Covid-19: Predicting Survival With Data Mining

Elisabeth Berg Bøgebjerg (elib@itu.dk), Frida Helene Beck-Larsen (frib@itu.dk)

***Abstract*— In this paper we will explore how a Covid-19 dataset provided by the Mexican government can be used to examine what characteristics such as lifestyle, underlying medical problems, and possible medical treatment are decisive for dying while being infected with Covid-19. We will apply the Decision Tree algorithm to predict whether a person dies from Covid-19 and examine if the Principal Component Analysis algorithm can decompose the characteristics to get faster and more precise predictions. We find that the most important features for predicting death are hospitalization, intubation, pneumonia, and age. Furthermore, we find that the most important features from the Decision Tree algorithm can be reduced to one component with Principal Component Analysis without losing performance.**

1. INTRODUCTION

The coronavirus, including Covid-19, is an infectious virus and respiratory illness that can cause everything from mild colds to serious respiratory infections, which can result in death [1]. The disease spreads by transferring the virus from an infected person to another person through droplets and virus particles that are released into the air when an infected person breathes or talks [1]. Covid-19 caused a pandemic and since 2019 there have been more than 642.379.243 globally confirmed cases of Covid-19, and a total number of 6.624.118 deaths have been registered to WHO (dated December 8th, 2022) [2]. The Danish Health Authority states that certain groups are in risk to serious disease course, these groups include e.g., elderly people, obesity, smoking, heart-, lung-, and cancer diseases, certain chronic diseases and pregnancy [3].

We are working with the dataset 'COVID-19 Dataset' [4]. The dataset is provided by the Mexican government and contains data divided into 21 features on 1,048,567 anonymized persons. The features describe sex, age, lifestyle such as obesity and smoking, underlying illnesses such as diabetes, pneumonia, asthma, cardiovascular diseases, or hypertension. The features also describe which medical treatment of Covid-19 has been given, e.g., if the person has been hospitalized, intubated, or admitted to the Intensive Care Unit, and if the person survived or died.

To explore which factors and comorbidities are associated with death while being infected with Covid-19, we will examine the following question:

- What are the most decisive characteristics for predicting whether a person is dying when infected with Covid-19?

To examine the above question, we will apply the supervised learning method Decision Tree algorithm and the unsupervised learning method Principal Component Analysis (PCA). The Decision Tree algorithm is implemented to predict the risk of a person's death based on lifestyle, underlying medical conditions, and medical treatment history. The PCA algorithm is used to reduce the features of the dataset to examine if it makes more precise predictions.

We decided to work with this dataset due to the range of features, that can shed light on which underlying comorbidities and illnesses, lifestyle factors, and medical treatment that increases the risk of death when affected by Covid-19. The dataset includes several lifestyle and medical predictor variables, which can be a valuable tool to governments, health care authorities, and health care professionals in the decision-making process of prioritizing different persons when there is a limited number of medical units or hospital beds.

2. A LOOK INTO THE DATASET

The dataset has 21 features that hold personal data including lifestyle and chronic illnesses. The features medical unit (type of institution) date died, age, USMR (treated with medical unit of first, second, or third level), and classification (covid test findings) describes the information about the person based on ratio-scaled attributes. If a person died the date died feature contains the death date, and if the person survived it is registered as '9999-99-99'.

The rest of the features are defined by binary attributes. They include sex, patient type (hospitalized), pneumonia (air sacs inflammation), pregnancy, diabetes, COPD (chronic obstructive pulmonary disease), asthma, inmsupr (immunosuppressed), hypertension, cardiovascular (heart or blood vessels related disease), renal chronic, other disease, obesity, tobacco, intubated, and ICU (admitted to Intensive Care Unit). For these features 1 means yes and 2 means no. The sex attribute is also defined by 1 and 2, in this case 1 means female and 2 means male. The pregnant feature has the attributes 1 for true and 2 for false for all females and the attributes 97 and 98 for false for all males. For the patient type feature 1 means returned home and 2 means hospitalization. Missing values are registered as 97, 98, or 99. Figure 1 shows a small subset of the data.

	USMER	MEDICAL_UNIT	SEX	PATIENT_TYPE	DATE_DIED	INTUBED	PNEUMONIA	AGE	PREGNANT	DIABETES
0	2	1	1	1	03/05/2020	97	1	65	2	2
1	2	1	2	1	03/06/2020	97	1	72	97	2
2	2	1	2	2	09/06/2020	1	2	55	97	1
3	2	1	1	1	12/06/2020	97	2	53	2	2
4	2	1	2	1	21/06/2020	97	2	68	97	1

Figure 1: DataFrame using the head() function

3. PRE-PROCESSING

This section describes the work that has been conducted before applying the algorithms. The pre-processing includes feature selection, data cleaning, and data transformation to make the data manageable and complete. Finally, a visual presentation of the data we are working with follows.

3.1 Feature selection

We have chosen to work with most of the features from the dataset in order to determine which features are most decisive in case of dying of Covid-19. The included features are related to lifestyle, health problems, and medical treatment during hospitalization. The features USMER, medical unit, and other disease were dropped due to their irrelevance and the lack of details.

Furthermore, we have renamed the columns intubed, hipertension, and date died to the more precise and correct descriptions intubated, hypertension, and died.

3.2 Data Cleaning

To ensure data quality such as accuracy, completeness, and consistency we have checked for incomplete and noisy data [5]. To begin with the dataset has been cleaned as part of the pre-processing, which included handling missing values and outliers.

All values for males in the pregnant column are registered as 97. Values in the columns intubated and ICU were registered as 97 and 99 if the patient type was 2 (when the person was not hospitalized). We replaced the values in the columns pregnant, intubated, and ICU with 2 for 'No'. All other data points with missing data (registered as 97, 98 or 99) have been dropped for this project. If the dataset had fewer data points it could have been necessary to replace missing data values with values based on the values from other similar data points.

We applied a histogram to the age feature as a basic statistical description technique to detect if there were any outliers in the dataset. The initial result showed that the ages of the people are between 0 – 120. In conclusion, the dataset is relatively clean, and we have not found any clear outliers or noisy data objects that could result in imprecise results.

The dataset also holds information about persons who tested negative for Covid-19 or had an inconclusive test result. In order to work with data that only contains persons who contracted Covid-19 we have filtered it by the classification feature where values 1-3 states that the persons were diagnosed with Covid-19 in different degrees.

3.3 Data Transformation

In order to work with the died feature, the column has been transformed to binary features. If the person survived after being infected with Covid-19, is it registered as 2, and if a person died is it registered as 1.

The final part of the pre-processing included creating bins for the age feature. The binning method smooths a data value by looking at the values around it and distributes all data values into a number of bins [5]. We have created age bins with `KBinsDiscretizer` and encoded them '*ordinal'* instead of '*onehot'* [6]. Using '*onehot'* encoding with Decision Trees can cause problems since the algorithm will create a sparser Decision Tree and look at each dummy feature as independent [7].

After the pre-processing the dataset holds 18 columns and 388.071 data points. Due to the large dataset, we work with a sample of the dataset that holds 2000 data points.

3.4 Data Visualization

Below follows chosen data visualizations of the data sample after cleaning the data and transforming the died column, but before we used `KbinsDiscretizer`.

The mean age of the persons is 45 and the distribution of the ages is shown in the histogram in Figure 2. There is a close to equal distribution of females and men (Figure 3). Most of the persons in the sample survived Covid-19 infection (Figure 4) and almost ¾ of the persons avoided hospitalization (Figure 5). Lastly, almost 6% of the people were intubated (Figure 6).

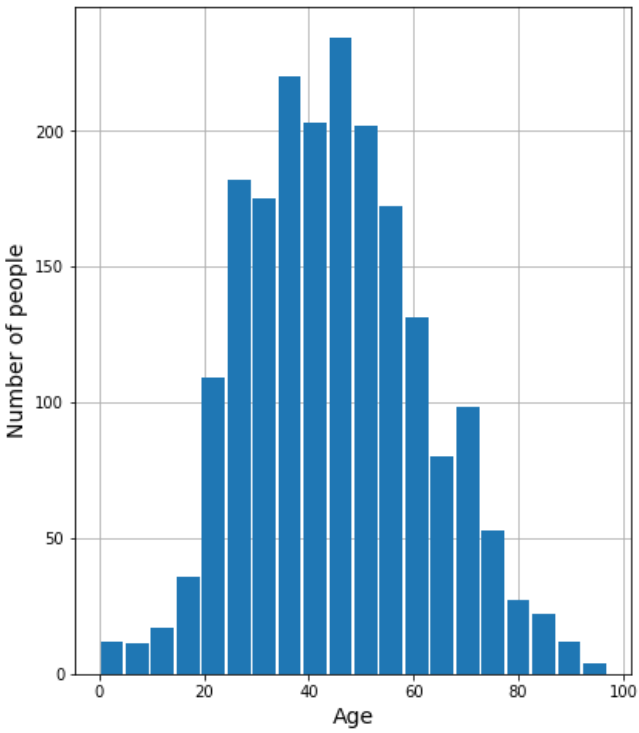

Figure 2: Histogram of age distribution with 20 bins

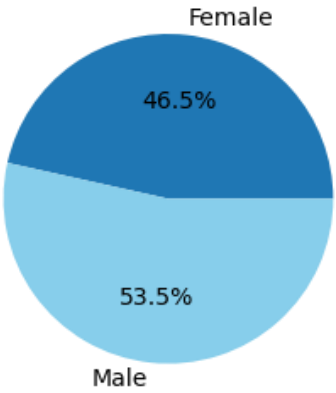

Figure 3: Pie chart showing the proportion of men and women

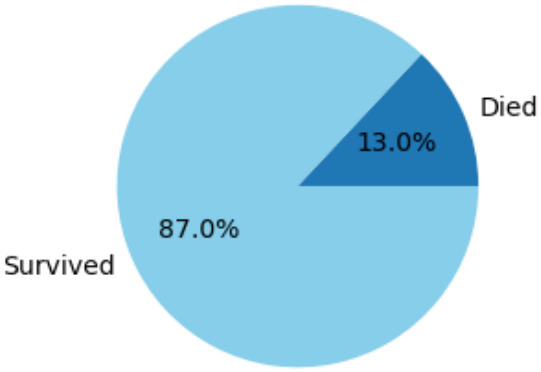

Figure 4: Pie chart showing the proportion of died and survived persons

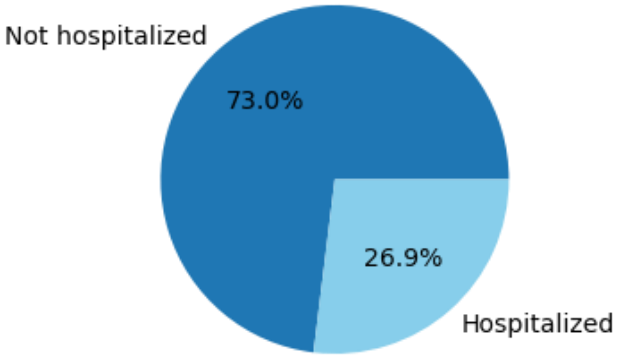

Figure 5: Pie chart showing the proportions of hospitalization

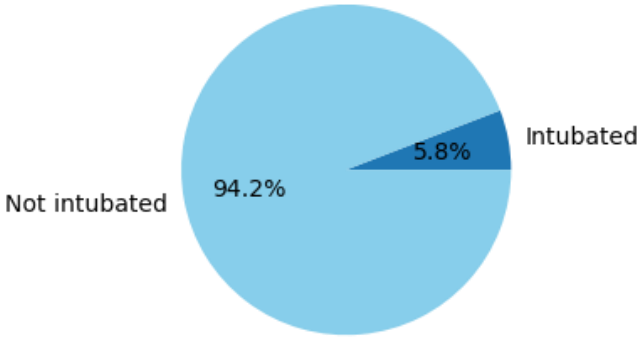

Figure 6: Pie chart showing the proportions of intubation

4. METHODS

In this section follows a description of the chosen methods we will use to answer the research question. We will use implementations of the Decision Tree algorithm and Principal Component Analysis (PCA) from Scikit Learn.

We created two models; one with the Decision Tree algorithm, and one with both PCA and the Decision Tree algorithm. The aim of this approach is to use the results from the first model for feature selection in the second model and to examine if the same accuracy can be maintained or improved of the Decision Tree by reducing the dimensionality of the data. Firstly, the Decision Tree algorithm is run on the data, and the accuracy is calculated. Afterward, PCA is applied before the Decision Tree algorithm.

To create the two models, we made use of `Pipeline` and `GridSearchCV` from Scikit Learn.

With the Pipeline library, we have organized some of the data transformation to automate the steps of the analysis and ensured that the models can be reused across datasets [8].

The GridSearch library was utilized to find the most optimal hyperparameters. `GridSearchCV` is used as a cross-validation tool by calculating the score for each combination of the predefined parameters, and hereby determine the most optimal values for our model. The algorithm goes through predefined hyperparameters and fits it to our model and returns the best parameters [9].

We evaluate the models by splitting the data into a training and test set, where the training set will be 2/3 of the data. We set random state to 0, such that the results can be reproduced. Splitting the dataset is also done to avoid overfitting. Overfitting can occur when the model fits exactly to the data, which can mean that the model is unable to perform on unseen data [10].

To extract the performance of the models we make use of the `accuracy_score` to measure percentage of test tuples that are correctly classified [10]. Due to imbalanced data in the data sample (the distribution of persons who died and survived), we use the `F1 score` to take the number of prediction errors and type of errors into consideration [11].

The F1 score describes the harmonic mean of precision and recall, which provides an alternative way to use both measures [10]

4.1 Decision Tree

The Decision Tree algorithm has been implemented from Scikit Learn, where we have worked with the library `DecisionTreeClassifier`[12], which is a version of the `CART` algorithm [13].

The Decision Tree algorithm is applied to determine whether a person is most likely to survive or die after being infected with Covid-19 based on sex, age, underlying medical conditions, and hospital treatment. The algorithm creates a tree structure, where internal nodes represent the dataset's features, each branch represents the decision rules, and the leaf nodes represent the result [10]. The nodes are split based on a split criterion, like the entropy measure and the goal is to reduce impurity of the nodes [10]. The split criterion indicates how pure the split is by looking at how many tuples have been correctly placed. A value of 0 indicates that the accuracy of the tuple that has been assigned to a class label is a pure prediction.

The process of implementing the `DecisionTreeClassifier` consists of fitting it to a training set, predicting results on a test set, and calculating the accuracy of the result. The training set is created from the dataset and contains tuples and the class label, which is represented as discrete values. When the algorithm is given a tuple that belongs to a class label, it is tested in the Decision Tree that has been created by the training set. The leaf node of the Decision Tree contains the prediction of the class label, and the tuple runs through the tree from the root node to a leaf node to make a prediction. In the visual representation of the Decision Tree each branch that goes left from each internal node means *true*, and each branch that goes right from each internal node means *false*. The orange nodes indicates that the person who contracted Covid-19 died, and the blue nodes indicates that the person survived. The shade of each node (the lightness/darkness) shows the purity of the node, meaning that the darker nodes are purer. Furthermore, each internal node contains the target value that is tested, the entropy value, the number of samples, and a value that describes the distribution of *true/false* cases.

4.1.1 Model 1: Pipeline with Decision Tree

Firstly, the dataset has been split into a training set and a test set, where the feature data (y), which the model will be built on, is the died column, and the target values (X) are the rest of the chosen columns.

We have set the grid search parameters to search the different binning strategies (uniform, quantile, or k-means), and the number of bins for the `KBinsDiscretizer`.

The Decision Tree algorithm from Scikit Learn has several hyperparameters, from which we have chosen to search the parameter for the split criterion, the minimum number of samples for a node to be split, and the minimum number of samples required for a node to be a leaf node.

The grid search showed the optimal number of age bins from the search space is 4, and they should be split with the uniform strategy, which means that each bin has an equal size [14]. The bins consist of the age groups 0-24, 24-48, 48-73, and 73-97. For the `DecisionTreeClassifier`, the split criterion should be based on the entropy measure. The minimum number of samples for a node to be split is 2, and the minimum number of samples required for a node to be a leaf node is 12.

The Decision Tree that has been created with these specifications is shown in the Results section.

4.2 Principal Component Analysis

For the Principal Component Analysis, we will use the libraries `PCA` and `StandardScaler` from Scikit Learn [15].

Principal Component Analysis (PCA) is implemented as a method of dimensionality reduction. The PCA compresses and combines the correlated data features into a set of linearly uncorrelated features and examines if the data can be reduced without losing valuable information [16]. Additionally, PCA helps to take advantage of a faster running time and fewer features makes it easier to visualize the data.

The procedure of applying PCA to the dataset consists of normalizing the data to ensure that all the data attributes are on the same scale, such that some attributes are not given more weight than others [5]. Then *k* orthonormal principal components are computed, and these components are sorted in decreasing order based on significance [5].

To find the optimal number of components, the variance must be maximized. The Explained Variance is a statistical measure of how much variation can be attributed to each component. By the Explained Variance the components are individually ranked by their importance and be used to interpret the most important components. The component's importance is defined by how large the explained variance is.

One way to maximize variance is to set a threshold for cumulative explained variance where thresholds between 80% and 90-95% have been suggested [17, 18, 19].

4.2.1 Model 2: Pipeline with Principal Component Analysis and Decision Tree

Firstly, the data is divided into a training and test set. The hyperparameters are the same as we found best when running grid search in Model 1.

After binning the age feature, the data is normalized to make sure all features are on the same scale to prepare it for PCA.

The most optimal number of components for PCA are then found with `GridSearchCV` method. To find the best number of features, we ran several experiments with different

numbers of features based on the features importance's obtained from Model 1.

We got the best performance from the features patient type, intubated, pneumonia, and age. The best hyperparameters is 1 principal component, and the cumulative explained variance is 54%.

The Decision Tree that has been created with these specifications is shown in the Results section.

5. RESULTS

In the following section, the performance of each model is evaluated by interpreting their accuracy and F1 scores. Furthermore, we will present and interpret the Decision Tree's for Model 1 and Model 2 and examine what features is most important.

5.1 Performance of Model 1

Model 1 was run with the best parameters found from the grid search, which resulted in an accuracy of 0.89 on the test set. This score shows that in 89% of all predictions the model was able to make the correct prediction. Additionally, the F1-score for persons who died is 0.57, whereas the score is 0.94 for survived persons. Thus, Model 1 performs better when predicting survival than death.

5.1.1 Results of Model 1

The Decision Tree from Model 1 is shown in Figure 7. The full Decision Tree can be found in the code. bs

The figure of the Decision Tree in Model 1 shows that if a person who contracted Covid-19 has been hospitalized (patient type >= 1.5), connected to a ventilator (intubated <= 1.5) and is a female the person is at risk of dying from an Covid-19 infection. Persons who have not been hospitalized is most likely to survive an Covid-19 infection no matter of other factors like pneumonia, diabetes, age, and sex.

The left side from the first node of the Decision Tree (persons who had not been hospitalized) shows that the feature 'pneumonia' and its children to the right have low entropy scores meaning that these features have a higher level of purity, and shows the likelihood is close to 100%. Generally, the left side of the tree shows that a person has better chances of surviving if the person has not been hospitalized regardless of other factors.

The right side of the Decision Tree (persons who have been hospitalized) shows that if a person has been connected to a ventilator (intubated), is a female, suffers from hypertension, and is admitted to intensive care unit they are in greater risk of dying from an Covid-19 infection. It is worth noticing the higher entropy value in the 'intubated' class, which has a value of 0.96. This testifies a lower level of purity, meaning that the prediction is less clean than classes with lower entropy values. This also means that if a person has not been connected to a ventilator the model predicts that a person is most likely to survive.

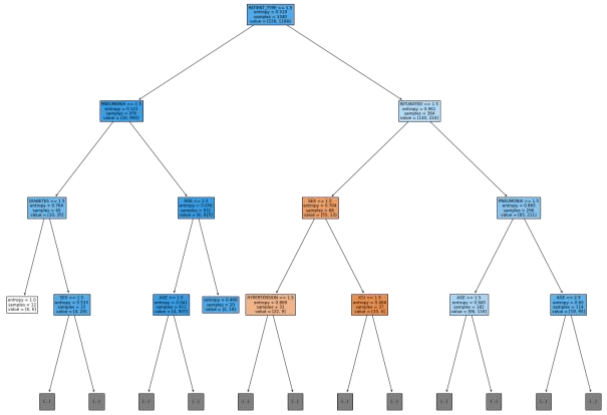

Figure 7: Decision Tree from Model 1 with a depth of 3

The feature importance overview shows that being hospitalized or not is the most important feature when predicting survival, which has a remarkable margin to the second most important feature (Figure 8). The features intubated, pneumonia, and age are the next most decisive features. The factors that seemed less crucial are the gender, if a person has diabetes, hypertension, or is obese. The factors which proved not to be crucial in predicting whether a person dies of Covid-19 are being admitted to intensive care unit, being pregnant, suffering from chronic obstructive pulmonary diseases, asthma diagnosis, immunosuppression, heart or blood vessels related diseases, chronic renal diseases, and if the person is a tobacco user.

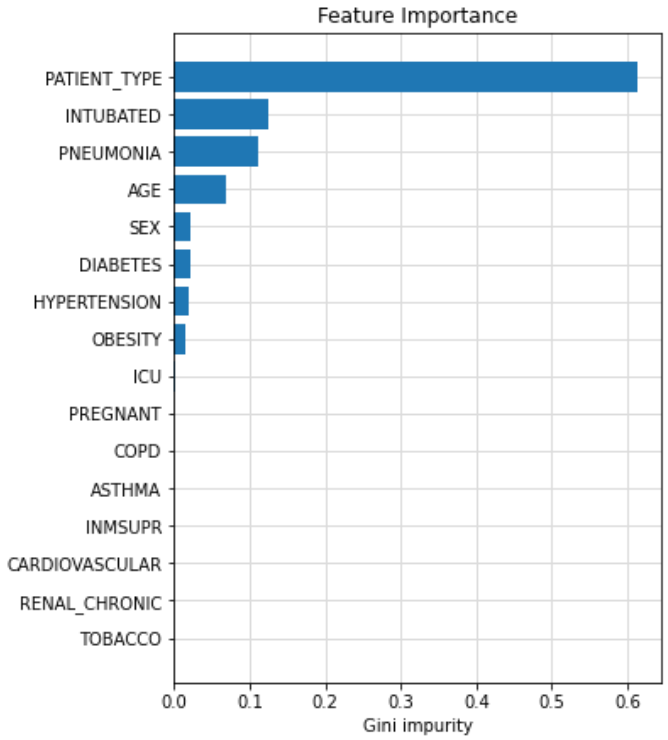

Figure 8: Overview of feature importance

5.2 Performance of Model 2

We found the best performance by reducing the features patient type, intubated, pneumonia, and age to one principal component. It results in an accuracy in the predictions of the Decision Tree of 0.89, which is from the same accuracy as Model 1, but on fewer features.

Like Model 1, this model is also better at predicting survival than death. The F1 score for persons who died is 0.57, and 0.94 for persons who survived, which is the same as in Model 1. Thus, the features patient type, intubation, pneumonia, and age can predict survival as well as all features and with dimensionality reduction from 4 features to 1 principal component. The explained variance of the one principal component is 54%, but since the accuracy and F1 scores are the same as in Model 1, Model 2 still has enough variance from the features to make equally correct predictions.

5.2.1 Results of Model 2

The Decision Tree built by Model 2 is shown in figure 9. All splits are made based on PC1, which shows the challenge using dimensionality reduction, as we lose information, which makes interpretation more difficult. However, since we were able to make predictions with the same accuracy as in Model 1, this approach can be used to run future predictions faster, since the tree in Model 2 too has a depth of 6, whereas the tree in Model 1 has a depth of 8.

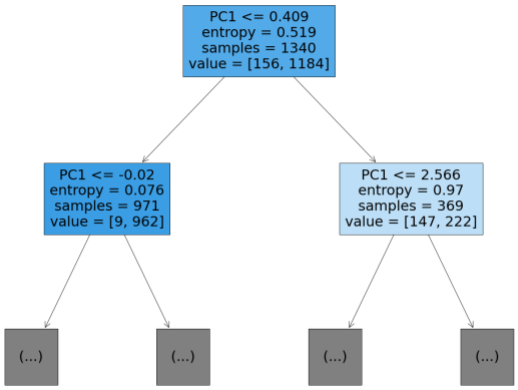

Figure 9: Decision tree from model 2 with depth 1.

6. Conclusion

In conclusion, the two models were best at predicting whether persons survived. Both models were equally accurate, which suggests that future predictions can be conducted with improved performance on fewer features with reduced dimensionality without losing accuracy.

The first model with the Decision Tree Classifier showed that the most decisive features for predicting if a person infected with Covid-19 will survive or die are patient type (hospitalization) and intubation. The less decisive features include if the persons suffer from pneumonia, diabetes, and/or hypertension and as well their age, sex, and if they are obese. Previous studies have proven that the majority of Covid-19 patients who have been intubated died and state that older age increases the risk of dying with Covid-19 when intubated [20, 21]. Thus, our results support these findings.

Model 2 with PCA and Decision Tree showed that the 4 most important features from Model 1 could be decomposed to one principal component without losing accuracy.

7. Discussion

In this section we will discuss our approach and the obtained results.

The analysis is conducted on a sample of 2000 persons, where 260 persons died, and 1740 survived. The imbalance in the data could possibly explain why the F1 score was significantly higher for prediction of survival. Imbalanced data is frequent in real-world applications, and the entropy value is sensitive to skewed class distribution [22]. A more balanced sample could perhaps improve the model. Another option could have been to add weights to the class labels to address the imbalance [12].

By applying the Decision Tree algorithm to the dataset, we saw that the most decisive factor was if a person had been hospitalized. It also showed that if a person is hospitalized, have been connected to a ventilator, and is a male the next the node the tree will test against is whether the person has been admitted to the intensive care unit. This made us reflect on the relationship between the different features. According to the weekly reports of the Danish Health Authority (Statens Serum Institut) a person is admitted to the intensive care unit if they are connected to a ventilator [23]. If the treatment practice is the same in Denmark and in Mexico (where the data is derived from) it may mean that the patient is both hospitalized and admitted to the intensive care unit when connected to a ventilator, even when the results did not show a preponderance of these factors. This perspective raised the question about whether a researcher is required to possess domain knowledge to know the context and which features are significant [24]. This domain knowledge could also have been valuable during the pre-processing to find outliers, by examining different combinations of features.

The results showed that being hospitalized while being affected by Covid-19 is the most decisive factor when predicting whether a person is at risk of dying. The results also showed that as soon a person is connected to a ventilator, the risk of dying is higher. From the results and the fact that the dataset consists of a great range of statistical facts, lifestyle diseases, medical history, and possible course of treatment for each person who contracted Covid-19 we have reflected on how the features of the dataset could have been used to examine which factors increase the risk of being hospitalized, admitted to intensive care unit, and connected to a ventilator. By investigating which underlying diseases, both lifestyle, chronic and general diseases the persons may suffer from in

relation to factors such as gender and age, we could analyze which combinations of characteristics and thus which patient groups are at increased risk for a complicated Covid-19 course. The results from this research could have been used as a tool for health care professionals to help them prioritize which patients should be prioritized in cases of a limited capacity at the hospitals. Additionally, the knowledge could be used to convey this risk to the patient groups, so the persons could take precautions to avoid contracting Covid-19.

A perspective that was raised during the implementation of the Principal Component Analysis (PCA) was the fact that most of the features in the dataset are binary attributes. PCA is programmed to find the lowest squared deviations, which is equal to the most minimized variance. The squared deviation is the squared difference between two values, and the sum of squared deviations for a feature is its mean, which is the variance. When finding the mean of the features of the dataset that mainly contains binary attributes we get a scale of only two values, which affects the variance. Based on this variance, PCA creates a centroid point, which can explain why we get the most optimal number of components to be 1. You could argue that finding the mean of binary attributes is inappropriate due to the scale of the values.

References

[1] Coronasmitte.dk. "Hvad er COVID-19, og hvordan opstod sygdommen?". Accessed Dec. 8, 2022. https://coronasmitte.dk/viden-om-covid-19/artikler/2021/marts/hvad-er-covid-19-og-hvordan-opstod-sygdommen

[2] WHO. WHO Coronavirus (COVID-19) Dashboard. Accessed Dec. 8, 2022. https://covid19.who.int/

[3] Sundhedsstyrelsen. "Personer med øget risiko ved COVID-19 – fagligt grundlag". Marts 21, 2021. https://www.sst.dk/da/Udgivelser/2020/Personer-med-oeget-risiko-ved-COVID-19

[4] Meir Nizri. COVID-19 Dataset. https://www.kaggle.com/datasets/meirnizri/covid19-dataset?resource=download

[5] Han, J., Kamber, M., & Pei, J. "Data Preprocessing" in *Data Mining – Concepts and Techniques*, 3rd edition. Morgan Kaufmann, 2011, chapter 3, pages 83 – 120.

[6] Scikit Learn. sklearn.preprocessing.KBinsDiscretizer. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html

[7] Ravi, R. One-Hot Encoding is making your Tree-Based Ensembles worse, here's why. Towards Data Science, 2019. Accessed Jan. 2, 2023. https://towardsdatascience.com/one-hot-encoding-is-making-your-tree-based-ensembles-worse-heres-why-d64b282b5769

[8] Scikit Learn. sklearn.pipeline.Pipeline. https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

[9] Scikit Learn. sklearn.model_selection.GridSearchCV. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[10] Han, J., Kamber, M., & Pei, J. "Classification: Basic Concepts" in *Data Mining – Concepts and Techniques*, 3rd edition. Morgan Kaufmann, 2011, chapter 8, pages 330 – 392.

[11] Korstanje, J. The F1 score. Towards Data Science, 2021. Accessed Jan. 6, 2023. https://towardsdatascience.com/the-f1-score-bec2bbc38aa6

[12] Scikit Learn. sklearn.tree.DecisionTreeClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[13] Scikit Learn. 1.10. 6. Tree algorithms: ID3, C4.5, C5.0 and CART. Accessed Jan. 8, 2023. https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

[14] Scikit Learn. Demonstrating the different strategies of KBBinsDiscretizer. Accessed Jan. 5, 2023. https://scikit-learn.org/stable/auto_examples/preprocessing/plot_discretization_strategies.html

[15] Scikit Learn. sklearn.decomposition.PCA. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[16] Han, J., Kamber, M., & Pei, J. "Advanced Cluster Analysis" in *Data Mining – Concepts and Techniques*, 3rd edition. Morgan Kaufmann, 2011, chapter 11, pages 497 – 542.

[17] Géron. A. "The Machine Learning Landscape" in *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd edition. O'Reilly Media, 2019.

[18] Holland, S. M. Principal Components Analysis. Department of Geology, University of Georgia, Athens. 2008.

[19] Lindgren, I. Dealing with Highly Dimensional Data using Principal Component Analysis (PCA). Towards Data Science, 2020. Accessed Jan. 4, 2023. https://towardsdatascience.com/dealing-with-highly-dimensional-data-using-principal-component-analysis-pca-fea1ca817fe6

[20] Luo M, Cao S, Wei L, Zhao X, Gao F, Li S, Meng L, Wang Y. (2020) Intubation, mortality, and risk factors in critically ill Covid-19 patients: A pilot study. J Clin Anesth.
[21] Salehi, M., Mohammadi, M., Abtahi, S. et al. (2022) Risk factors of death in mechanically ventilated COVID-19 patients: a retrospective multi- center study. Available at Research Square [https://doi.org/10.21203/rs.3.rs-1362678/v1]

[22] Chaabane, I., Guermazi, R., & Hammami, M. (2020). Enhancing techniques for learning decision trees from imbalanced data. Advances in Data Analysis and Classification, volume 14, pages 677-745.

[23] Statens Serum Institut. Ugentlige opgørelser med overvågningsdata. Covid-19 Statens Serum Institut. Accessed Jan. 8, 2023. https://covid19.ssi.dk/overvagningsdata/ugentlige-opgorelser-med-overvaagningsdata

[24] Han, J., Kamber, M., & Pei, J. "Cluster Analysis: Basic Concepts and Methods" in *Data Mining – Concepts and Techniques*, 3rd edition. Morgan Kaufmann, 2011, chapter 11, pages 443 – 496.

Appendix

Statement of contribution

This appendix provides an overview of how we assigned our written work. As a small note we want to clarify that we both equally contributed to all sections of the whole document.

Abstract - Elisabeth & Frida
1. Introduction – Elisabeth & Frida
2. A look into the dataset – Elisabeth & Frida
3. Pre-processing - Frida
4. Methods – Elisabeth & Frida
    4.1 Decision Tree - Elisabeth
    4.2 Principal Component Analysis – Frida
5. Results
    5.1 Performance & Results of Model 1 – Elisabeth
    5.2 Performance & Results of Model 2 - Frida
6. Conclusion – Elisabeth & Frida
7. Discussion – Elisabeth

Link to CoLab

Link to our code and dataset on CoLab.

https://colab.research.google.com/drive/1n0UdQ-5pbg8-nUuQ5bMRWH6oS23wpiTp?usp=sharing