Project: Cancer Detection with Neural Networks

Author: Miro Zilaji, 10.4.2025

Creating a CAT or PIKE

Can … // algorithm x // calculate / solve the exact … // (customer) problem?

Neural Networks can be used as a tool to predict cancer development and/or possibility of recovery.

Big Data / Data Mining / Data Science Key message:

(Problem):

This study uses Neural Networks as a tool to predict cancer development and/or possibility of recovery. This will be investigated using the "UAE Cancer Patient Dataset." This dataset is designed for research, analysis, and machine learning applications in healthcare. It includes 10,000+ records of synthetic cancer patient data from the United Arab Emirates (UAE) with 20 features, such as:

✓ Patient demographics (Age, Gender, Nationality, Ethnicity)

✓ Diagnosis details (Cancer Type, Stage, Diagnosis Date)

✓ Treatment information (Treatment Type, Hospital, Physician)

✓ Health-related factors (Smoking Status, Comorbidities, Weight, Height)

✓ Outcomes (Recovered, Under Treatment, Deceased)1.2 Justification of the Topic

(I) Intervention:

(Library and algorithm selection, e.g., pandas for financial data…)

[Which calculation am I primarily considering?]:

import pandas as pd # Import the pandas library for data manipulation and analysis

 # import numpy as np # Import the numpy library for numerical operations

import tensorflow as tf # Import the TensorFlow library for machine learning

from sklearn.model_selection import train_test_split # Import the train_test_split function to split data

from sklearn.preprocessing import StandardScaler, OneHotEncoder # Import StandardScaler for numerical feature scaling and OneHotEncoder for categorical feature encoding

from sklearn.compose import ColumnTransformer # Import ColumnTransformer to apply different transformations to different columns

 # from sklearn.pipeline import Pipeline # Import Pipeline to chain multiple data transformations and a model

import matplotlib.pyplot as plt # Import matplotlib for plotting

import seaborn as sns # Import seaborn for enhanced visualizations

from sklearn.metrics import classification_report, confusion_matrix, roc_curve, auc, precision_recall_curve, average_precision_score # Import various metrics for model evaluation

train_test_split to split the data into two separate sets: training and testing. This is super important for evaluating how well our model performs on data it hasn't seen before. Let's break down why:

- **Training set**: This is the data the model learns from — it fits the polynomial regression to these points.
- **Testing set**: This is the data the model has *never seen* — we use it to see how well the model can generalize to new data