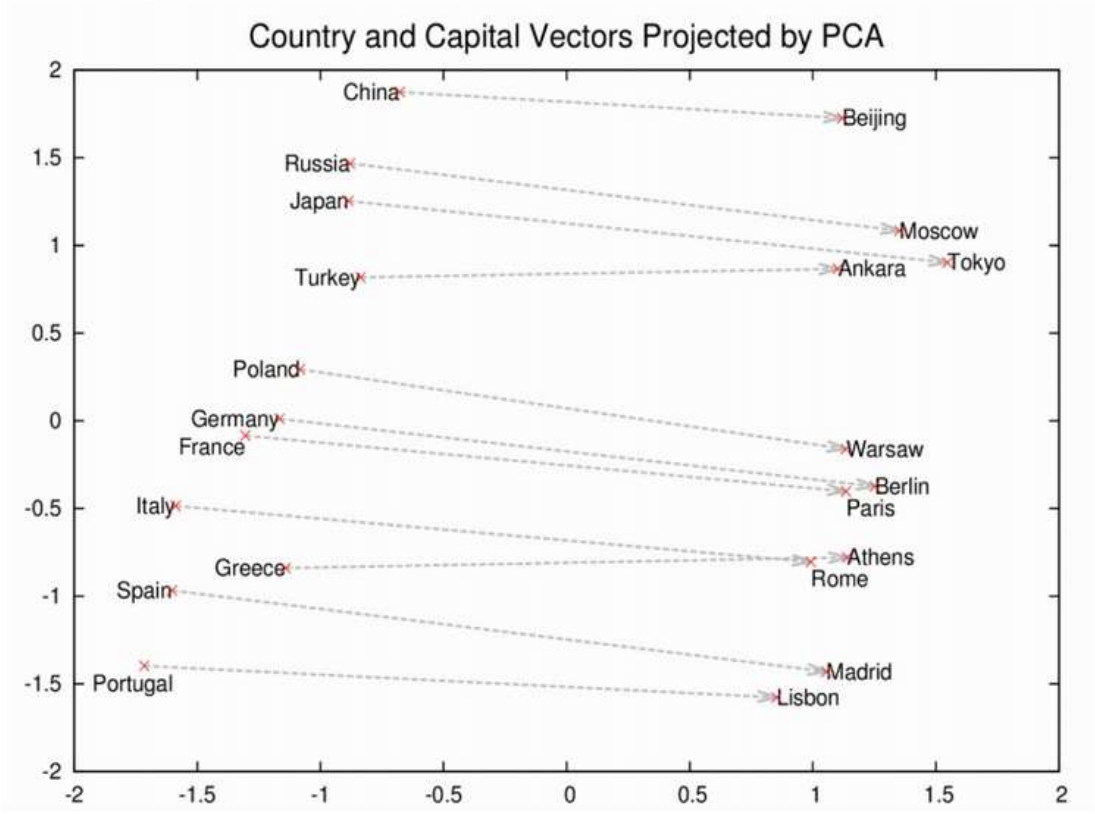


Figure 14 Word2Vec presentations of Country vs Capital vectors in embeddings. (Mikolov et al., 2013)

You can also query Word2vec models for other associations. The calculation also works with words that are not analogies that reflect each other. There can be interesting discoveries this way.

- *Usa - weapons = Canada*
- *Philosophy - politics = Organizational leadership*
- *Human + Philosophy = Psychology*
- *Russia - army = Ukraine*
- *Usa - dollar = Tao - Wu-Wei*
- *Environment - economics = Sustainable Development*

The above samples are calculated with pre-trained vectors trained on the Google News dataset 300. The number 300 represents the number of dimensions in the word space.

4.6.5 Lemmatization and stemming

In many languages, read and written, words appear in multiple *inflected* forms. Meaning that they are made up of words that seem to be derived from another word. For instance, the verb *'to walk'* may occur as *'walk', 'walked', 'walks' or 'walking'*. The non-inflected

form, *'walk'*, is called the *lemma* for the word. The connection of the base form with a part of speech is often called a *lexeme* of the word. (Wikipedia contributors, 2021b)

Lemmatization is closely related to stemming, and the difference is that a stemmer works on a single term *without* knowledge of the context and therefore cannot distinguish between words that have different connotations depending on part of speech.

Stemming is finding the root of the word, the root not having to be a word itself. An example of how the Porter algorithm reduces, *argue*, *argued*, *argues*, *arguing*, and *argus* to the stem *argu*. (Jabeen, 2018)

Juliet Beth Lovins created the first algorithm for word stemming already year 1968 (Lovins, 1968) and Martin Porter continued the work further. As Martin Porter says:

"*One can make some distinction between root and stem. Lovins (1968) sees the root as the stem minus any prefixes. But here we will think of the stem as the residue of the stemming process, and the root as the inner word from which the stemmed word derives, so we think of root to some extent in an etymological way. It must be admitted that when you start thinking hard about these concepts root, stem, suffix, prefix ... they turn out to be very difficult indeed to define." (Porter, n.d.)*

Customer feedback classification will use stemming since it gives better results than lemmatization. Natural Language Toolkit (NLTK) package has a stemmer, Snowball Stemmer, that supports all the three main languages needed, Finnish, Swedish and English. Snowball Stemmer supports the following languages: Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Portuguese, Romanian, Russian, Spanish and Swedish. ("NLTK :: nltk.stem.snowball," n.d.)

4.6.6 N-gram

N-gram in NLP is typically the number of words appearing next to each other. There can be word pairs that occur constantly together, and as a pair, a classification performed against such word pair gives better results.

From the example sentence *"The cow jumps over the moon"* from (Ganesan, 2018)

If N=2 (or bigrams), then the ngrams would be:

the cow
cow jumps
jumps over
over the
the moon

So there is 5 bigrams in this case. Notice that moving one word forward from *the->cow to cow->jumps to jumps->over*, etc., next bigram can be generated.
If N=3, the n-grams would be:

the cow jumps
cow jumps over
jumps over the
over the moon

When **N is 1**, they are called **unigrams**, and they are basically single words in a sentence. When **N is 2** , this is called **bigrams**, and when **N is 3**, they are **referred as trigrams**. (Ganesan, 2018)

4.6.7 BERT and Finnish BERT (FinBERT)

Bidirectional Encoder Representations from Transformers (**BERT**) is a pre-trained language model developed by Jacob Devlin and his colleagues from Google in 2018. BERT is trained on BooksCorpus (800M words) and Wikipedia (2,500M words) and

It is a machine learning technique for natural language processing (NLP), and it can be retrained with one additional output layer to fine-tune the model for a wide variety of applications. (Devlin et al., 2018)

Finnish BERT was developed by Turku NLP group, and it is a BERT deep transfer learning model for Finnish. The model can be further fine-tuned to accomplish cutting-edge results for Finnish natural language processing assignments.

Turku NLP project says Finnish BERT has been pre-trained for 1 million steps on over 3 billion tokens (24B characters) of Finnish text pulled from news, online discussion,

and the internet. By contrast, Multilingual BERT was trained on Wikipedia texts, where the Finnish Wikipedia text is roughly 3% of the amount used to train Finnish BERT.

These features allow Finnish BERT to outperform Multilingual BERT and all previously proposed models when fine-tuned for Finnish natural language processing tasks. From Virtanen et al (2019), we can see that Finnish BERT performs the best for Part-of-speech tagging with different datasets compared to the multilingual BERT and other two models. Cased Finnish BERT Part of speech tagging results was 98.23% compared when Cased Multilingual BERT reached 96.97%. (Virtanen et al., 2019)

4.6.8 Named Entity Recognition NER and Finnish NER

Named entity recognition, NER is essentially word classification; the goal of NER is to label names of people, addresses, organizations, and other entities of interest in text documents. There are three main approaches to NER: lexicon-based, rule-based, and machine learning-based.

- The lexicon-based method uses a list of words for identifying entities.
- Rule-based systems use manually or automatically construct rules for entity detection.
- Machine learning-based approaches use probability models and features derived from the input text.

Finnish NER was created by Luoma et al. (2020) and it is founded on the Finnish BERT and it outperforms BERT NER in English, Finnish and Dutch. F1-scores for Finnish NER are 93.78% in compared to BERT NER tagging F1-score 93.11%. (Luoma et al., 2020)

The Finnish NER tagger can be used with online demo from Finnish NER site. (Figure 15) https://turkunlp.org/fin-ner.html.

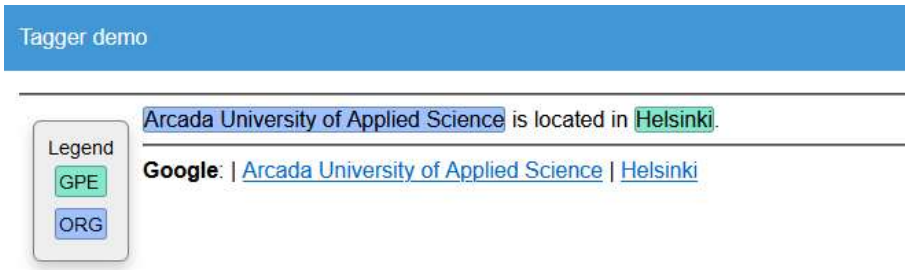


Figure 15 An example of Finnish NER tagged sentence using Finnish NER tagger demo. https://turkunlp.org/fin-ner.html

The are other tools for Finnish NER also available. For example, FiNER, part of the Finnish Tagtools application collection. FiNER is a rule-based named-entity recognition tool for Finnish, developed at the University of Helsinki for the FIN-CLARIN consortium. Microsoft Azure has Text Analytics cloud service that supports named entity recognition in Finnish.

In Ajanki (2021) evaluation, Turku NER gained the highest F1 scores with values above 90% on all named entity types when comparing these three NER taggers. (Figure 16)

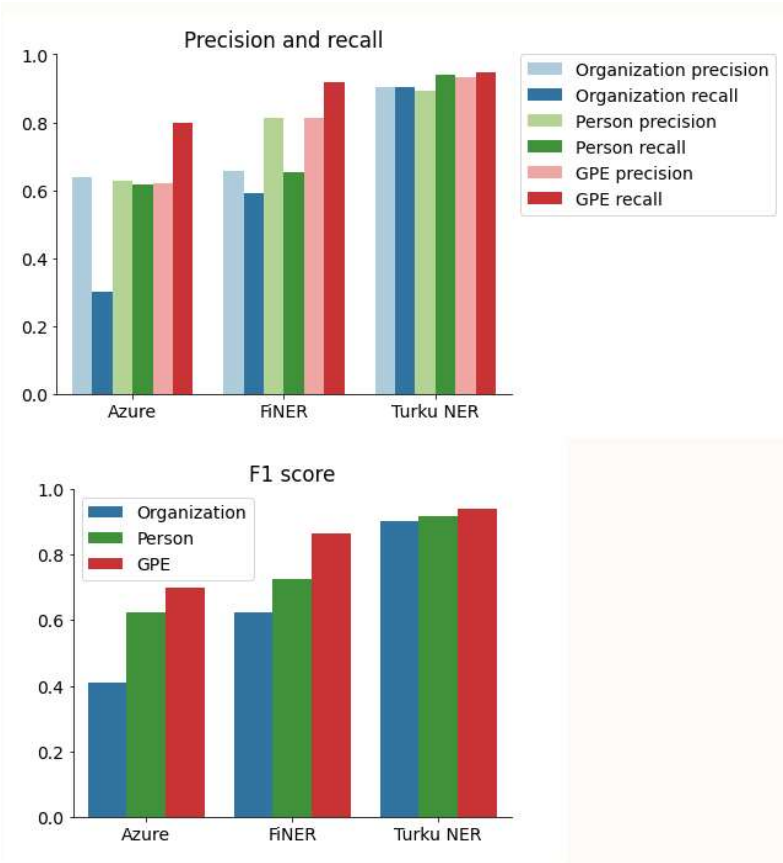


Figure 16 Comparison between Turku NER, FiNER and Azure (Ajanki, 2021)

Ajanki (2021) states that the Turku NER is the most accurate of the tested services and the Turku NER should be the first choice for named entity recognition in most cases.

4.6.9 Generative Pre-trained Transformer, GPT-NEO, GPT-J, GPT-3

Generative Pre-trained Transformer (GPT) models are autoregressive language models that use deep learning to produce human-like text.

GPT models can be used widely in the area of human interaction due to their ability to adapt. You can generate code with these pre-trained models, do chatbots with question answering, sentiment analysis, intent classification, tutorial generation, paraphrasing and summarization, text generation from a subject, keyword or key phrase extraction, machine translation.

The idea and power of these GPT models are in the pre-training. They are trained to learn by using thousands or millions of examples and thus have developed a human brain kind of structure in the neural network model.

GPT-3, introduced in May 2020, was developed by OpenAI researchers, and it has GPT-3's full version has a capacity of 175 billion parameters.

GPT-J (released in June 2021) and GPT-Neo (released March 2021) are the open-source alternatives to GPT-3, created by EleutherAI collective, a group of researchers working on open-source AI. GPT-J has six billion parameters, making it the most advanced open-source NLP model. GPT-Neo has 1.3 billion and 2.7 billion parameter versions available. The models can also be fine-tuned to the tasks in hand, and both GPT-J and GPT-Neo are available readily available using the NLP Cloud API or as open-source code in Github.

Also, there is a possibility to use generative models in other functions as well. For example, synthetic data can be generated using GPT models in case there is not much data available for NLP model development. (Whitfield, 2021)

5 CUSTOMER FEEDBACK PROCESSING IN MICROSOFT AZURE CLOUD

5.1 Introduction

The architecture needs to be defined for the Microsoft Azure cloud to get the classification application in use. Both the application pipeline and the classification model pipeline. Microsoft Azure cloud provides multiple options, we can develop Infrastructure as a Service (IaaS), we can have Platform as a Service (PaaS), and we can have Software as a Service (SaaS). (Figure 17) In these services, we have different responsibilities for us to manage. In our use case, we will be using Azure Cloud Platform as a Service; we will be responsible for the Applications and Data, the vendor, Microsoft, manages other areas.

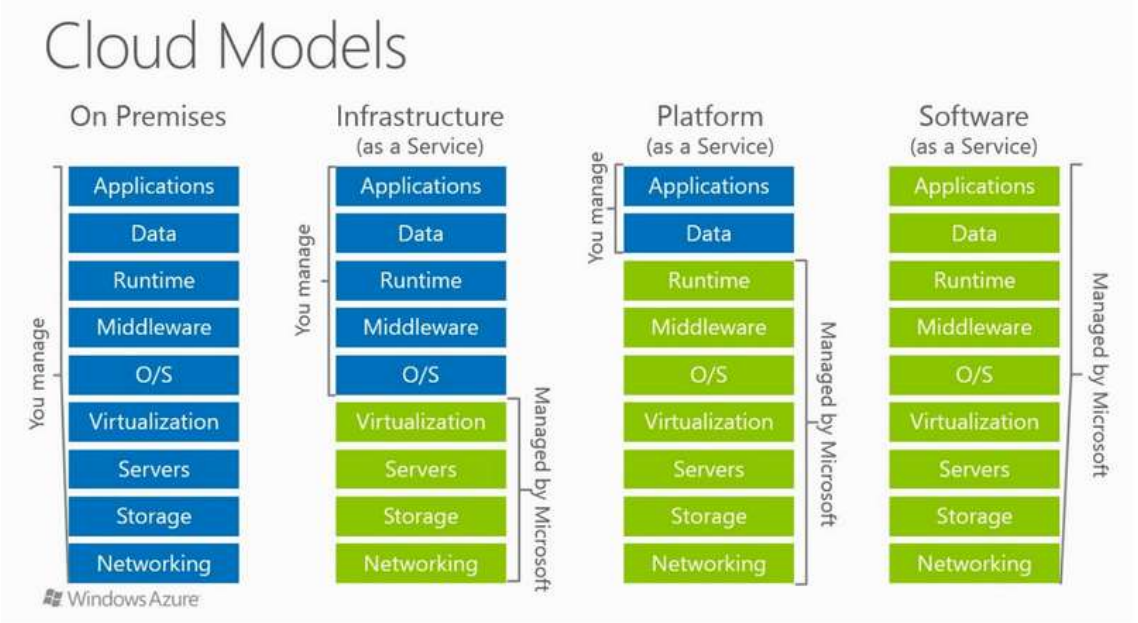


Figure 17 Microsoft Azure Cloud models, IaaS, PaaS and SaaS. (Smith, 2019)

5.2 Microsoft Azure Services

This section introduces Microsoft Azure services utilized in the development of Customer feedback processing. Let's cover them briefly.

Microsoft **Azure Machine Learning Studio** is a service used to train and retrain machine learning models. The ML studio services include the model registry for keeping track of the model versions.

Azure Blob Storage is Microsoft's object storage solution for the cloud. Blob storage is the best option for holding enormous quantities of unstructured data.

Blob storage is optimal choice when providing images or documents straight to a browser or storing files for distributed access, or for streaming video and audio, even for writing to log files, storing data for backup and restore, creating disaster recovery, and archiving and storing data for analysis by an on-premises or Azure-hosted service. (Microsoft, n.d.)

Azure Data Lake Storage cans store data in Microsoft Azure cloud in a more structured file format. Data Lake Storage Gen2 is very cost-effective because it is built on top of low-cost Azure Blob Storage.

Azure Data Factory is a service used for schedule file processing of all kinds. The fetching of the file through APIs and processing the files further is performed with the ADF. It orchestrates the sequence of how the data is handled and stored.

Azure Functions is a serverless microservice that can be used to call to perform specific tasks. For example, one Azure function can give a classification to a particular data, or one can perform named entities removal.

The Azure SQL database is a database that can store data in the Microsoft Azure cloud. Azure SQL Database is managed as a service (PaaS) database engine that handles most database management functions like patching, upgrading and creating backups, and monitoring without user involvement. (Microsoft, n.d.)

5.3 Customer Feedback Processing Application Pipeline

The customer feedback proof-of-concept application architecture in Microsoft Azure cloud is designed with the Webropol survey user in mind. (Figure 18) The Azure Data Factory (ADF) fetches the Webropol survey answers in a file format from the Webro-

pol, using Webropol API. Both REST API and SOAP API are available; REST uses JSON format, and SOAP uses XML format. The file is fetched in comma-separated value (.csv) format and is stored in the Azure Data Lake Storage Gen2 (ADLS) as a raw file. The ADF then triggers the functions for cleansing the file, anonymization, and classification. The processed files will also be stored in the ADL. After the functions are ready, ADF will store the data in the Azure SQL database for the FTA's internal PowerBI reporting to access the data. Also, a maintenance view is built with Microsoft PowerApps for FTA officers so they can access the database directly and edit the classifications or edit the texts; for example, if anonymization has failed to identify personal information and text needs to be edited.

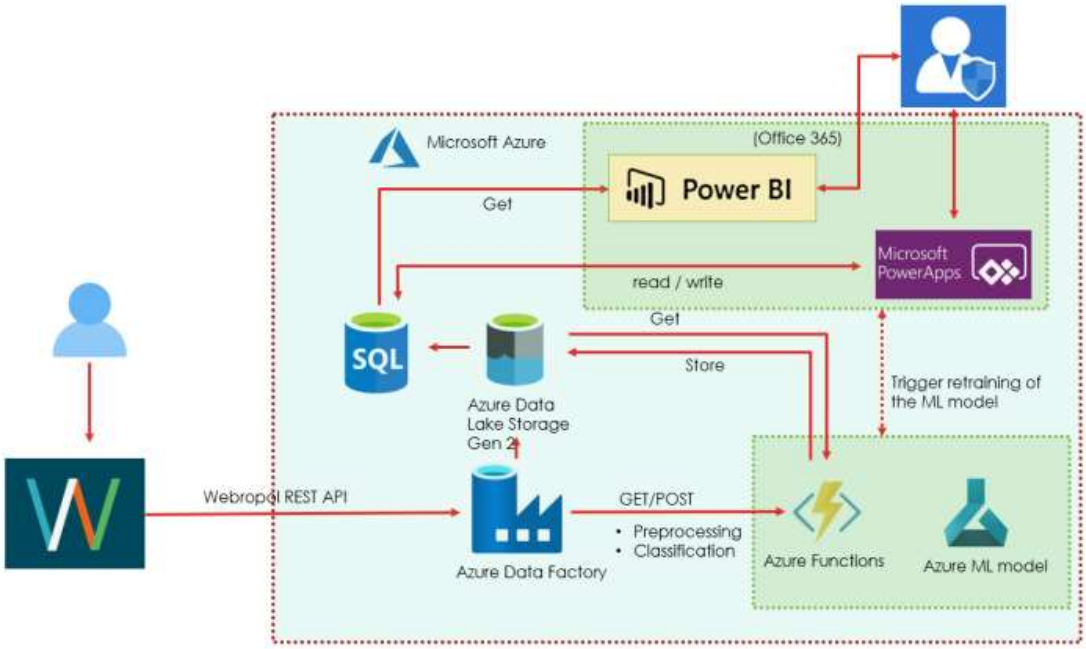


Figure 18 The Customer feedback processing pipeline in Microsoft Azure

6 CASE STUDIES

6.1 Introduction

In these case studies, the usefulness of the machine learning model for the text classification tasks is estimated. There are two principal research questions regarding the case studies:

1) can feedback be automatically classified with sufficient accuracy
2) what kind of classification method would perform the best in the task

This section wraps the data collection and splitting, preprocessing of the data, the model selection, development of both SVM and deep learning models, including the hyperparameter tuning. Finally, this section goes through the comparison of the results and the testing of how increasing the probability threshold affects the classification results.

6.2 Data

The dataset has 13801 rows of manually labeled feedback and 37 different categories. Data is gathered **from May 2021 until December 2021**, and there is also metadata available like timestamps and customer-defined categories. The feedback is gathered through Webropol, and users can access it through Vero.fi pages.

The data is not evenly distributed along with the categories; some categories got under ten or less feedback. The data is covered more precisely in the section 2.1 Data Source and Data. The testing dataset size is 2761 rows of feedback, and the training dataset size is 11 040 rows of feedback.

.

6.4 Text Preprocessing

In the case study, the corpus was already manually pseudonymized. Regex, a regular expression tool, was used to find and replace any other sensitive information, like decision numbers and social security numbers, if there would be some left. The text is then tokenized, and words are stemmed. After the stemming, the Tf-Idf weights are calculated for the stemmed words.

6.5 Model Development

6.5.1 Model Selection

For best model selection, there are a few criteria. The classification results need to be optimal, and there is a need to divide the classification results by their probabilities to be manually classified if the probability threshold is not met.

Model selection steps from Google's text classification guide (Figure 19) says that when the ratio of sample count per word count in a sample is under 1500, the use of more straightforward classification models is more effective such as Support Vector Machine (SVM), and if the ratio is more than 1500, the use of neural network models for classification yields better results. (Google, n.d.) Both neural network classification and support vector model was developed for the classifying task. Also, the fact that sentiment analysis or such is not needed in this classification task makes the use of simple machine learning models not ruled out. The dataset contains 11 040 samples, and the mean word count in a sample is 34.7, so the ratio is 318, lower than the threshold of 1500.

```
1. Calculate the number of samples/number of words per sample ratio.
2. If this ratio is less than 1500, tokenize the text as n-grams and use a
simple multi-layer perceptron (MLP) model to classify them (left branch in the
flowchart below):
  a. Split the samples into word n-grams; convert the n-grams into vectors.
  b. Score the importance of the vectors and then select the top 20K using the scores.
  c. Build an MLP model.
3. If the ratio is greater than 1500, tokenize the text as sequences and use a
   sepCNN model to classify them (right branch in the flowchart below):
  a. Split the samples into words; select the top 20K words based on their frequency.
  b. Convert the samples into word sequence vectors.
  c. If the original number of samples/number of words per sample ratio is less
     than 15K, using a fine-tuned pre-trained embedding with the sepCNN
     model will likely provide the best results.
4. Measure the model performance with different hyperparameter values to find
   the best model configuration for the dataset.
```

Figure 19 Model development steps for text classification by Google (Google, n.d.)

6.5.2 SVM Model Creation

The SVM model was trained with training dataset that contains 11 040 rows of feedback. Sci-kit learn package's model, svm.LinearSVC Linear Support Vector Classification model was selected as it was fast to train and performed nicely. It does not have the probability calculation available. The parameters available for LinearSVC model are presented in Table 4.

Table 4 LinearSVC parameters (scikitlearn, n.d.)

Parameter	Description
penalty {'l1', 'l2'}, default='l2'	Specifies the norm used in the penalization. The 'l2' penalty is the standard used in SVC. The 'l1' leads to coef_ vectors that are sparse.
loss {'hinge', 'squared_hinge'}, default='squared_hinge'	Specifies the loss function. 'hinge' is the standard SVM loss (used e.g. by the SVC class) while 'squared_hinge' is the square of the hinge loss. The combination of penalty='l1' and loss='hinge' is not supported.
dual bool, default=True	Select the algorithm to either solve the dual or primal optimization problem. Prefer dual=False when n_samples > n_features.
tol float, default=1e-4	• Tolerance for stopping criteria.
C float, default=1.0	Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive.
multi_class {'ovr', 'crammer_singer'}, default='ovr'	Determines the multi-class strategy if y contains more than two classes. "ovr" trains n_classes one-vs-rest classifiers, while "crammer_singer" optimizes a joint objective over all classes. While crammer_singer is interesting from a theoretical perspective as it is consistent, it is seldom used in practice as it rarely leads to better accuracy and is more expensive to compute. If "crammer_singer" is chosen, the options loss, penalty and dual will be ignored.
fit_intercept bool, default=True	Whether to calculate the intercept for this model. If set to false, no intercept will be used in calculations (i.e. data is expected to be already centered).
intercept_scaling float, default=1	When self.fit_intercept is True, instance vector x becomes [x, self.intercept_scaling], i.e. a "synthetic" feature with constant value equals to intercept_scaling is appended to the instance vector. The intercept becomes intercept_scaling * synthetic feature weight Note! the synthetic feature weight is subject to l1/l2 regularization as all other features. To lessen the effect of regularization on synthetic feature weight (and therefore on the intercept) intercept_scaling has to be increased.
class_weight dict or 'balanced', default=None	Set the parameter C of class i to class_weight[i]*C for SVC. If not given, all classes are supposed to have weight one. The "balanced" mode uses the values of y to automatically adjust weights inversely proportional to class frequencies in the input data as n_samples /

	(n_classes * np.bincount(y)).
verbose int, default=0	Enable verbose output. Note that this setting takes advantage of a per-process runtime setting in liblinear that, if enabled, may not work properly in a multithreaded context.
random_state int, RandomState instance or None, default=None	Controls the pseudo random number generation for shuffling the data for the dual coordinate descent (if dual=True). When dual=False the underlying implementation of LinearSVC is not random and random_state has no effect on the results. Pass an int for reproducible output across multiple function calls.
max_iter int, default=1000	The maximum number of iterations to be run.

The probabilities are needed to be able to sort the low probability scored feedback classifications to a separate category for the officer to sort them manually. To calculate the probabilities of the predictions for a model that has not such ability build-in, there is tools to do that. Such as CalibratedClassifierCV, which enables probability calibration with isotonic regression or logistic regression, provided with the Scikit-library package. (scikitlearn, n.d.). SVC from same scikit-learn package would have had the probability calculation available, but the training of it took longer, and the classification results were less than with LinearSVC. LinearSVC was fast to train and had better results.

6.5.3 Hyperparameter Tuning

Hyperparameter tuning is an essential part of model development to get the model function optimally. The development of the feedback classification model, the scikit-learn package's GridSearchCV is used for finding optimal hyperparameters. GridsearchCV uses the training data 5 folds and optimizes the hyperparameters using the training dataset given. 11 040 rows of data are used for training and cross-validation, leaving around 2761 rows for final testing.

Creating a pipeline for both Tf-Idf conversion with ngram selection and model hyperparameter selection, it is efficient to find the best hyperparameters for the model using the GridSearchCV.

For the pipeline, Tf-Idf converter is given values the ngram values of (1,1) and (1,2) are selected, and LinearSVC C values of (0.1, 1, and 10). These are the most critical parameters affecting the classification results.

Hyperparameter tuning can be time-consuming. The GridSearchCV does the K-folds Cross-Validation, and it is set to do five-folds, and every model fit is calculated with a different set of parameters, so we end up having two ngram-parameters and three C-parameters, and those are tested will five times with the five folds we selected for the

cross-validation. So, it takes 2x3x5 model fits, 30 fits in total. Model, or pipeline, with the best score, is naturally selected for use.

6.5.4 Probability calculation

The probability calculation is implemented in the model pipeline to provide probability percentages for the prediction in a scale of 0 to 1. The classification is the highest-scoring probability, but there is access to the other probabilities also.

The first aim of having these probabilities available is that some of the feedback can be automatically sorted into a manually processed basket. If the set threshold for the probability is not met, the feedback will fall to the manually processed category.

Also, in the customer feedback classification case, the three highest-scoring probability classifications are provided for the officer manually classifying the feedback in the manual category basket.

As said earlier, the CalibratedClassifiesCV is used to do the probability calculations, and as the name says, it also does cross-validation for the data. This is not mandatory since we have done the cross-validation already with the GridSearchCV.

6.5.5 SVM Implementation

Best estimator, or combination of Tf-Idf converter, model and probability estimator, is found in hyper parameter tuning with GridSearchCV, and it is selected to use in model. Best performance was achieved with hyperparameters: ngram (1,1) and C-value of 1.

6.5.6 Neural Network Model Implementation

TensorFlow/Keras neural network model was implemented with one hidden dense layer of 200 nodes and ‘tanh’ activation method. The output layer was implemented with ‘softmax’ activation. For optimization, an Adam-optimizer was used. The model was trained in 20 epocs with the same 11 040 row training data as the LinearSVC model. The preprocessing of the texts was also identical.

6.5.7 Testing and results

The testing data is 2761 rows. The test dataset contained 33 classes of the original 37, since some of the classes contained just a few feedback, and we needed the feedback from those categories to train the model. When the classification was performed on the testing data, 2761 rows of feedback and 37 classes, the classification accuracy was 72% (Figure 20) with the LinearSVC, and 65% using the neural network model. This is in line with the cross-validated results, which were the same.

This indicates that the LinearSVC performs better in this customer feedback classification task compared to Neural Network implementation.

	precision	recall	f1-score	support
ASIAKASTIEDOT	0.68	0.55	0.61	65
AUTOVEROTUS	1.00	0.17	0.29	18
CHAT	0.69	0.51	0.59	49
ESITÄYTETTY VEROILMOITUS	0.51	0.45	0.48	206
HENKILÖASIAKKAAT - VERO.FI	0.74	0.93	0.82	55
ILMIÖT	0.20	1.00	0.33	4
KIINTEISTÖVEROTUS	0.55	0.27	0.36	44
KÄYNTIASIOINTI	0.73	0.67	0.70	45
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.75	0.84	0.79	325
METSÄTALOUS	0.00	0.00	0.00	2
MUU KUIN PALAUTE	0.99	0.94	0.96	109
MUU SÄHKÖINEN ASIOINTIPALVELU	0.91	1.00	0.95	10
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	10
OMA-ALOITTEISET VEROT	0.61	0.55	0.58	55
OMAVERON VIESTI	0.76	0.52	0.62	25
OMAVERON YLEISET TOIMINNOT	0.76	0.84	0.80	786
PERINTÖ- JA LAHJAVEROTUS	0.47	0.43	0.45	21
POSTI	0.33	0.14	0.20	7
PUHELINPALVELU	0.79	0.90	0.84	185
RAKSI	0.88	0.70	0.78	10
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	3
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.63	0.70	0.66	234
TUNNISTAUTUMINEN TAI VALTUUDET	0.69	0.59	0.63	102
VALMISTEVEROTUS	0.00	0.00	0.00	3
VALVONTA	0.67	0.24	0.35	17
VARAINSIIRTOVEROTUS	0.89	0.55	0.68	29
VEROKORTTI JA ENNAKKOVEROT	0.75	0.76	0.75	274
VEROTUKSEN OIKAISUPYYNNÖT	0.45	0.48	0.47	21
YHTEISÖJEN VEROTUS	0.60	0.23	0.33	13
YRITYKSEN VEROTUS	1.00	0.07	0.13	14
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	15
accuracy			0.72	2761
macro avg	0.58	0.47	0.48	2761
weighted avg	0.71	0.72	0.71	2761

Figure 20 Precision, recall and f1-score for LinearSVC model with the testing dataset and 0-100% probabilities

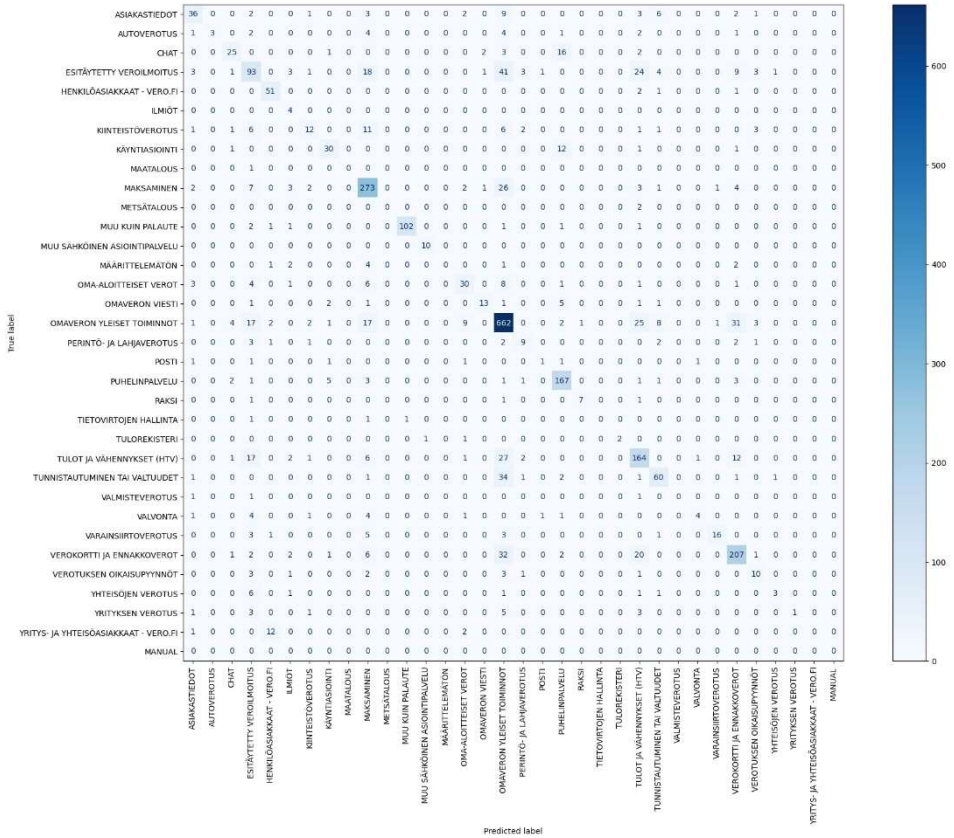


Figure 21 Confusion matrix for the LinearSVC model with the testing dataset and 0-100% probabilities.

The confusion matrix (Figure 21) shows how the predicted category performed compared to the true category.

Increasing the probability threshold increases the f1-score, and on the other hand, also increases the number of feedbacks sorted to the manually processed basket, the MANUAL-category in the confusion matrix.

For example, the classification results with 60%-100% probability scores, the f1-score has increased to 0.88. Since the manually processed feedback increases, while we increase the probability threshold, there is a fine line of usefulness to be found. With 50-100% probability, we get an f1-score of 0.83, and 27% of the incoming feedback will fall into the manually handled category. With the 60% probability, the amount of manually processed feedback will climb up to 42%. Please, check the attached appendices 0-9 for the confusion matrixes, f1-score, recall, and precision scores for each threshold limit in detail.

Table 5 below shows the change in the f1-score and the percentual amount of feedback sorted to the manually processed basket compared with the prediction probability percentage threshold change. The fine line between manually processed feedback and the gained f1-score enhancement is probably around the 50%-60% probability thresholds.

Table 5 Comparison of the probability thresholds effect on f1-score and the percentage of the feedbacks categorized to manual processing

Threshold %	f1-score	manual %
0-100	0.72	0
10-100	0.72	0
20-100	0.72	0.3
30-100	0.75	5.1
40-100	0.78	13.8
50-100	0.83	27.4
60-100	0.88	41.9
70-100	0.93	57.2
80-100	0.96	76.0
90-100	0.99	95.3

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

It seems apparent that machine learning can be used in the classification of feedback texts with proper results. Tf-Idf vectorized LinearSVC model seems to yield the best categorization. Turku NER is the state-of-the-art NER tagger currently for Finnish and can be used for pseudonymization along with the Regular Expression -tool. Also, the architecture for the feedback processing seems to work. The Azure Functions developed to process both the classification and pseudonymization worked, although Turku NER needed to have high system memory since the Tensorflow's demanded ten times more system memory than the size of the machine learning model to function on cloud services.

Most of the currently available market solutions use clustering or rule-based approach to classify texts, but this kind of finetuned and evolving machine learning solutions are not yet available. Also, the possibility to expand the customer feedback processing to other areas or having multiple streams of texts is a clear bonus for the design.

Back to the three research questions stated in the beginning:

1. What could be an appropriate method for classifying customer feedback?
2. What would be a proper architecture for customer feedback processing in Microsoft Azure?
3. What could be a practical method or tool for anonymization/pseudonymization of the customer feedback?

For the first question, a suitable customer feedback classification method seems to be the Tf-Idf weighted LinearSVC model with prediction calculations.

For the second question, the architecture for the feedback processing is suitable. The Azure Functions seem to be useful; they work in small scale for batch processing of incoming data, but when the feedback processing evolves to cover more of the incoming

feedback streams, the use of Azure Functions might be too consuming, and other options like processing the data in Azure Databricks could be the way to go.

The third question. It is evident that the Turku NER gives the best results for NER tagging, and the tagged entities can then be removed from the corpus. NER tagged pseudonymization with regular expression removal will probably be the optimal choice for pseudonymization.

To further develop the feedback processing, we could use the Microsoft Azure Text Analytics tools, like sentiment analysis, to add more intelligence to the feedback processing. Also, the use the generative pre-trained transformers like GPT-J would be worth exploring, and they would provide the most opportunities in the feedback and customer interaction field in the future.

The thesis didn't dive into the areas of security or ethics or purging of gathered data in this thesis since they would be entirely new chapters to be written in these areas, or new thesis even, but I want to state that the areas are not forgotten and have been under consideration throughout the process of developing the customer feedback processing with machine learning.

FINAL WORDS

There is a place for everything in this universe; aligning and tuning in to it is the key to its secrets. Thank you all for supporting with tuning in, aligning, and making dreams into reality. Thank you, Amin Majd, Leonardo Espinosa-Leal, Magnus Westerlund, Andrej Scherbakov-Parland, Kaj-Mikael Björk and Roberto V. Zicari from Arcada for your significant contribution to this growth journey. Thank you, Jarno Tuimala. I have no words to express my gratitude. Thank you, Johanna Lattu, Janne Myyry, Petri Pyhältö, Antti Honkasalo, Jussi Miettinen, and Mirkka Raninen from Finnish Tax Administration, for all you've done, also for the support and encouragement you've given. Without your huge contribution, this thesis would not have been written. Thank you, Karri Pulkkinen, Tanja Sireeni, Antti Suanto from our project team; having the same goal is substantial. Thank you Katja Yli-Krekola, Hannu Rauhala, Päivi Nummelin and Marko Pynnönen for all the daily conversations during the winter. Your company has made this journey so pleasant and relaxed. Thank you, friends, mentors, and co-workers Marika Tolkkila, Tiina Karjalainen, Mittal Mehta, Pasi Kaunisvesi, Jussi Ylhäisi and Kalle Aaltonen for all the support you gave. I also have deep gratitude for my spouse Sakari for all the support during the times of Covid. You have taken care of our three girls as I invested my time and energy studying and working full time. I feel incredibly grateful for all of you.

There is a numinous [mind] naturally residing within [有神自在身];
One moment it goes, the next it comes,
And no one is able to conceive of it.
If you lose it, you are inevitably disordered;
If you attain it, you are inevitably well ordered.
Diligently clean out its lodging place [敬除其舍]
And its vital essence will naturally arrive [精將自來].
Still your attempts to imagine and conceive of it.
Relax your efforts to reflect on and control it.
Be reverent and diligent
And its vital essence will naturally stabilize.
Grasp it and don't let go
Then the eyes and ears won't overflow
And the mind will have nothing else to seek.
When a properly aligned mind resides within you [正心在中],
The myriad things will be seen in their proper perspective

*(Original Tao: inward training (*nei-yeh*) and the foundations of Taoist mysticism, Harold D. Roth, 1999)*

8 REFERENCES

Ajanki, A., 2021. Evaluation of Finnish named-entity recognition (NER) services [WWW Document]. URL https://aajanki.github.io/fi-ner-eval/ (accessed 1.16.22).

Bex, T., 2021. Comprehensive Guide on Multiclass Classification Metrics [WWW Document]. Medium. URL https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd (accessed 12.29.21).

Claesen, M., De Moor, B., 2015. Hyperparameter Search in Machine Learning. arXiv:1502.02127 [cs, stat].

Concepts — ML Glossary documentation [WWW Document], n.d. URL https://ml-cheatsheet.readthedocs.io/en/latest/nn_concepts.html (accessed 12.29.21).

Copeland, M., 2016. The Difference Between AI, Machine Learning, and Deep Learning? | NVIDIA Blog [WWW Document]. The Official NVIDIA Blog. URL https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/ (accessed 12.29.21).

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M.G., Pineau, J., 2018. An Introduction to Deep Reinforcement Learning. FNT in Machine Learning 11, 219–354. https://doi.org/10.1561/2200000071

Ganesan, K., 2018. What are N-Grams? Kavita Ganesan, PhD. URL https://kavita-ganesan.com/what-are-n-grams/ (accessed 12.29.21).

Ghahramani, Z., 2004. Unsupervised Learning, in: Bousquet, O., von Luxburg, U., Rätsch, G. (Eds.), Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 72–112. https://doi.org/10.1007/978-3-540-28650-9_5

Google, n.d. Step 2.5: Choose a Model | Text classification guide [WWW Document]. URL https://developers.google.com/machine-learning/guides/text-classification/step-2-5?hl=fi (accessed 10.5.21).

Jabeen, H., 2018. Stemming and Lemmatization in Python [WWW Document]. DataCamp Community. URL https://www.datacamp.com/community/tutorials/stemming-lemmatization-python (accessed 12.29.21).

Jayaswal, V., 2020. Performance Metrics: Confusion matrix, Precision, Recall, and F1 Score [WWW Document]. Medium. URL https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262 (accessed 12.29.21).

Joachims, T., 1998. Text categorization with Support Vector Machines: Learning with many relevant features, in: Nédellec, C., Rouveirol, C. (Eds.), Machine Learning: ECML-98, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 137–142. https://doi.org/10.1007/BFb0026683

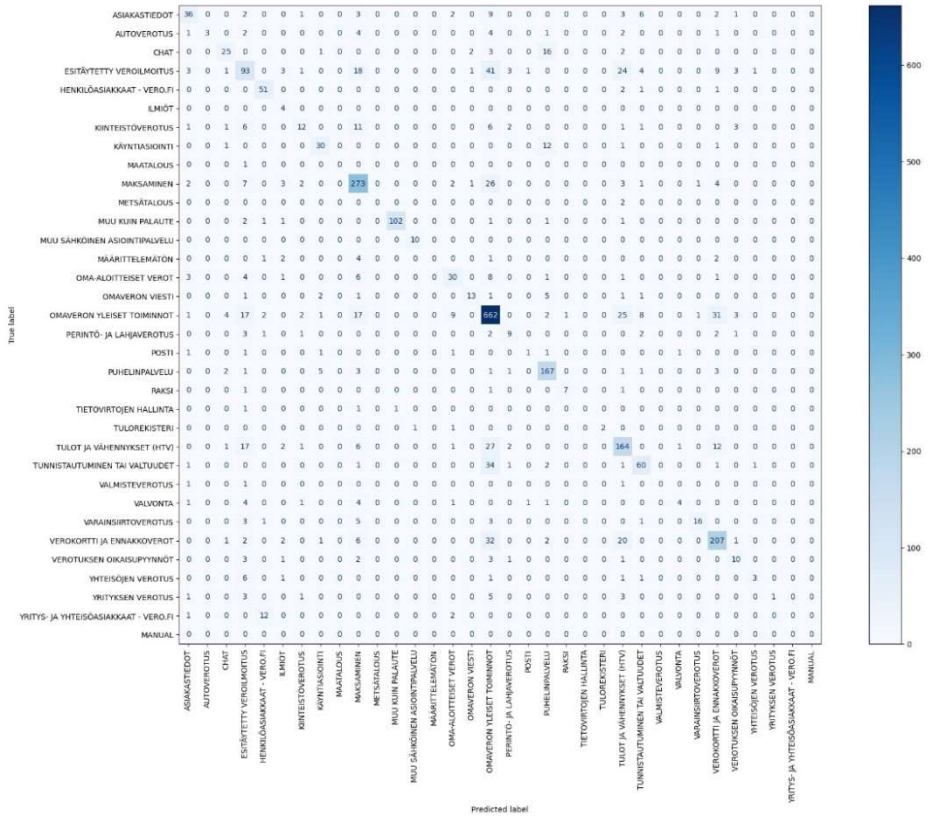
Khalusova, M., n.d. Maria Khalusova | Machine Learning Model Evaluation Metrics part 2: Multi-class classification [WWW Document]. URL

https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/ (accessed 12.29.21).
Kingma, D.P., Mohamed, S., Jimenez Rezende, D., Welling, M., 2014. Semi-supervised Learning with Deep Generative Models, in: Advances in Neural Information Processing Systems. Curran Associates, Inc.
Kinnunen, T., n.d. Figure 9: Principle of support vector machine (SVM). A maximummargin... [WWW Document]. ResearchGate. URL https://www.researchgate.net/figure/Principle-of-support-vector-machine-SVM-A-maximummargin-hyperplane-that-separates-the_fig6_222682226 (accessed 1.16.22).
Kumar, A., 2021. K-Fold Cross Validation - Python Example. Data Analytics. URL https://vitalflux.com/k-fold-cross-validation-python-example/ (accessed 12.29.21).
Liu, B., 2011. Supervised Learning, in: Liu, B. (Ed.), Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Data-Centric Systems and Applications. Springer, Berlin, Heidelberg, pp. 63–132. https://doi.org/10.1007/978-3-642-19460-3_3
Lovins, J.B., 1968. Development of a Stemming Algorithm*.
Luoma, J., Oinonen, M., Pyykönen, M., Laippala, V., Pyysalo, S., 2020. A Broad-coverage Corpus for Finnish Named Entity Recognition, in: Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France, pp. 4615–4624.
Microsoft, n.d. Introduction to Blob (object) storage - Azure Storage [WWW Document]. URL https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction (accessed 1.16.22a).
Microsoft, n.d. What is the Azure SQL Database service? - Azure SQL Database [WWW Document]. URL https://docs.microsoft.com/en-us/azure/azure-sql/database/sql-database-paas-overview (accessed 1.16.22b).
Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality.
NLTK :: nltk.stem.snowball [WWW Document], n.d. URL https://www.nltk.org/_modules/nltk/stem/snowball.html (accessed 12.29.21).
OpenAI, 2021. OpenAI Codex. Wikipedia.
Oppermann, A., 2020. What is Deep Learning and How does it work? [WWW Document]. Medium. URL https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac (accessed 2.12.22).
Pilault, J., Elhattami, A., Pal, C., 2021. Conditionally Adaptive Multi-Task Learning: Improving Transfer Learning in NLP Using Fewer Parameters & Less Data. arXiv:2009.09139 [cs, stat].
Porter, M., n.d. Snowball: A language for stemming algorithms [WWW Document]. URL http://snowball.tartarus.org/texts/introduction.html (accessed 12.29.21).
Purva, H., 2020. BoW Model and TF-IDF For Creating Feature From Text. Analytics Vidhya. URL https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/ (accessed 10.20.21).
Samuel Morland, 2022. . Wikipedia.
scikitlearn, n.d. sklearn.svm.LinearSVC [WWW Document]. scikit-learn. URL https://scikit-learn/stable/modules/generated/sklearn.svm.LinearSVC.html (accessed 1.15.22a).
scikitlearn, n.d. sklearn.calibration.CalibratedClassifierCV [WWW Document]. scikit-learn. URL https://scikit-

learn/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html (accessed 1.15.22b).
Shah, T., 2020. About Train, Validation and Test Sets in Machine Learning [WWW Document]. Medium. URL https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7 (accessed 1.4.22).
Smith, M., 2019. IaaS, PaaS, SaaS: A beginner's guide to the cloud [WWW Document]. URL https://www.chorus.co/resources/news/iaas-paas-saas-a-beginners-guide-to-the-cloud (accessed 2.14.22).
Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., Pyysalo, S., 2019. Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., ... & Pyysalo, S. (2019). Multilingual is not enough: BERT for Finnish. arXiv preprint arXiv:1912.07076.
Whitfield, D., 2021. Using GPT-2 to Create Synthetic Data to Improve the Prediction Performance of NLP Machine Learning Classification Models. arXiv:2104.10658 [cs].
Wikipedia contributors, 2021a. Transfer learning. Wikipedia.
Wikipedia contributors, 2021b. Stemming. Wikipedia.

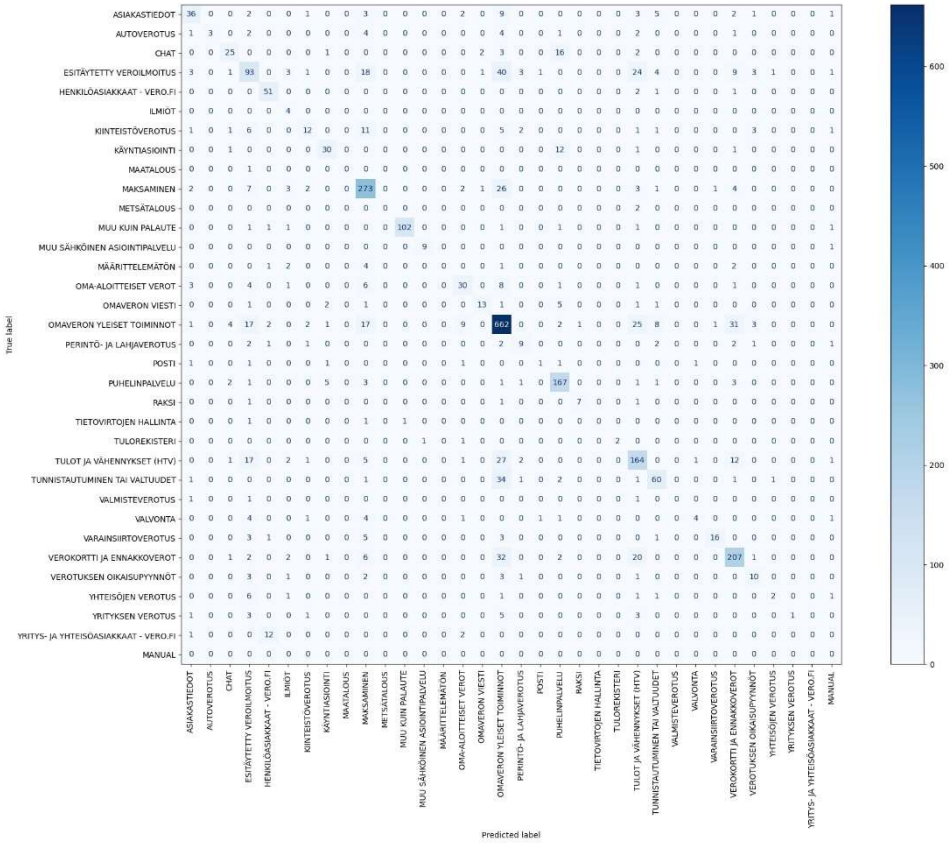
9 APPENDICES

APPENDIX 0 The confusion matrix and classification score for probability threshold of 0%



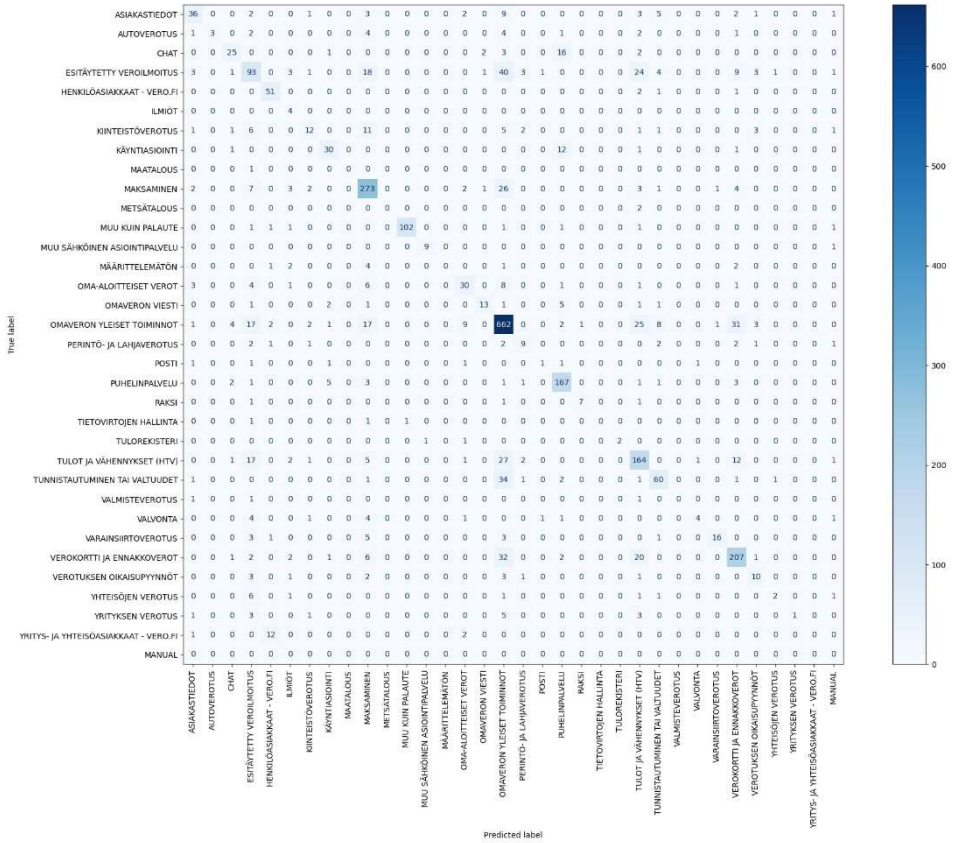
	precision	recall	f1-score	support
ASIAKASTIEDOT	0.68	0.55	0.61	65
AUTOVEROTUS	1.00	0.17	0.29	18
CHAT	0.69	0.51	0.59	49
ESITÄYTETTY VEROILMOITUS	0.51	0.45	0.48	206
HENKILÖASIAKKAAT - VERO.FI	0.74	0.93	0.82	55
ILMIÖT	0.20	1.00	0.33	4
KIINTEISTÖVEROTUS	0.55	0.27	0.36	44
KÄYNTIASIOINTI	0.73	0.67	0.70	45
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.75	0.84	0.79	325
METSÄTALOUS	0.00	0.00	0.00	2
MUU KUIN PALAUTE	0.99	0.94	0.96	109
MUU SÄHKÖINEN ASIOINTIPALVELU	0.91	1.00	0.95	10
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	10
OMA-ALOITTEISET VEROT	0.61	0.55	0.58	55
OMAVERON VIESTI	0.76	0.52	0.62	25
OMAVERON YLEISET TOIMINNOT	0.76	0.84	0.80	786
PERINTÖ- JA LAHJAVEROTUS	0.47	0.43	0.45	21
POSTI	0.33	0.14	0.20	7
PUHELINPALVELU	0.79	0.90	0.84	185
RAKSI	0.88	0.70	0.78	10
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	3
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.63	0.70	0.66	234
TUNNISTAUTUMINEN TAI VALTUUDET	0.69	0.59	0.63	102
VALMISTEVEROTUS	0.00	0.00	0.00	3
VALVONTA	0.67	0.24	0.35	17
VARAINSIIRTOVEROTUS	0.89	0.55	0.68	29
VEROKORTTI JA ENNAKKOVEROT	0.75	0.76	0.75	274
VEROTUKSEN OIKAISUPYYNNÖT	0.45	0.48	0.47	21
YHTEISÖJEN VEROTUS	0.60	0.23	0.33	13
YRITYKSEN VEROTUS	1.00	0.07	0.13	14
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	15
accuracy			0.72	2761
macro avg	0.58	0.47	0.48	2761
weighted avg	0.71	0.72	0.71	2761

APPENDIX 1 The confusion matrix and classification score for probability threshold of 10%



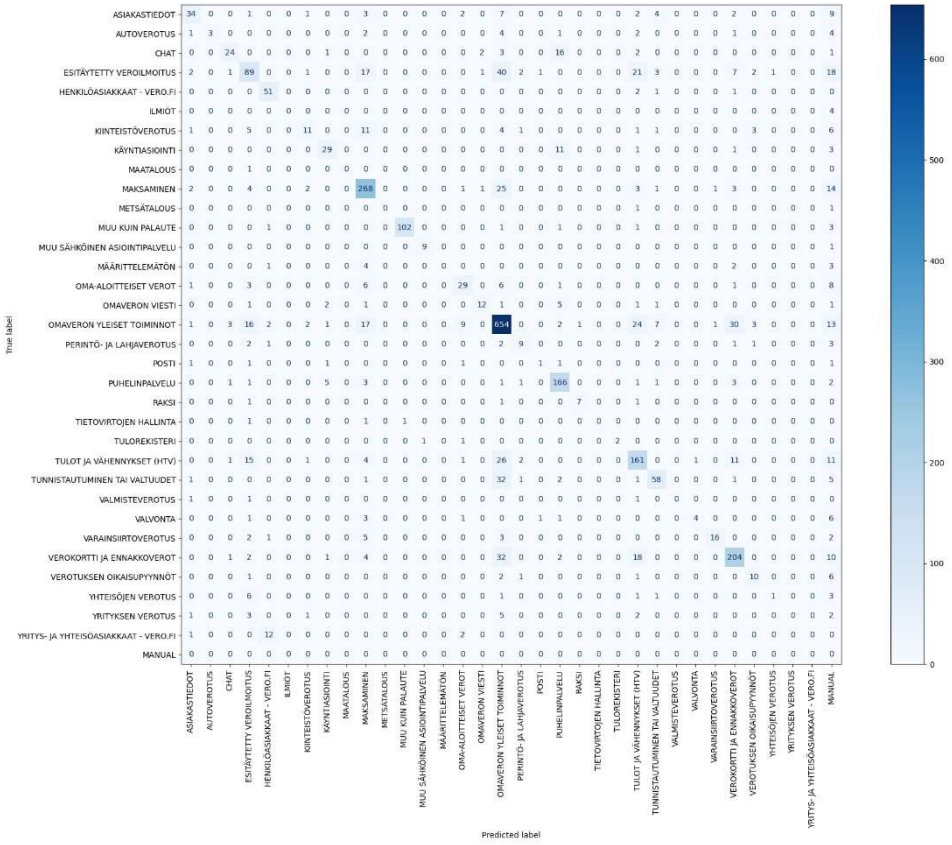
	precision	recall	f1-score	support
ASIAKASTIEDOT	0.68	0.55	0.61	65
AUTOVEROTUS	1.00	0.17	0.29	18
CHAT	0.69	0.51	0.59	49
ESITÄYTETTY VEROILMOITUS	0.51	0.45	0.48	206
HENKILÖASIAKKAAT - VERO.FI	0.74	0.93	0.82	55
ILMIÖT	0.20	1.00	0.33	4
KIINTEISTÖVEROTUS	0.55	0.27	0.36	44
KÄYNTIASIOINTI	0.73	0.67	0.70	45
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.75	0.84	0.79	325
METSÄTALOUS	0.00	0.00	0.00	2
MUU KUIN PALAUTE	0.99	0.94	0.96	109
MUU SÄHKÖINEN ASIOINTIPALVELU	0.91	1.00	0.95	10
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	10
OMA-ALOITTEISET VEROT	0.61	0.55	0.58	55
OMAVERON VIESTI	0.76	0.52	0.62	25
OMAVERON YLEISET TOIMINNOT	0.76	0.84	0.80	786
PERINTÖ- JA LAHJAVEROTUS	0.47	0.43	0.45	21
POSTI	0.33	0.14	0.20	7
PUHELINPALVELU	0.79	0.90	0.84	185
RAKSI	0.88	0.70	0.78	10
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	3
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.63	0.70	0.66	234
TUNNISTAUTUMINEN TAI VALTUUDET	0.69	0.59	0.63	102
VALMISTEVEROTUS	0.00	0.00	0.00	3
VALVONTA	0.67	0.24	0.35	17
VARAINSIIRTOVEROTUS	0.89	0.55	0.68	29
VEROKORTTI JA ENNAKKOVEROT	0.75	0.76	0.75	274
VEROTUKSEN OIKAISUPYYNNÖT	0.45	0.48	0.47	21
YHTEISÖJEN VEROTUS	0.60	0.23	0.33	13
YRITYKSEN VEROTUS	1.00	0.07	0.13	14
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	15
accuracy			0.72	2761
macro avg	0.58	0.47	0.48	2761
weighted avg	0.71	0.72	0.71	2761

APPENDIX 2 The confusion matrix and classification score for probability threshold of 20%



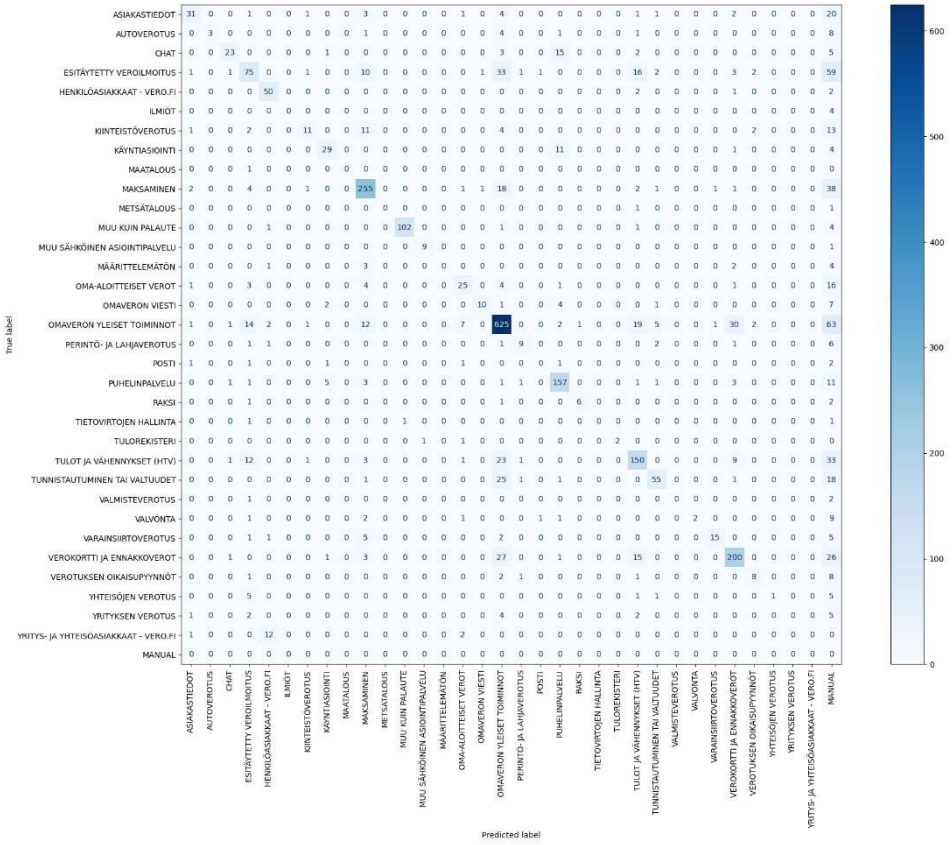
	precision	recall	f1-score	support
ASIAKASTIEDOT	0.69	0.56	0.62	64
AUTOVEROTUS	1.00	0.17	0.29	18
CHAT	0.69	0.51	0.59	49
ESITÄYTETTY VEROILMOITUS	0.52	0.45	0.48	205
HENKILÖASIAKKAAT - VERO.FI	0.74	0.93	0.82	55
ILMIÖT	0.20	1.00	0.33	4
KIINTEISTÖVEROTUS	0.55	0.28	0.37	43
KÄYNTIASIOINTI	0.73	0.67	0.70	45
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.75	0.84	0.79	325
METSÄTALOUS	0.00	0.00	0.00	2
MUU KUIN PALAUTE	0.99	0.94	0.97	108
MUU SÄHKÖINEN ASIOINTIPALVELU	0.90	1.00	0.95	9
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	10
OMA-ALOITTEISET VEROT	0.61	0.55	0.58	55
OMAVERON VIESTI	0.76	0.52	0.62	25
OMAVERON YLEISET TOIMINNOT	0.76	0.84	0.80	786
PERINTÖ- JA LAHJAVEROTUS	0.47	0.45	0.46	20
POSTI	0.33	0.14	0.20	7
PUHELINPALVELU	0.79	0.90	0.84	185
RAKSI	0.88	0.70	0.78	10
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	3
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.63	0.70	0.66	233
TUNNISTAUTUMINEN TAI VALTUUDET	0.70	0.59	0.64	102
VALMISTEVEROTUS	0.00	0.00	0.00	3
VALVONTA	0.67	0.25	0.36	16
VARAINSIIRTOVEROTUS	0.89	0.55	0.68	29
VEROKORTTI JA ENNAKKOVEROT	0.75	0.76	0.75	274
VEROTUKSEN OIKAISUPYYNNÖT	0.45	0.48	0.47	21
YHTEISÖJEN VEROTUS	0.50	0.17	0.25	12
YRITYKSEN VEROTUS	1.00	0.07	0.13	14
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	15
accuracy			0.72	2752
macro avg	0.57	0.47	0.48	2752
weighted avg	0.72	0.72	0.71	2752

APPENDIX 3 The confusion matrix and classification score for probability threshold of 30%



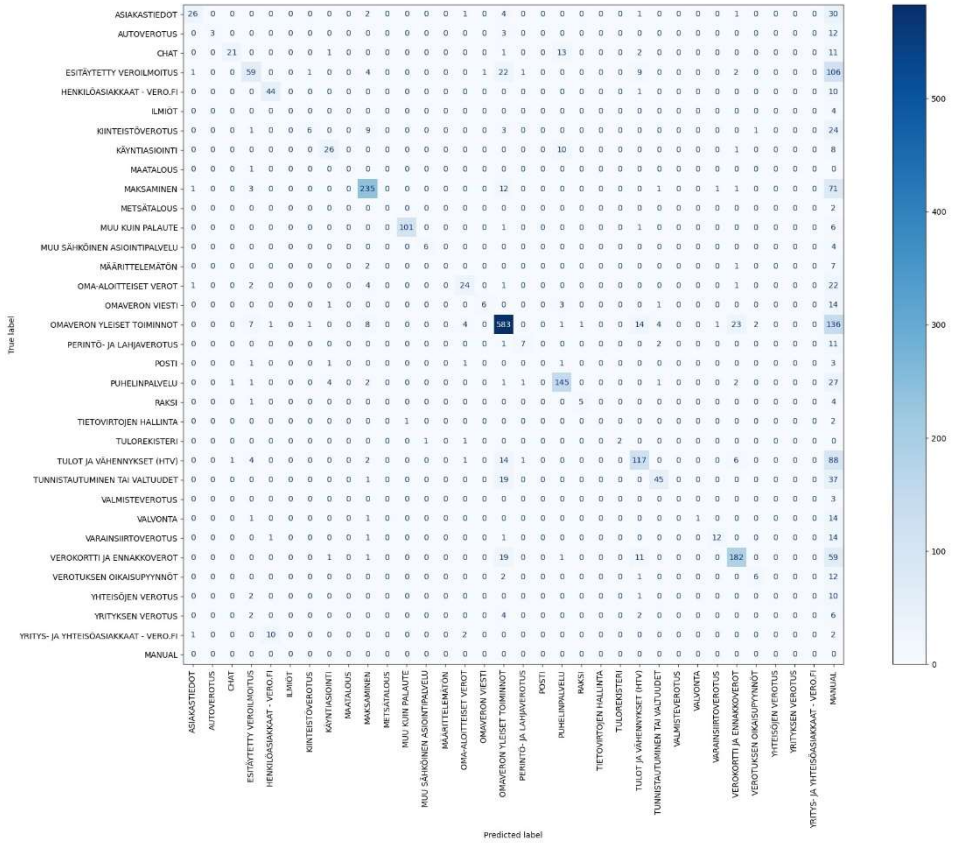
	precision	recall	f1-score	support
ASIAKASTIEDOT	0.72	0.61	0.66	56
AUTOVEROTUS	1.00	0.21	0.35	14
CHAT	0.77	0.50	0.61	48
ESITÄYTETTY VEROILMOITUS	0.57	0.47	0.52	188
HENKILÖASIAKKAAT - VERO.FI	0.74	0.93	0.82	55
KIINTEISTÖVEROTUS	0.58	0.29	0.39	38
KÄYNTIASIOINTI	0.72	0.69	0.71	42
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.77	0.86	0.81	311
METSÄTALOUS	0.00	0.00	0.00	1
MUU KUIN PALAUTE	0.99	0.96	0.98	106
MUU SÄHKÖINEN ASIOINTIPALVELU	0.90	1.00	0.95	9
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	7
OMA-ALOITTEISET VEROT	0.62	0.62	0.62	47
OMAVERON VIESTI	0.75	0.50	0.60	24
OMAVERON YLEISET TOIMINNOT	0.77	0.85	0.81	773
PERINTÖ- JA LAHJAVEROTUS	0.53	0.50	0.51	18
POSTI	0.33	0.17	0.22	6
PUHELINPALVELU	0.79	0.91	0.85	183
RAKSI	0.88	0.70	0.78	10
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	3
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.65	0.72	0.68	223
TUNNISTAUTUMINEN TAI VALTUUDET	0.72	0.60	0.66	97
VALMISTEVEROTUS	0.00	0.00	0.00	3
VALVONTA	0.80	0.36	0.50	11
VARAINSIIRTOVEROTUS	0.89	0.59	0.71	27
VEROKORTTI JA ENNAKKOVEROT	0.76	0.77	0.77	264
VEROTUKSEN OIKAISUPYYNNÖT	0.53	0.67	0.59	15
YHTEISÖJEN VEROTUS	0.50	0.10	0.17	10
YRITYKSEN VEROTUS	0.00	0.00	0.00	12
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	15
accuracy			0.75	2621
macro avg	0.57	0.47	0.50	2621
weighted avg	0.73	0.75	0.73	2621

APPENDIX 4 The confusion matrix and classification score for probability threshold of 40%



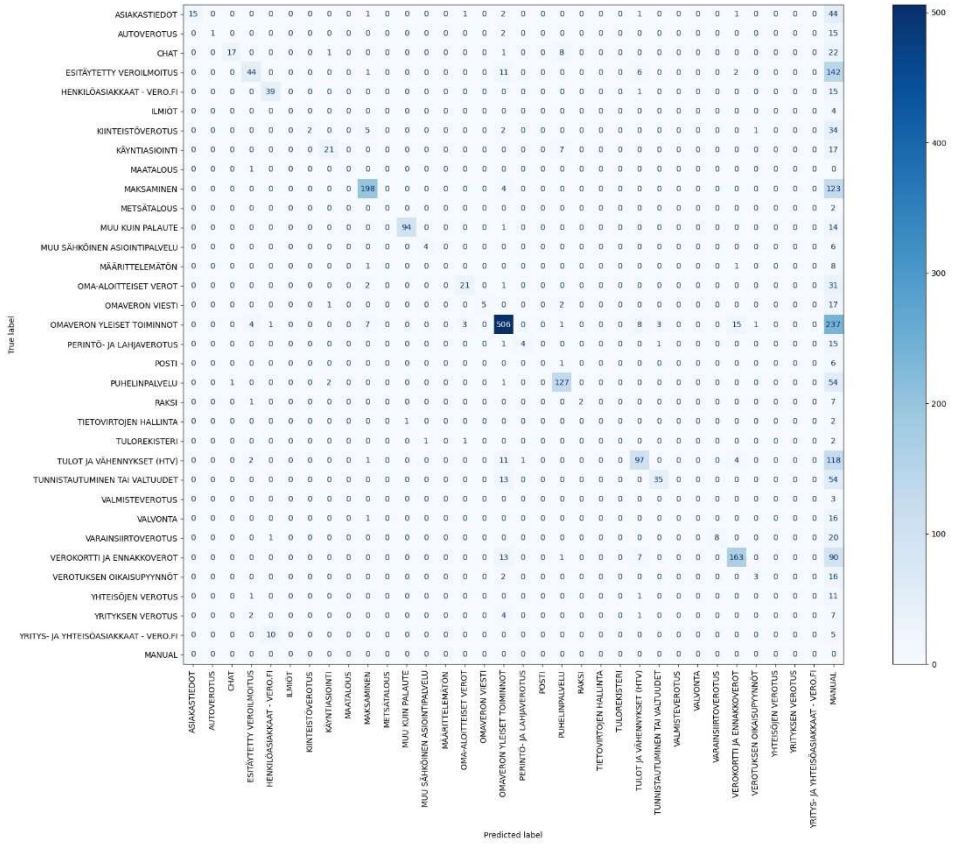
	precision	recall	f1-score	support
ASIAKASTIEDOT	0.78	0.69	0.73	45
AUTOVEROTUS	1.00	0.30	0.46	10
CHAT	0.82	0.52	0.64	44
ESITÄYTETTY VEROILMOITUS	0.59	0.51	0.55	147
HENKILÖASIAKKAAT - VERO.FI	0.74	0.94	0.83	53
KIINTEISTÖVEROTUS	0.69	0.35	0.47	31
KÄYNTIASIOINTI	0.74	0.71	0.72	41
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.81	0.89	0.85	287
METSÄTALOUS	0.00	0.00	0.00	1
MUU KUIN PALAUTE	0.99	0.97	0.98	105
MUU SÄHKÖINEN ASIOINTIPALVELU	0.90	1.00	0.95	9
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	6
OMA-ALOITTEISET VEROT	0.62	0.64	0.63	39
OMAVERON VIESTI	0.83	0.56	0.67	18
OMAVERON YLEISET TOIMINNOT	0.80	0.86	0.83	723
PERINTÖ- JA LAHJAVEROTUS	0.64	0.60	0.62	15
POSTI	0.00	0.00	0.00	5
PUHELINPALVELU	0.81	0.90	0.85	174
RAKSI	0.86	0.75	0.80	8
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	2
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.70	0.75	0.72	201
TUNNISTAUTUMINEN TAI VALTUUDET	0.80	0.65	0.72	84
VALMISTEVEROTUS	0.00	0.00	0.00	1
VALVONTA	1.00	0.25	0.40	8
VARAINSIIRTOVEROTUS	0.88	0.62	0.73	24
VEROKORTTI JA ENNAKKOVEROT	0.78	0.81	0.80	248
VEROTUKSEN OIKAISUPYYNNÖT	0.57	0.62	0.59	13
YHTEISÖJEN VEROTUS	1.00	0.12	0.22	8
YRITYKSEN VEROTUS	0.00	0.00	0.00	9
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	15
accuracy			0.78	2379
macro avg	0.60	0.49	0.51	2379
weighted avg	0.77	0.78	0.77	2379

APPENDIX 5 The confusion matrix and classification score for probability threshold of 50%



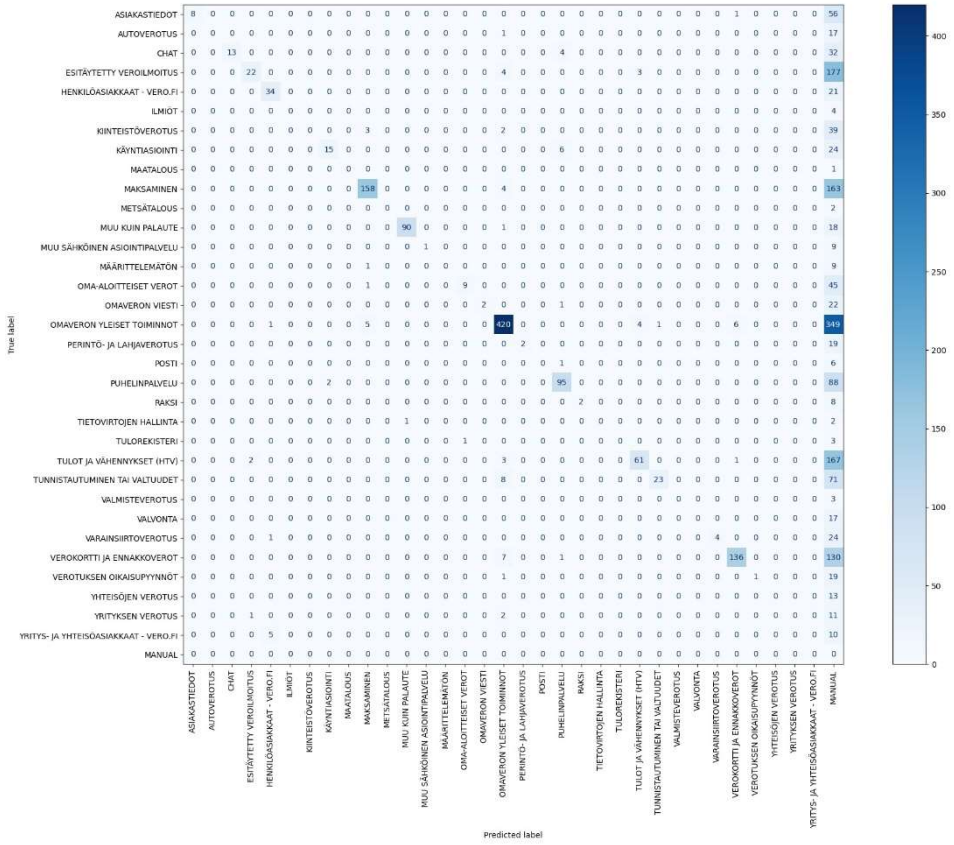
	precision	recall	f1-score	support
ASIAKASTIEDOT	0.87	0.74	0.80	35
AUTOVEROTUS	1.00	0.50	0.67	6
CHAT	0.91	0.55	0.69	38
ESITÄYTETTY VEROILMOITUS	0.69	0.59	0.64	100
HENKILÖASIAKKAAT - VERO.FI	0.79	0.98	0.87	45
KIINTEISTÖVEROTUS	0.75	0.30	0.43	20
KÄYNTIASIOINTI	0.76	0.70	0.73	37
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.86	0.93	0.89	254
MUU KUIN PALAUTE	0.99	0.98	0.99	103
MUU SÄHKÖINEN ASIOINTIPALVELU	0.86	1.00	0.92	6
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	3
OMA-ALOITTEISET VEROT	0.71	0.73	0.72	33
OMAVERON VIESTI	0.86	0.55	0.67	11
OMAVERON YLEISET TOIMINNOT	0.84	0.90	0.87	650
PERINTÖ- JA LAHJAVEROTUS	0.70	0.70	0.70	10
POSTI	0.00	0.00	0.00	4
PUHELINPALVELU	0.83	0.92	0.87	158
RAKSI	0.83	0.83	0.83	6
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	1
TULOREKISTERI	1.00	0.50	0.67	4
TULOT JA VÄHENNYKSET (HTV)	0.73	0.80	0.76	146
TUNNISTAUTUMINEN TAI VALTUUDET	0.83	0.69	0.76	65
VALVONTA	1.00	0.33	0.50	3
VARAINSIIRTOVEROTUS	0.86	0.80	0.83	15
VEROKORTTI JA ENNAKKOVEROT	0.83	0.85	0.84	215
VEROTUKSEN OIKAISUPYYNNÖT	0.67	0.67	0.67	9
YHTEISÖJEN VEROTUS	0.00	0.00	0.00	3
YRITYKSEN VEROTUS	0.00	0.00	0.00	8
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	13
accuracy			0.83	2002
macro avg	0.64	0.55	0.58	2002
weighted avg	0.82	0.83	0.82	2002

APPENDIX 6 The confusion matrix and classification score for probability threshold of 60%



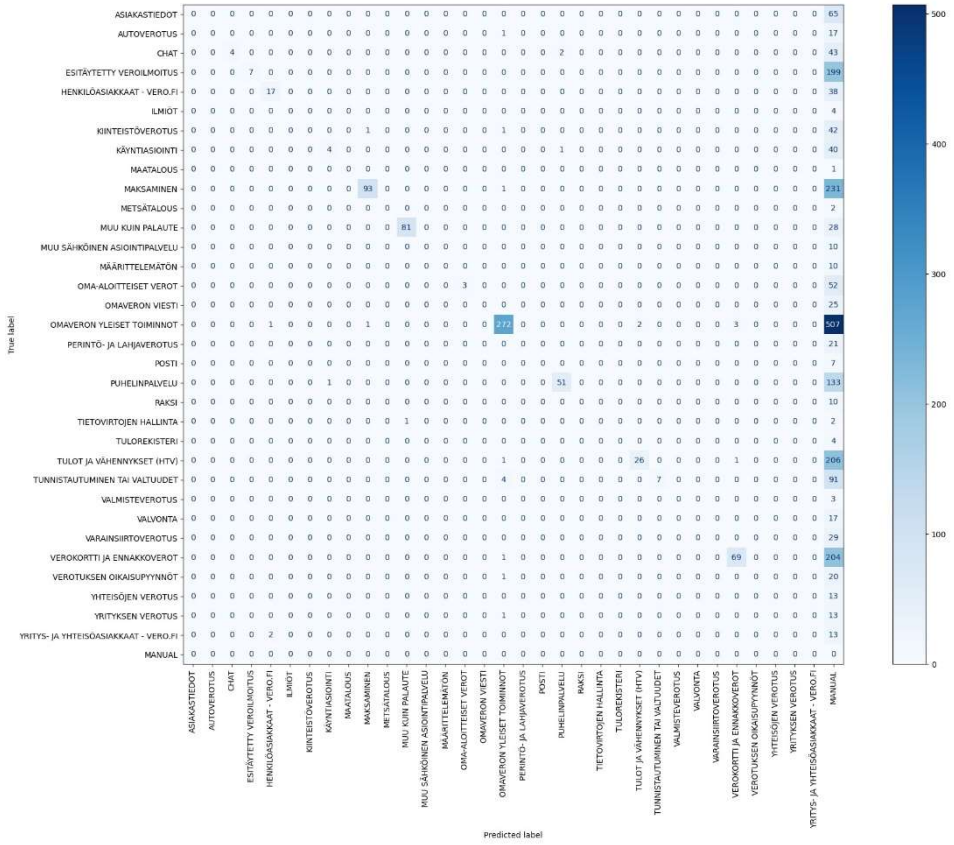
	precision	recall	f1-score	support
ASIAKASTIEDOT	1.00	0.71	0.83	21
AUTOVEROTUS	1.00	0.33	0.50	3
CHAT	0.94	0.63	0.76	27
ESITÄYTETTY VEROILMOITUS	0.80	0.69	0.74	64
HENKILÖASIAKKAAT - VERO.FI	0.76	0.97	0.86	40
KIINTEISTÖVEROTUS	1.00	0.20	0.33	10
KÄYNTIASIOINTI	0.84	0.75	0.79	28
MAATALOUS	0.00	0.00	0.00	1
MAKSAMINEN	0.91	0.98	0.95	202
MUU KUIN PALAUTE	0.99	0.99	0.99	95
MUU SÄHKÖINEN ASIOINTIPALVELU	0.80	1.00	0.89	4
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	2
OMA-ALOITTEISET VEROT	0.81	0.88	0.84	24
OMAVERON VIESTI	1.00	0.62	0.77	8
OMAVERON YLEISET TOIMINNOT	0.88	0.92	0.90	549
PERINTÖ- JA LAHJAVEROTUS	0.80	0.67	0.73	6
POSTI	0.00	0.00	0.00	1
PUHELINPALVELU	0.86	0.97	0.91	131
RAKSI	1.00	0.67	0.80	3
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	1
TULOREKISTERI	0.00	0.00	0.00	2
TULOT JA VÄHENNYKSET (HTV)	0.80	0.84	0.82	116
TUNNISTAUTUMINEN TAI VALTUUDET	0.90	0.73	0.80	48
VALVONTA	0.00	0.00	0.00	1
VARAINSIIRTOVEROTUS	1.00	0.89	0.94	9
VEROKORTTI JA ENNAKKOVEROT	0.88	0.89	0.88	184
VEROTUKSEN OIKAISUPYYNNÖT	0.60	0.60	0.60	5
YHTEISÖJEN VEROTUS	0.00	0.00	0.00	2
YRITYKSEN VEROTUS	0.00	0.00	0.00	7
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	10
accuracy			0.88	1604
macro avg	0.62	0.53	0.55	1604
weighted avg	0.86	0.88	0.87	1604

APPENDIX 7 The confusion matrix and classification score for probability threshold of 70%



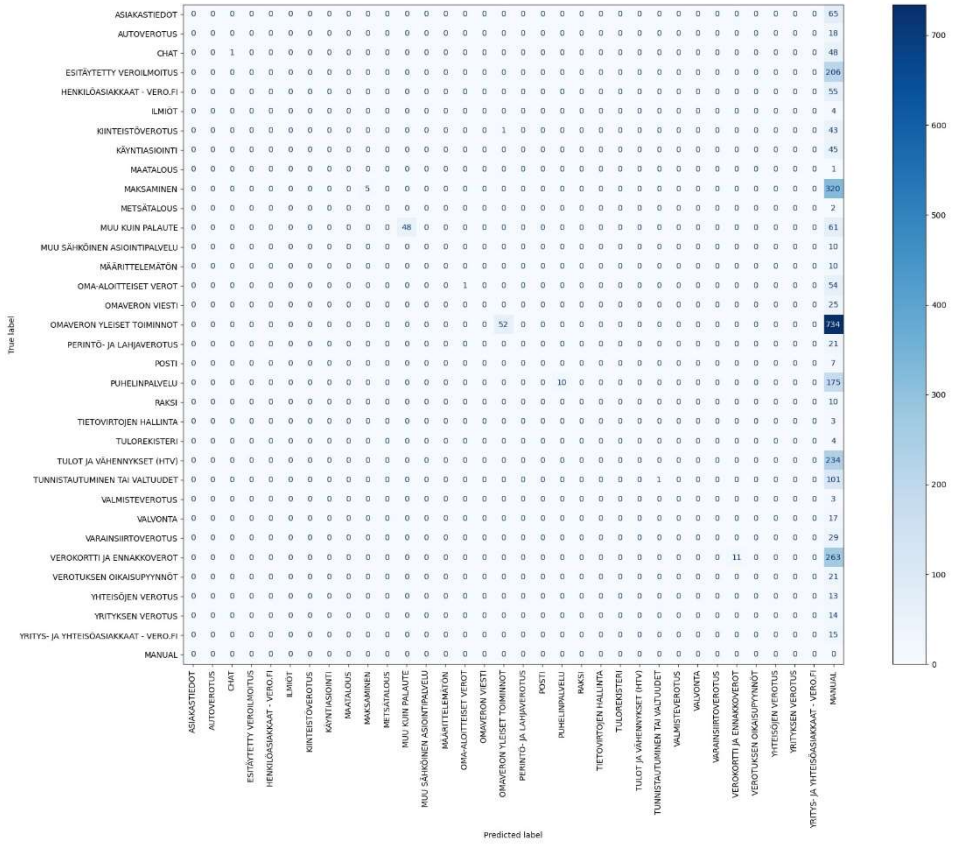
	precision	recall	f1-score	support
ASIAKASTIEDOT	1.00	0.89	0.94	9
AUTOVEROTUS	0.00	0.00	0.00	1
CHAT	1.00	0.76	0.87	17
ESITÄYTETTY VEROILMOITUS	0.88	0.76	0.81	29
HENKILÖASIAKKAAT - VERO.FI	0.83	1.00	0.91	34
KIINTEISTÖVEROTUS	0.00	0.00	0.00	5
KÄYNTIASIOINTI	0.88	0.71	0.79	21
MAKSAMINEN	0.94	0.98	0.96	162
MUU KUIN PALAUTE	0.99	0.99	0.99	91
MUU SÄHKÖINEN ASIOINTIPALVELU	1.00	1.00	1.00	1
MÄÄRITTELEMÄTÖN	0.00	0.00	0.00	1
OMA-ALOITTEISET VEROT	0.90	0.90	0.90	10
OMAVERON VIESTI	1.00	0.67	0.80	3
OMAVERON YLEISET TOIMINNOT	0.93	0.96	0.94	437
PERINTÖ- JA LAHJAVEROTUS	1.00	1.00	1.00	2
POSTI	0.00	0.00	0.00	1
PUHELINPALVELU	0.88	0.98	0.93	97
RAKSI	1.00	1.00	1.00	2
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	1
TULOREKISTERI	0.00	0.00	0.00	1
TULOT JA VÄHENNYKSET (HTV)	0.90	0.91	0.90	67
TUNNISTAUTUMINEN TAI VALTUUDET	0.96	0.74	0.84	31
VARAINSIIRTOVEROTUS	1.00	0.80	0.89	5
VEROKORTTI JA ENNAKKOVEROT	0.94	0.94	0.94	144
VEROTUKSEN OIKAISUPYYNNÖT	1.00	0.50	0.67	2
YRITYKSEN VEROTUS	0.00	0.00	0.00	3
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	5
accuracy			0.93	1182
macro avg	0.67	0.61	0.63	1182
weighted avg	0.91	0.93	0.92	1182

APPENDIX 8 The confusion matrix and classification score for probability threshold of 80%



	precision	recall	f1-score	support
AUTOVEROTUS	0.00	0.00	0.00	1
CHAT	1.00	0.67	0.80	6
ESITÄYTETTY VEROILMOITUS	1.00	1.00	1.00	7
HENKILÖASIAKKAAT - VERO.FI	0.85	1.00	0.92	17
KIINTEISTÖVEROTUS	0.00	0.00	0.00	2
KÄYNTIASIOINTI	0.80	0.80	0.80	5
MAKSAMINEN	0.98	0.99	0.98	94
MUU KUIN PALAUTE	0.99	1.00	0.99	81
OMA-ALOITTEISET VEROT	1.00	1.00	1.00	3
OMAVERON YLEISET TOIMINNOT	0.96	0.97	0.97	279
PUHELINPALVELU	0.94	0.98	0.96	52
TIETOVIRTOJEN HALLINTA	0.00	0.00	0.00	1
TULOT JA VÄHENNYKSET (HTV)	0.93	0.93	0.93	28
TUNNISTAUTUMINEN TAI VALTUUDET	1.00	0.64	0.78	11
VEROKORTTI JA ENNAKKOVEROT	0.95	0.99	0.97	70
VEROTUKSEN OIKAISUPYYNNÖT	0.00	0.00	0.00	1
YRITYKSEN VEROTUS	0.00	0.00	0.00	1
YRITYS- JA YHTEISÖASIAKKAAT - VERO.FI	0.00	0.00	0.00	2
accuracy			0.96	661
macro avg	0.63	0.61	0.62	661
weighted avg	0.95	0.96	0.95	661

APPENDIX 9 The confusion matrix and classification score for probability threshold of 90%



	precision	recall	f1-score	support
CHAT	1.00	1.00	1.00	1
KIINTEISTÖVEROTUS	0.00	0.00	0.00	1
MAKSAMINEN	1.00	1.00	1.00	5
MUU KUIN PALAUTE	1.00	1.00	1.00	48
OMA-ALOITTEISET VEROT	1.00	1.00	1.00	1
OMAVERON YLEISET TOIMINNOT	0.98	1.00	0.99	52
PUHELINPALVELU	1.00	1.00	1.00	10
TUNNISTAUTUMINEN TAI VALTUUDET	1.00	1.00	1.00	1
VEROKORTTI JA ENNAKKOVEROT	1.00	1.00	1.00	11
accuracy			0.99	130
macro avg	0.89	0.89	0.89	130
weighted avg	0.98	0.99	0.99	130