Text

Description automatically generated

Riga Technical University Computer Systems Department

Fundamentals of Artificial Intelligence

Practical work 2: " **Applying methods of machine learning*”.***

GitHub link of the project:

https://github.com/shamilM2/Applying-methods-of-machine-learning

work prepared by

**Shamil Mammadrzayev 211ADB022**

**INDEX**

**1. Overview of the Orange tool workflow for the entire assignment**

**2. Part 1 – Pre-processing/Exploring the data**

**3. Part II – Unsupervised learning**

**4. Part III – Supervised learning**

**5.** **Sources Used**

**Overview of the Orange tool workflow**

**for the entire assignment:**

A picture containing diagram, text, line

Description automatically generated

**Part 1 – Pre-processing/Exploring the data.**

**Description of the dataset:**

* **Title**:

Pima Indians Diabetes Database

* **Source**:

UCI Machine Learning Repository

* **Author and/or owner of the dataset**:

National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK)

* **Dataset link**:

<https://www.kaggle.com/uciml/pima-indians-diabetes-database>

* **Licensing regarding the dataset (if any**):

CC0: Public Domain

* **A way how the dataset was collected**:

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The Pima Indians participated in one of the longest running diabetes studies, dating back 50 years. The Lancet reported in 1971 that diabetes is almost a way of life for the Pima, where type 2 diabetes and obesity are highly prevalent and occur at a younger age than in the general population. Although the main study ended in 2007, today, Pima Indians continue to participate in several studies of diabetes and its complications.

Source : <https://www.michiganmedicine.org/health-lab/disease-connection-answers-may-exist-within-arizona-tribe#:~:text=The%20Pima%20I>

* **Context**:

The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. All patients here are females at least 21 years old of Pima Indian heritage.

**Description of the content of the dataset:**

* **The number of data objects in the dataset:**

768 instances

* **A representation of the features (attributes) of the dataset together with their roles in the Orange tool**:

A screenshot of a computer

Description automatically generated with medium confidence

* **The number of classes in the dataset, the meaning of each class and the way of representing classes**:

There is only one target, which is Outcome. This Outcome shows us the presence of diabetes or its absence. So, we have two classes(0 and 1 respectively)

* **The number of data objects belonging to each class**:

500 of class 0 (absence of diabetes)

and 268 of class 1 (presence of diabetes)

* **The number and meaning of features in the dataset, as well as their value types and ranges**:

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Meaning | Value type | Range of values |
| Pregnancies | Number of times pregnant | Numeric | 0 to 17 |
| Glucose | Plasma glucose concentration (2 hours in an oral glucose tolerance test) | Numeric | 0 to 199 |
| Blood Pressure | Diastolic blood pressure (mm Hg) | Numeric | 0 to 122 |
| Skin Thickness | Triceps skin fold thickness (mm) | Numeric | 0 to 99 |
| Insulin | 2-Hour serum insulin (mu U/ml) | Numeric | 0 to 846 |
| BMI | Body mass index (weight in kg/(height in m)^2) | Numeric | 0 to 67.1 |
| Diabetes Pedigree Function | Indicates the function which scores likelihood of diabetes based on family history. | Numeric | 0.078 to 2.42 |
| Age | Age (years) | Numeric | 21 to 81 |
| Outcome | Class variable that indicates the diabetes | Binary | 0(absence of diabetes) and 1(presence of diabetes) |

*Table 1*

* **A snippet of the structure of your datafile in which the columns of your datafile and class labels are shown together with some data objects:**

**A screenshot of a computer

Description automatically generated with low confidence**

**Conclusions coming from the analysis of scatter plots, histograms, and distributions about the separability of my classes:**

* **Are classes in my dataset balanced, or is one class (several classes) prevailing? It is determined by how many data objects belong to each class.**

**A screenshot of a computer

Description automatically generated with low confidence**

We could clearly see that the dataset is not really balanced. The number of objects belonging to the class 0 (absense of diabetes) is slightly less than twice the number of objects belonging to the class 1( presense of diabetes)

Also, looking at the histograms, it is clear that there are more women who never had the pregnancy.

More women,who are younger.

Two of features are pretty much balanced : BMI and Blood Pressure

* **Does the visual representation of the data allow the structure of the data to be seen? It is a question of whether data objects belonging to different classes are separable**

For this question I am going to use the functionality of Orange tool which allows us to make program generate the most informative projections:  
A screenshot of a computer

Description automatically generated with medium confidence

As we can see from the picture, there is no such big difference between these projections.

**A picture containing text, screenshot, plot, diagram

Description automatically generated**

*(2-dimensional scatter plot illustrating the separability of classes in my dataset based on these features: Age(Axis x) and Glucose(Axis y))*

**A picture containing text, screenshot, plot, colorfulness

Description automatically generated**

*(2-dimensional scatter plot illustrating the separability of classes in my dataset based on these features: BMI(Axis x) and Glucose(Axis y))*

As we can see, even on the two best projections it is pretty hard to separete the classes, as intances are pretty merged, however there are some parts of the plot in which one kind of class is dominating more.

* **How many can data groupings be identified by studying the visual representation of the data? It is a question of whether there are any separable groupings of data if the data objects of different classes merge:**

As, stated previously it is seen that in my case, data objects of different classes are overlapping and merging.

* **Are the identified data groupings close to or far from each other?**

We could clearly see that they are close to each other.

By using Distributions element, we could clearly see that my data follows the normal distribution.

A screenshot of a computer

Description automatically generated with medium confidence

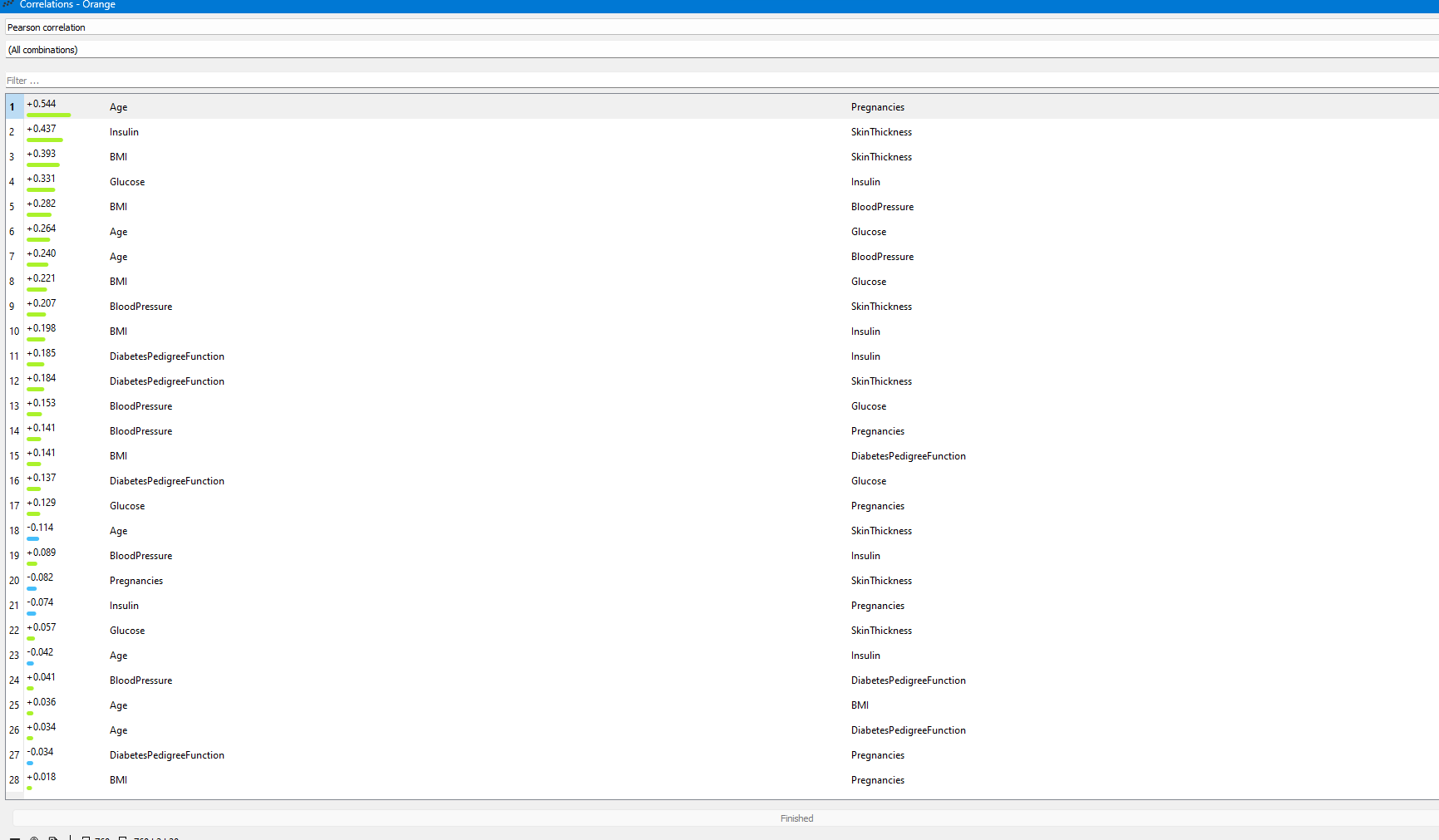
**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated with medium confidence**

Correlations:

****

*Correlations*

**Conclusions coming from the analysis of statistical calculations (central tendency and dispersion):**

**Central Tendency:**

1) For Pregnancies we could see that the histogram is right skewed which means we have more women with lower amount of pregnancy and trough the levels the number of women is decreasing.

2) For Glucose level mean value is pretty close to the center, so we have pretty symmetric distribution.

3) Same we could say for the Blood Pressure

4) In this case we have the very low number of means which shows us that Skin Thickness level is distributed very badly, also it is clear that the most of women have 0 level of Skin Thickness. The second most popular level is 2.

5) For Insulin histogram we could see again the lowest level dominating

6) The BMI level is pretty much balanced distributed. The biggest number of women have 5th and 4th levels.

7) Diabetes histogram is identical for mean values and the histogram structure to the first feature (Pregnancies)

8) The last histogram showing more younger women.

Dispersion:

Knowing that the higher value of dispersion indicates that for this feature we have the bigger spread amongst of the entire classes, we could verify that the bigger spread and diversity amongst Insulin, Age, Pregnancies, Diabetes Pedigree Function, Skin thickness, while the lower amount of dispersion for Glucose, BMI and Blood Pressure shows lower spread of distribution.

**A screenshot of a computer

Description automatically generated with medium confidence**

**Part II – Unsupervised learning**

First, based on the experiments I have maid, I have decided to include these features in order to obtain the maximum score:

**A screenshot of a computer

Description automatically generated**

I have tried to include/exclude different features, and this combination gave the best result, not considering the cases, where I was taking very few numbers of features, which oversimplifies it.

**1)K-Means Algorithm**

**Description of hyperparameters:**

1)Number of Clusters:

This parameter is to choose the number of the clusters. It could be either fixed or from X to Y(average).

2)Pre-processing:  
Allows to normalize the columns.

3)Initialization method:

Either k-Means++ (first center is selected randomly, subsequent are chosen from the remaining points with probability proportioned to squared distance from the closest center) or Random (clusters are assigned randomly at first and then updated with further iterations)

4)Re-runs

Allows to put the number of times that algorithm will rerun from initial positions.

5) Maximum iterations

Allows to set the maximum number of iterations.

Source: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/kmeans.html>

**My experiments:**

**A screenshot of a computer

Description automatically generated with medium confidence**

As it is possible to see on the picture, I have initialized the number clusters(k) from 2 to 6. My best Silhouette Score is with 2 cluster – 0.397.

Here is the list of scatter plots with better separability:

**A screenshot of a computer

Description automatically generated**

The plot:

**A screenshot of a computer screen

Description automatically generated with medium confidence**

*(2-dimensional scatter plot illustrating the separability of clusters in my dataset based on these features: Age(Axis x) and Pregnancies(Axis y))*

**Conclusion:**

It was very hard to obtain the high silhouette score without excluding nearly all the features. The maximum that it was possible to get with leaving at least 4 features was 0.397 which is very far from 1.

Nevertheless, the scatter plot illustrating the separability of clusters based on Age and Pregnancies looks pretty separable, excluding some merged points.

**2)Hierarchical clustering**

**Description of hyperparameters:**

1)Linkage parameter is to choose how to measure distances between clusters:

* Average – average distance between the closest elements of the two clusters
* Weighted - uses the WPGMA method.
* Complete - computes the distance between the clusters' most distant elements.
* Ward - computes the increase of the error sum of squares.
* Single - computes the distance between the closest elements of the two clusters.

**A screenshot of a computer

Description automatically generated with medium confidence** **A screenshot of a computer

Description automatically generated with medium confidence**

2) To choose the labels

3) To prune the dendrogram by selecting max depth.

4) In this “Selection” parameter it is possible to choose how to select the clusters:

* Manually
* By writing the height ration
* Or by selection top nodes

5) To Zoom

Source: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html>

**My experiments:**

A screenshot of a computer

Description automatically generated with medium confidence

1. In this case it is obvious that height ration around 90% does not change a lot of things, as with having 4 clusters, we still have very mixed clusters

A picture containing text, screenshot, software, multimedia software

Description automatically generated  
  
2) In this case having height ration of 39% and overall 69 clusters does not really solve our problem, as the clusters are still very diverse.

A screen shot of a graph

Description automatically generated with low confidenceA picture containing text, screenshot, software, display

Description automatically generated

3)In this third case we have managed to get one cluster by moving the height ration to 20%. This cluster mainly consist of class 0. However, there are still not so much separability.

**Conclusion:**

Overall, this algorithm worked not really well, because of diversity of the dataset, even with a smaller number of features, it is very hard to split the clusters in the way of splitting them by the classes. Even with moving the cut-off line it was hard to obtain good results.

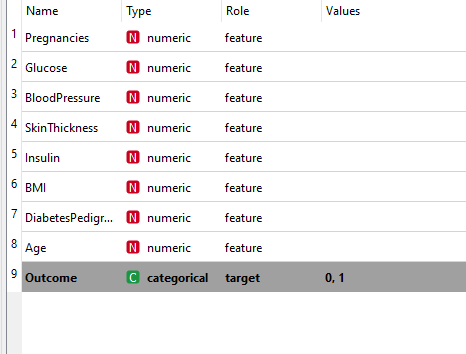
**Overall Conclusions:**

In my opinion I have done all the experiments in the correct way, but still was not able to get better separability of clusters. I think the reason for this is the specifics of this dataset, which provides very diverse values and makes it hard to cluster the objects.

Regarding, choosing the features to leave for this experiment:  
first of all, I was trying to rely on analysis of my statistics, obtained from previous part of assignment, but somehow the results were much worse. So, just by testing different combinations, I have found the best one (leaving at least 4 features, in order not to oversimplify the dataset).

**Part III – Supervised learning**

For supervised learning algorithms I have decided to include all features:



**Short description (1/3 of A4) of the essence of the supervised learning algorithms you have used and your motivation for choosing two of them (excluding the artificial neural network):**

For this part of the assignment, I have chosen these two algorithms aside the artificial neural network:  
1) kNN

This is the algorithm which is mainly for classification and regression. It is known as very simple algorithm. The essence of it is that it tries to find the neighbour which is most similar to the one that is labelled as goal class. After finding it, it labels this neighbour with that class. To make sure if the neighbour is similar, the algorithm uses the metrics such as distance between data objects, as mostly similar data objects are close to each other.

2)Logistic Regression

This is the algorithm which is used for Binary classification tasks. The binary classification tasks are the one where the output could take one of two possible values (0 or 1). For example, this algorithm could predict success/failure, good/bad, yes/no, win/loss, etc.

Logistic regression algorithm takes liner function results, applies them to the sigmoid function, and outputs the values based to their class. It finds the correct answers by adjusting the weights used in calculations.

Overall, I think these two algorithms are suitable for my case, as I have the classification type. More specifically, Binary Classification task which is ideal for using the Logistic Regression algorithm.

**Description of the hyperparameters available in the Orange tool and their meaning for each algorithm:**

**kNN:**

1)You can set the number of neighbours

2) Choose one of metric systems

Euclidean ("straight line", distance between two points)

Manhattan (sum of absolute differences of all attributes)

Maximal (greatest of absolute differences between attributes)

Mahalanobis (distance between point and distribution).

3)Choose which weights you want to use:

Uniform: all points in each neighbourhood are weighted equally.

Distance: closer neighbours of a query point have a greater influence than the neighbours further away.

Source: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/knn.html>

**Logistic Regression:**

1. You can choose the regularization type:  
   L1 or L2.
2. You can change the Strength. Default is = 1.

Source: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/logisticregression.html>

**Neural Network:**

1. Here it is possible to choose the number of the neurons per hidden layer. For example, if there are three layers, it could be defined as 2,3,2.
2. To choose activation function:
   * Identity: no-op activation, useful to implement linear bottleneck
   * Logistic: the logistic sigmoid function
   * tanh: the hyperbolic tan function
   * ReLu: the rectified linear unit function
3. To choose solver:

* Identity: no-op activation, useful to implement linear bottleneck
* Logistic: the logistic sigmoid function
* tanh: the hyperbolic tan function
* ReLu: the rectified linear unit function

1. To choose the Alpha: L2 penalty parameter
2. To choose the number of maximum iterations

Source: <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/neuralnetwork.html>

**Test and Train datasets amount:**

**A screenshot of a computer

Description automatically generated with medium confidence**

70% of the Data was used for train data. 538 out of 768 instances.

349 instances of class 0(absence of diabetes) and 189 instances of class 1(presence of diabetes) used for test data.

30% of the Data was used for test data. 230 out of 538 instances

151 instances of class 0(absence of diabetes) and 79 instances of class 1(presence of diabetes) used for test data.

**Logistic Regression Train**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Experiments | Regularization Type | Strength | Predicted correctly for class 0 | Predicted correctly for class 1 |
| Experiment 1 | L2 | 50 | 81% | 75.8% |
| Experiment 2 | L2 | 1 | 78.5% | 76.7% |
| Experiment 3 | L2 | 0.09 | 65.8% | 71.4% |
| Experiment 4 | L2 | 10 | 80.9% | 76.2% |
| Experiment 5 | L2 | 20 | 81% | 75.8% |

*Table 2*

**Conclusions**:

We could see in the Table 2 that the amount of predicted is changing with change of strength. It is hard to choose the model to use, as there is one showing more symmetric results, but another one hits record for one of class.  
I decide to choose model used in experiment 1 as it hits max predictions for class 1.

**kNN Train:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Experiments | Number of neighbours | Metric | Weight | Predicted correctly for class 0 | Predicted correctly for class 1 |
| Experiment 1 | 100 | Euclidian | By Distances | 71.9% | 77.6% |
| Experiment 2 | 5 | Euclidian | By Distances | 76.7% | 63.1% |
| Experiment 3 | 50 | Euclidian | By Distances | 74.5% | 75.5% |
| Experiment 4 | 80 | Euclidian | By Distances | 72.5% | 77.1% |
| Experiment 5 | 50 | Euclidian | Uniform | 73.8% | 78.3% |

*Table 3*

**Conclusions**:

Looking at the results of this experiments in Table 3, it is clear that the most successful one was the Experiment 3, as it nearly gave the same amount of successfully predicted for both of classes. That is the model that I am going to use.

**Neural Network Train:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Experiments | Neurons in hidden layer | Activation | Solver | Regularization | Max number of iterations | Predicted correctly for class 0 | Predicted correctly for class 1 |
| Experiment 1 | 1,1,1 | Identity | SGD | 0.1 | 1000 | 79% | 68.3% |
| Experiment 2 | 2,3,2 | Identity | SGD | 0.1 | 1000 | 80% | 71.5% |
| Experiment 3 | 2,3,2 | Identity | SGD | 2 | 1000 | 79.9% | 72.7% |
| Experiment 4 | 5,50,100 | Identity | SGD | 0.1 | 1000 | 80.9% | 74.8% |
| Experiment  5 | 5,50,100 | Identity | SGD | 1 | 1000 | 80.8% | 75.7% |

*Table 4*

**Conclusions**:

In these experiments I had nearly same results in each of experiments which could be seen in Table 4, although the last one was the most successful. So, I am choosing model from Experiment 5.

**Test Results**

**A screenshot of a computer

Description automatically generated with medium confidenceLogistic Regression:**

**A screenshot of a graph

Description automatically generated with low confidenceA screenshot of a graph

Description automatically generated with medium confidence**Train Results vs Test Results

**kNN Algorithm:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a graph

Description automatically generated with medium confidenceA screenshot of a graph

Description automatically generated with medium confidence**Train Results vs Test Results

**Neural Network:**

**A screenshot of a computer

Description automatically generated**

Train Results vs Test Results

**A screenshot of a graph

Description automatically generated with medium confidenceA screenshot of a graph

Description automatically generated with medium confidence**

**Conclusions:**

Overall, it is possible to say that for Logistic Regression there is some noticeable change between the results of train model and test model. The same applies for the Neural Network. Both of them have significant decrease in the correctly predicted number of objects.

While on other side, the kNN algorithm showed better performance. In comparison of train and test model it showed pretty same results, especially for class 0.

The test model of kNN algorithm in comparison with test models of other algorithms is the most balanced. As it provides similar results for both of classes. For the Logistic Regression and Neural Networks, we could see big differences in predictions of different classes.

All results could have been better in case of different chosen combinations of the features included.

**ROC Analysis:**

Train data for class 0:

**A screen shot of a graph

Description automatically generated with medium confidence**

Train data for class 1:

A screen shot of a graph

Description automatically generated with medium confidence

**Test data for class 0:**

**A screenshot of a graph

Description automatically generated with medium confidence**

**Test data for class 1:**

**A screenshot of a graph

Description automatically generated with medium confidence**

**Sources Used:**

**1)** <https://www.kaggle.com/uciml/pima-indians-diabetes-database>

**2)** <https://www.michiganmedicine.org/health-lab/disease-connection-answers-may-exist-within-arizona-tribe#:~:text=The%20Pima%20I>

**3)** <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/kmeans.html>

**4)** <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/unsupervised/hierarchicalclustering.html>

**5)** <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/knn.html>

**6)** <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/logisticregression.html>

**7)** <https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/model/neuralnetwork.html>