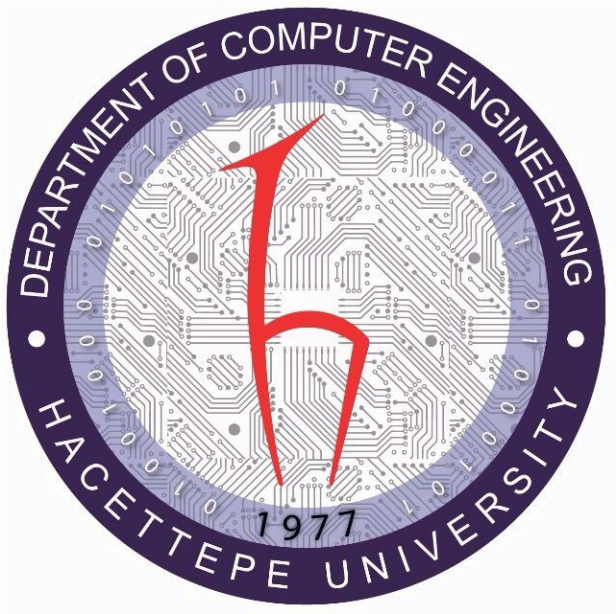
Hacettepe University
Computer Engineering Department



BBM 467 Data Intensive Applications
2021 Spring Semester

SDSP Project Report

Dilara İŞERİ - 21783561
Umut AYDEMİR - 21827148

Introduction

The first and most important step to treat a disease is to diagnose the disease correctly. Because the treatment of the misdiagnosed disease will also be wrong and the patient will not be able to recover. However, some diseases can be difficult to diagnose by doctors. This situation can have many reasons. The disease may be very rare in humans. And for this reason, doctors may not have much experience with this disease. Apart from this, the tests required to diagnose some diseases can be tiring and costly for the patient. This is an important factor that makes the diagnosis of the disease difficult.

In addition to these, the symptoms shown by the patient may also belong to different diseases. This situation causes doctors to sometimes misdiagnose. This situation is an important problem that needs to be solved. The aim of this project and the main problem it wants to solve is to diagnose diseases that are difficult to diagnose and require expensive examinations, by using resources in the best way and without compromising the quality of health services.

Our aim in this project is to provide patients with an application that predicts which diseases these patients have. In making this estimation, we aimed to use the symptoms and characteristics of the patients. Thus, we tried to ensure that diseases are diagnosed quickly, easily, cheaply and effectively.

Our project within the scope of this purpose basically consists of two parts. The first of these is the design and creation of a machine learning model that predicts the diseases of new patients using existing patient records. The second part of the project is the design of a web application that takes the patient's information as input and predicts the patient's disease using the machine learning model.

Task 1 Preparing a Machine Learning Model

The first of these is the design and creation of a machine learning model that predicts the diseases of new patients using existing patient records. In this step of the project, we used python programming language for data analysis and machine learning modeling. We implemented this step on the jupyter notebook. We have prepared a much more detailed notebook by running the codes we created on Jupyter notebook piece by piece. The data analysis we have done and the machine learning model we have developed are explained in detail on the jupyter notebook with their codes. This notebook has been delivered with the project. However, it will be useful to briefly mention the steps we have implemented in this report.

In this project, we used a ready-made dataset to develop the machine learning model. However, we analyzed and prepared the dataset to be able to use it in machine learning modeling. In a data analysis and machine learning project, the most important and most time consuming part is the analysis and preparation of the data. For this reason, we gave extra importance to understanding, analyzing and preparing the dataset.

Let's talk about the work we have done in Task 1 and briefly the summary of the jupyter notebook.

Data Understanding

In this section, we aimed to understand the dataset provided as ready and to determine the current features of our dataset. We used the pandas library to work on the dataset. In this section, after reading the dataset, we learned the shape of our data set using the pandas library. Our dataset consists of 400 rows and 51 columns. Then we learned which functions our dataset consists of and the data types of these functions. In this way, we have learned about our dataset. In addition, we determined what we should do in the data preparation section. In addition, in this section, we have also detected features containing missing values. Apart from that, we also dropped the ineffective features containing the same categorical value for all rows.

Data Analysis

The names of the features in the dataset are not specified. That's why in this section, we have specifically examined what is obvious. While examining these properties, we drew a bar plot or pie plot using the matplotlib library.

Later, we applied exploratory data analysis in this section. In this context, we tried to determine the relationships between the features in the dataset. For this, we used features that we know what they are.

In the data preparation section, we prepared our data for machine learning modeling. For this, we first filled the missing values in the dataset. For the categorical features, we replaced the missing values with the most repeating values in the column they belong to. Then we enumerated the categorical values and converted them to an integer data type.

Its dataset consists of 51 columns. Since we will receive this information from the user in the application part, this number is too high. We need to reduce the features in the dataset to a number of features that users can fill without getting bored. That's why we need to do feature selection. Accordingly, we first calculated the correlation between the features and dropped one of the features with high correlation between them.

Then we applied normalization to numeric features. We used MinMaxScaler for this. Thus, the numeric values were in the same range.

Modeling

We used a neural network while creating a machine learning model. While creating the model here, we first created a model with its original dataset. Then, by applying feature selection, we created a model with a dataset with a reduced number of features. We used the sklearn.feature_selection library while doing the feature selection process. Thus, we found the feature's importance points. Later, we collected these points by proportioning them to a hundred. We chose the features until we got 95 scores out of 100. Thus, we reduced our feature number to 19.

We chose the neural network as the machine learning model. Because one of the reasons for this was that we were asked about the probability of possible diseases as a model output. As a result of our research, we learned that we can produce this output with activation functions in the neural network model. We worked on a neural network for this. We used Keras to model with the Neural network. Keras is a simple tool for constructing a neural network. It is a high-level framework based on tensorflow, theano or cntk backends.The details of the neural network model we have created are as follows:

- The model consists of a total of 4 layers, including an input layer, 2 hidden layers and an output layer.
- We used the relu function as the activation function on the input layer and hidden layers. We used softmax as the activation function on the output layer. Because softmax calculates probability values of possible target layers as output. This is our aim.
- We used 128 neurons on the input layer, 64 neurons on the first hidden layer, 32 neurons on the second hidden layer and 4 neurons on the output layer.

First of all, we trained our model with train data with whole features. While training the model, we set the number of epochs to be 120. Then we calculated accuracy with test data. When we created and trained models without feature selection, we achieved 0.87 accuracy.

Then we created the same neural network model with the same activation functions. And we train the model with the selected data set and calculate accuracy with the test set. After eliminating unnecessary features, the accuracy of the created model was 90%. Eliminating unnecessary features had a positive effect on the model and enabled us to create a more accurate model.

Then we tried to set up different neural network models using the selected dataset.

- When we used sigmoid as the activation function, we achieved an accuracy of 87%. It seems that the relu activation function is more suitable for our dataset.
- We created a neural network model with 3 hidden layers together with the Relu activation function. The accuracy of this model is higher than the model with 2 hidden layers. The accuracy of this model came in over 90%.

As a result, the model we use in our application consists of 1 input layer, 2 hidden layers and 1 output layer. Relu is used as an activation function in the input layer and hidden layers. Since we want to obtain a probability output on the output layer, softmax is used as the activation function. 128 neurons are used on the input layer, 64 neurons on the first hidden layer, 32 neurons on the second hidden layer and 4 neurons on the output layer.

Task 2 Developing a Web Application

The first step for the development of the application was to select a framework that fits into our goal. We decided to use Streamlit for this project’s interface since it is used for data science purposes mostly.

Installing libraries required to run

To run the program, users have to install streamlit, tensorflow and keras by using

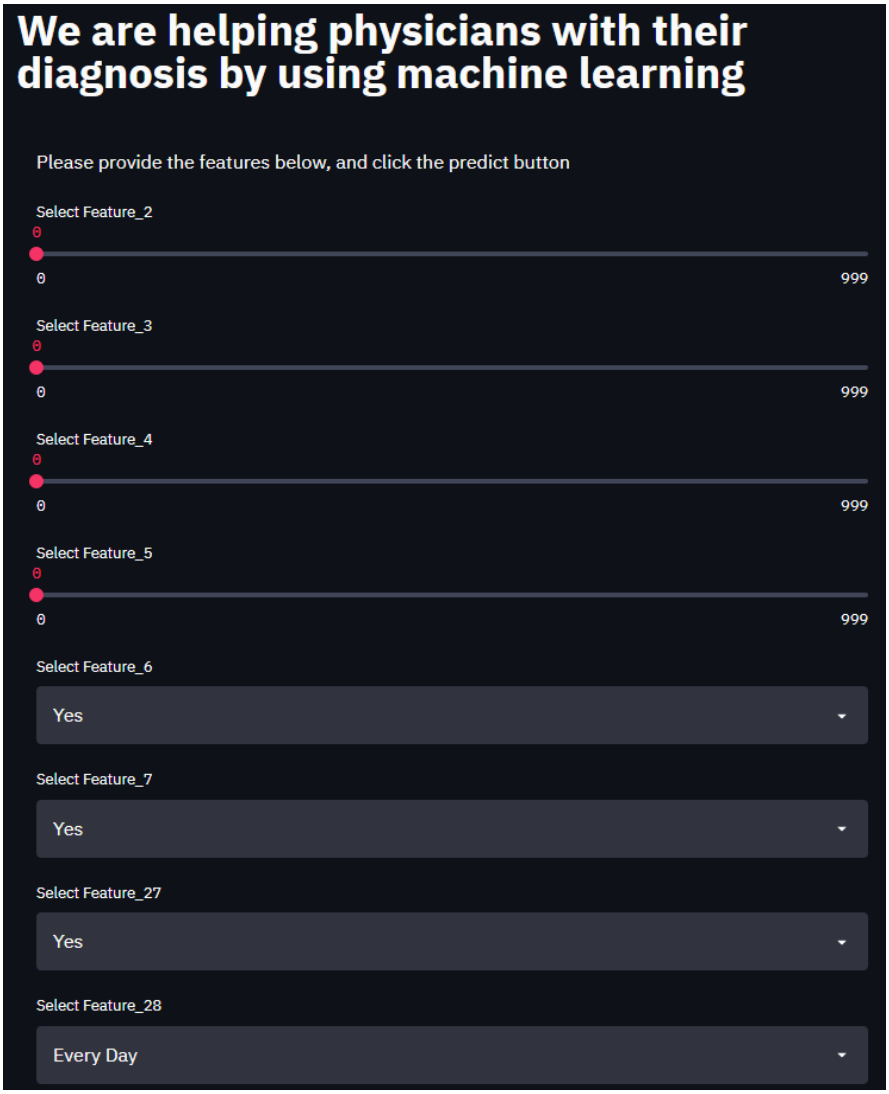
- pip install tensorflow (version = 2.5.0-rc3)
- pip install keras (version = 2.5.0)
- pip install streamlit

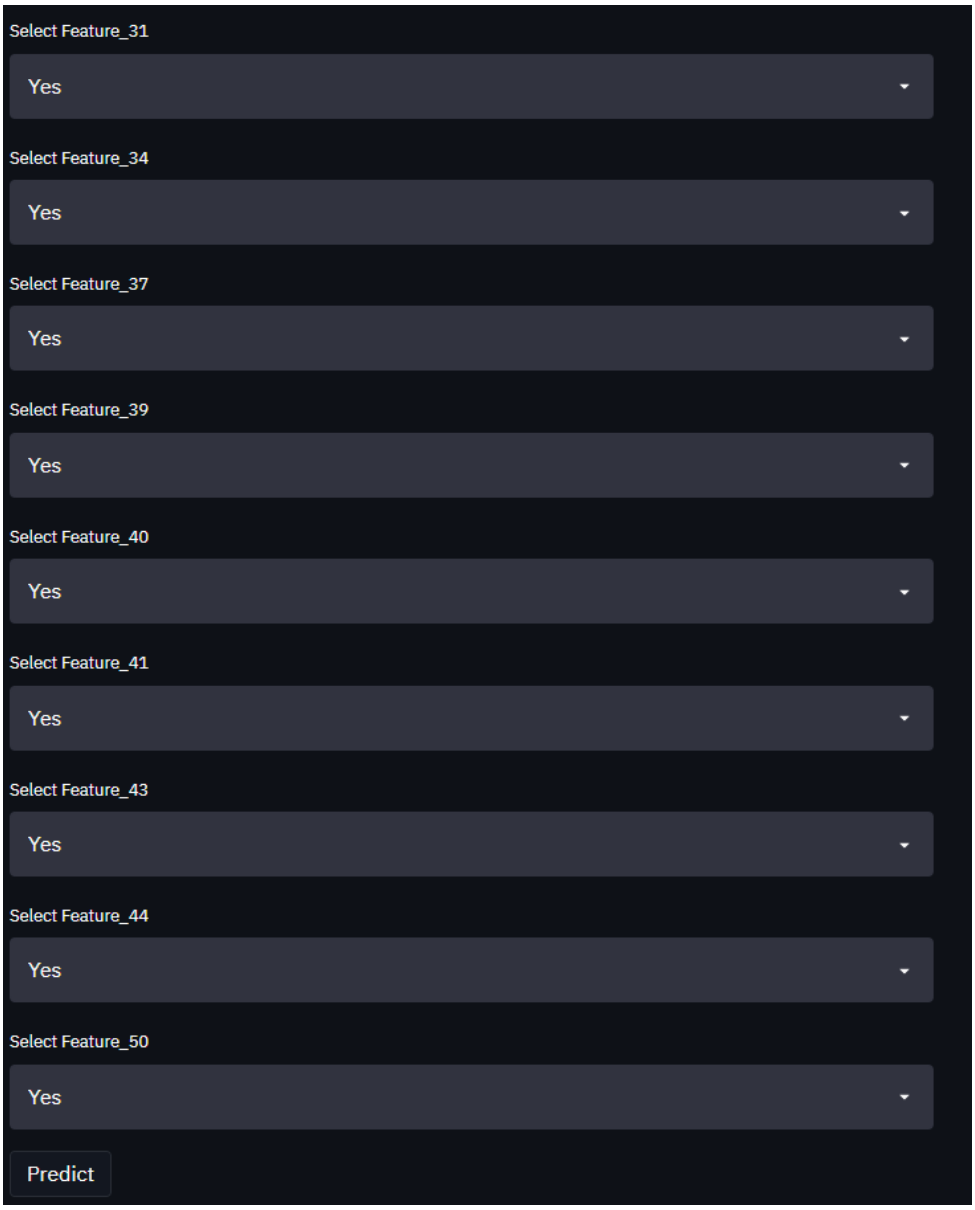
After installing these libraries, to run the program the user have to type

-streamlit run main.py

Creating Form

After implementing our model and data preparation which is shown above to the code, we used select boxes and sliders in order to take data from the user. We used sliders for Feature 2, 3, 4 and 5, and used select boxes for the rest of the form.While implementing the form, we took the features required dynamically from the features selected with data preparation. So that if a feature gets removed, form changes too. After removing the features that are not required, form is created. Here is some screenshots
from the form:



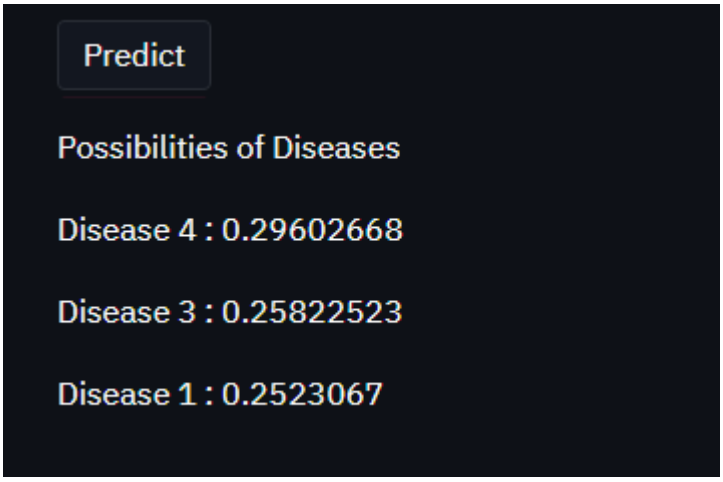


Using the data for predicting with the model

We kept the information which we received from the user within a dictionary. And this dictionary dynamically changes when the user changes a feature and this keeps going until the user clicks the predict button. If a user wants to predict, we cast this dictionary into a dataframe and predict with it. After clicking the predict button, if the user changes a feature the prediction resets.

Prediction

When the user clicks the predict button, we send the data frame through the model and list the top 3 possible diseases in order to show their possibilities like below.



Conclusion

In this data science project, we learned how we can handle daily life problems using machine learning models. And we put these solutions into web applications to make people use this easier and provide abstraction.

References

[1] https://discuss.streamlit.io
[2]https://stackoverflow.com
[3]https://docs.streamlit.io/en/stable/api.html
[4] https://learn.datacamp.com/courses/introduction-to-deep-learning-with-keras
[5] https://learn.datacamp.com/courses/introduction-to-deep-learning-in-python
[6] https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/
[7] https://towardsdatascience.com/building-our-first-neural-network-in-keras-bdc8abbc17f5
[8] https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax
[9] https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d
[10] https://machinelearningmastery.com/softmax-activation-function-with-python/