Alim ARDAL 01/06/2023

Introduction to Artificial Intelligence

Project

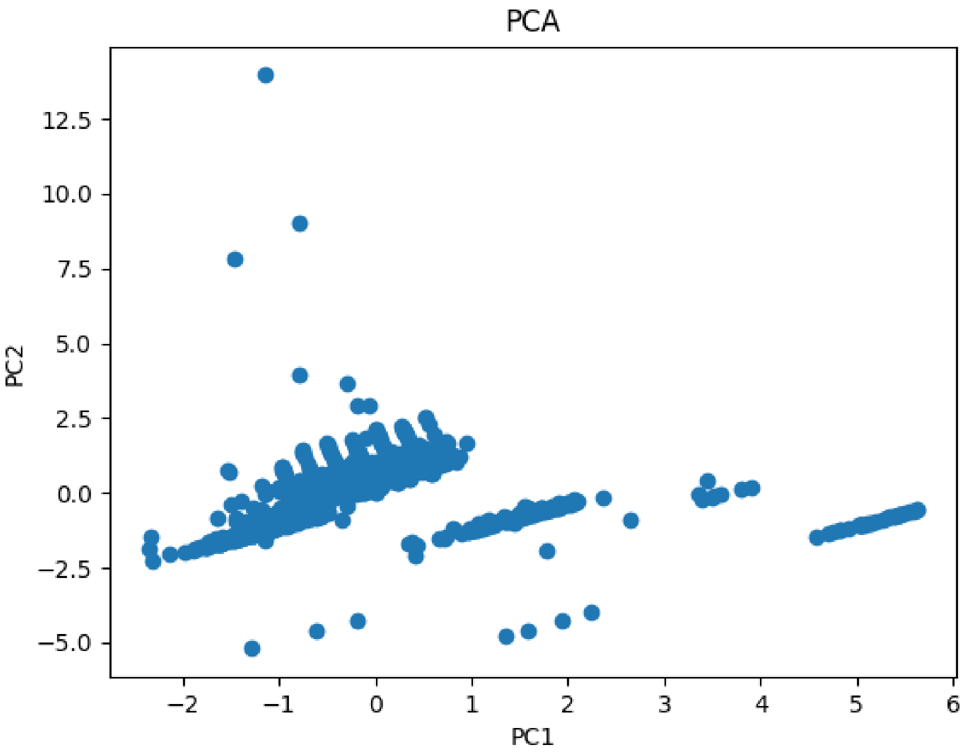
COVID-19 Infection Analysis and Prediction

1. Analysis of the dataset

A: The most correlated values are : (From Wuhan/Death = 0.35) (Age/Death = 0.26) (Symptom Onset/Death = 0.20) (Hospital visit date/Symptom Onset = 0.41)

This shows that People who died the most are from Wuhan, the place where the spreading of the virus started. Since there were no vaccine at this time and the virus was new, people were most likely to die if they were to be infected. The age was a pretty important factor too. The elderly needed to be cautious. Most people went to the hospital when they developed symptoms but some of them died.

B:



2. Bayes Nets

A: The probability of having symptoms given that the person visited Wuhan is 0.61

B: The probability of being a true patient given that the person has symptoms and visited Wuhan is 0.11

C: The probability of dying given that the person visited Wuhan is 0.01

D: The average recovery interval for patients who visited Wuhan is 23.25 days

3. Machine Learning

A: The confusion matrix shows the number of correct and incorrect predictions for each class. In our case, most correct predictions are in the first class (196 out of 202), indicating that this model predicts this class well.

Accuracy is the proportion of correct predictions among all predictions. The accuracy here is 0.91, meaning that 91% of the predictions are correct.

Recall is the proportion of true positives among all positive samples. The recall for the first class here is 0.97, meaning that 97% of positive samples from this class were correctly predicted. For other classes, recall is low or zero, indicating that our model does not predict these classes well.

The F1-score is a measure that combines precision and recall into a single metric. Here the F1-score for the first class is 0.95, indicating that our model has good performance for this class. For other classes, the F1-score is low or zero, indicating that our model does not predict these classes well.

Now, there are several ways to improve the results of this K-NN classification model.

1. Select the ideal number of neighbors (k): Cross-validation can be used to identify the ideal k value that produces the best results on the training data.

2. Normalize the data: When the data is normalized, distance-based algorithms like K-NN perform better. To scale the data, we can employ techniques like min-max normalization or standardization.

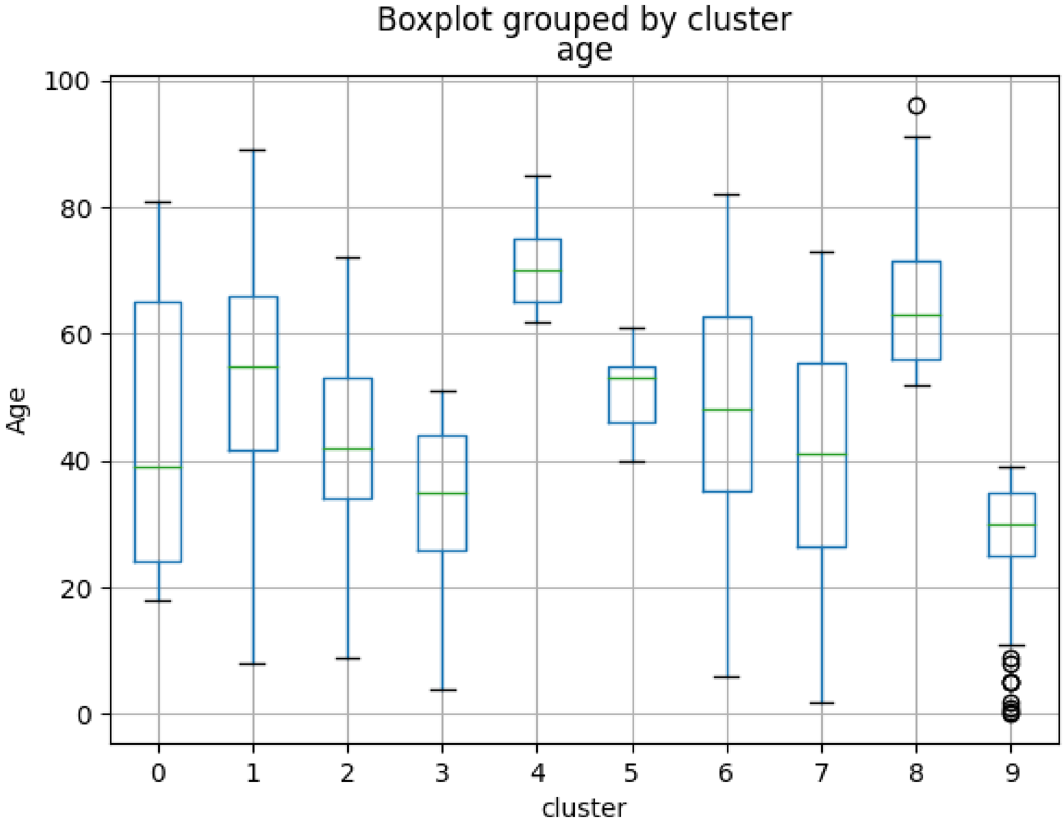
3. Pick the most pertinent features: Using feature selection approaches, we can pick the features that will most effectively aid in the prediction of the target variable.

4. Examine other classification methods to determine whether they produce better results. Examples of other classification algorithms are logistic regression, decision trees, and random forests.

B: In this code, the columns gender_female, gender_male, visiting Wuhan, and from Wuhan are chosen from the data to serve as the explanatory variables (features). To forecast the target variable age, these factors are fed into the linear regression model as input.

Building a predictive model involves making crucial decisions about the explanatory variables to use. The factors should be picked depending on how closely they relate to the target variable and how well they can help the model forecast the future. The most pertinent variables can be found via feature selection techniques, exploratory data analysis, or domain knowledge.

C:



In the box plot, the little circles that appear above and below the boxes signify outliers. An observation that deviates from the predicted range of values for a given variable is known as an outlier. A box plot's interquartile range (IQR), which ranges from the first quartile (Q1) to the third quartile (Q3), is represented by the box. The whiskers reach the values that are the furthest from 1.5 times the IQR. Outlier values are those that go outside of this range and are shown as little circles. Outliers can be used to spot samples with peculiar properties and study them more closely to ascertain why they differ from other samples.

4. Improving the results and Theoretical formalism

A: Balancing the data by reducing the majority class can help improve the prediction results by reducing the bias towards the majority class. This can lead to more accurate predictions for the minority class. However, it's important to note that this approach may result in a loss of information from the majority class.

B: Missing values in a dataset can be handled using a variety of techniques. One strategy is to make sure that data is appropriately gathered and entered to prevent missing numbers. A different strategy is to remove observations or variables with missing values. Another popular strategy is imputation, in which missing values are replaced with approximated values based on the available data. Imputing missing values can be done in several ways, such as by utilizing the mean, median, most prevalent, or zero or constant values. The type of data being used, and the particular issue being solved determine the technique to use.

C: The optimum hyperparameters for a machine learning model can be found via grid-search. Hyperparameters are variables that can be changed to influence how the model behaves. The performance of the model can vary depending on how different hyperparameter combinations are used.

For each hyperparameter we want to tweak, we define a set of potential values for it in grid-search. The algorithm then attempts every possible combination of these hyperparameter values and uses cross-validation to assess how well the model performs for each combination. Using distinct subsets of the data to train and test the model, cross-validation is a technique for estimating how well a model performs.

D: Grid-search is a brute-force method for finding the best hyperparameters for a machine learning model. It works by exhaustively trying all possible combinations of hyperparameters specified in a parameter grid and evaluating the performance of the model for each combination using a performance metric such as accuracy or mean squared error.

The performance of the model for each combination of hyperparameters is estimated using cross-validation. In k-fold cross-validation, the data is divided into k subsets of equal size. The model is trained on k-1 subsets and tested on the remaining subset. This process is repeated k times, with each subset serving as the test set once. The performance of the model for each combination of hyperparameters is then estimated as the average performance across the k folds.

The combination of hyperparameters that results in the best performance according to the chosen performance metric is then selected as the best hyperparameters for the model.

The impact of the hyperparameters on the results depends on the specific model and dataset being used. Different models have different hyperparameters that control their behavior, and the optimal values for these hyperparameters can vary depending on the characteristics of the data.

To compare our results with public results, we can look for published papers or benchmarks that use the same dataset and model as we and compare our performance to theirs. We can also try using different models and hyperparameter tuning methods to see if we can achieve better results.