UNIVERSIDAD DE MONTERREY

Artificial Intelligence

K-Nearest Neighbours

"Predicting the onset of diabetes based on diagnostic measures with a KNN-based classifier"

Technical Report

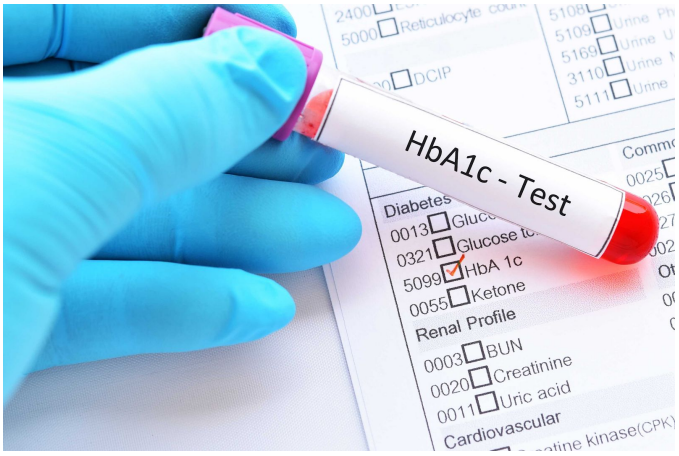

Figure 1. HbA1c Test for Diabetes.

PROFESSOR: DR. ANDRÉS HERNÁNDEZ GUTIÉRREZ

STUDENTS: Elizondo Benet, Diego
567003

Flores Ramones, Alejandro
537489

Lira Rangel, Karla Alejandra
526389

Monday December 14th, 2020

We hereby declare that we have worked on this project with academic integrity.

0. Index

1. Introduction

For this project, the task at hand consists in being able to classify a person or patient into one of two possible categories: likely to be suffering from diabetes, or not likely to be suffering from diabetes. To tackle this problem, as it will be explained afterwards in this report with further detail, a KNN-based classifier was developed.

The National Institute of Digestive and Kidney Diseases (NIDDK, 2016) succinctly states on its website: ‘diabetes is a disease that occurs when your blood glucose, also called blood sugar, is too high’.

As this project will be focused on diabetes prediction, it is important to know which factors increase the probability of someone suffering from diabetes. Once again, the NIDDK (2016) indicates that you are more likely to develop type 2 diabetes if you are age 45 or older, have a family history of diabetes, or are overweight. Physical inactivity, race, and certain health problems such as high blood pressure also affect your chance of developing type 2 diabetes. You are also more likely to develop type 2 diabetes if you have prediabetes or had gestational diabetes when you were pregnant.

The relevance of working with this problem is evident. For example, in the case of the United States, around 10% of the population suffers from diabetes, and it was the 7th leading cause of death of 2017. Additionally, about a fourth of people who suffer from diabetes goes undiagnosed, which is problematic as they don’t receive the medical attention that they need to help with their condition (American Diabetes Association, 2018).

This project consists in writing a KNN-based classifier in Python. Briefly put, K-Nearest Neighbours (KNN), consists in predicting whether a data point belongs to a certain class depending on the classes of the K closest points to it. This classifier will be trained with data from a real-world diabetes diagnostic measurements dataset obtained from Kaggle

It's worth mentioning that first, said dataset was analyzed in the exploratory data analysis stage. Following that, the classifier was trained, and then it was put to the test with a portion of the dataset not seen during training, that is, the testing data. Finally, the confusion matrix was obtained, as well as several performance metrics derived from those values.

2. Methodology

3.1. General Exploratory Data Analysis

The diabetes dataset used for this project was found on Kaggle (https://www.kaggle.com/uciml/pima-indians-diabetes-database). It contains samples originally from the National Institute of Diabetes and Digestive and Kidney Diseases.

It is important to note that all of the individuals included in the dataset are females over 20 years old, with Pima Indian heritage. This is a highly relevant fact, as we can't expect the KNN-based classifier trained with this data to work just as well with different populations. It might still work, but it is highly unlikely that the predictions would be as accurate.

The dataset consists of 768 samples obtained from real cases observed by the National Institute of Diabetes and Digestive and Kidney Diseases. It contains 8 different attributes, which are:

- Number of pregnancies.
- Plasma glucose concentration.
- Diastolic blood pressure.
- Triceps skin fold thickness.
- Serum insulin.
- Body Mass Index.
- Diabetes pedigree function.
- Age.

All of the values in this dataset are numeric, with only 2 of the features consisting of floating point values (BMI and Diabetes Pedigree Function), while the remaining 6 consist of whole numbers. Additionally, there are 0 NaN entries throughout the entire dataset. This means that the data is complete, and no entry has a missing attribute value.

Now, each feature will be shown, along with its distribution, its minimum and maximum value, its mean and median, as well as the number of NaN values.

1. Number of pregnancies (x_1)

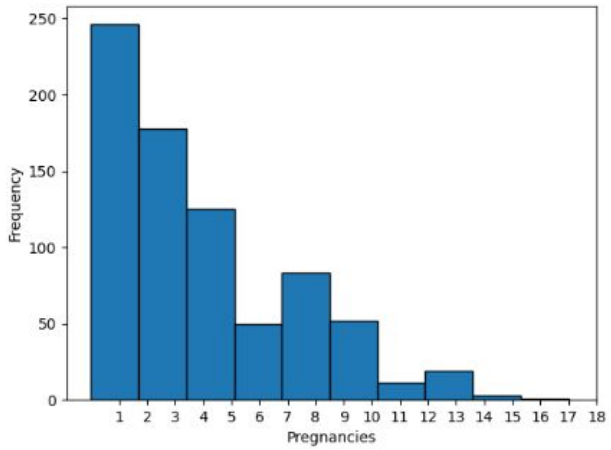

Figure 2. Pregnancies distribution.

Aspect	Value
Min	0
Max	17
Mean	3.85
Median	3
# of NaN	0

Table 1. Pregnancies statistics.

This feature's distribution makes sense, as with each additional pregnancy it is rarer to find a woman that has reached said number.

2. Plasma glucose concentration (x_2)

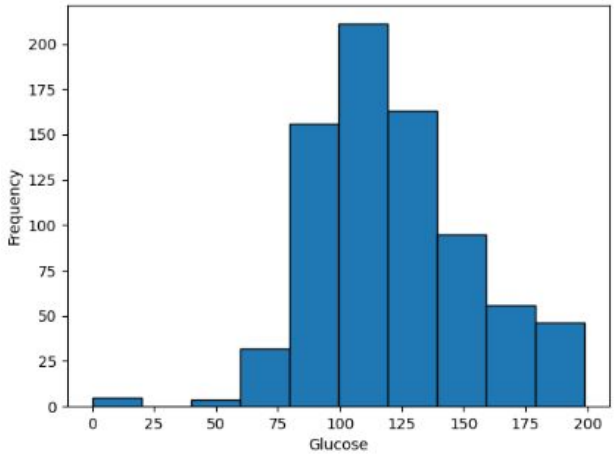

Figure 3. Glucose distribution.

Aspect	Value
Min	0
Max	199
Mean	121
Median	117
# of NaN	0

Table 2. Glucose statistics.

This feature seems to follow a normal distribution.

3. Diastolic blood pressure (x_3)

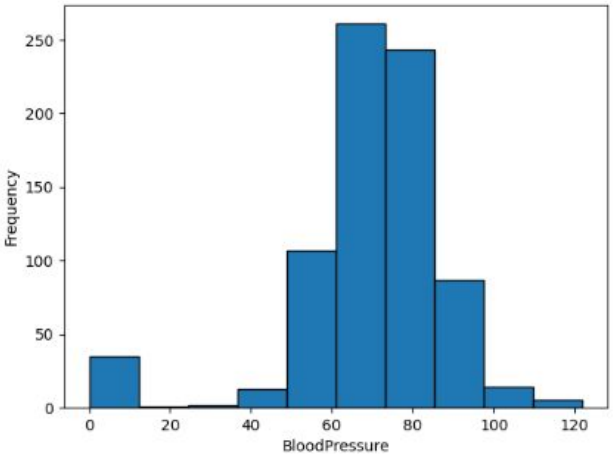

Figure 4. BloodPressure distribution.

Aspect	Value
Min	0
Max	122
Mean	69.1
Median	72
# of NaN	0

Table 3. BloodPressure statistics.

This feature also seems to be normally distributed, with some possible outliers on the left.

4. Triceps skin fold thickness (x_4)

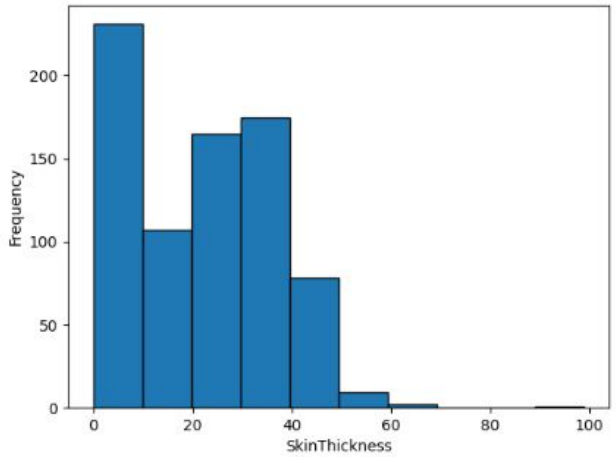

Figure 5. SkinThickness distribution.

Aspect	Value
Min	0
Max	99
Mean	20.5
Median	23
# of NaN	0

Table 4. SkinThicknes statistics.

Most samples seem to be of a low skin thickness.

5. Serum insulin (x_5)

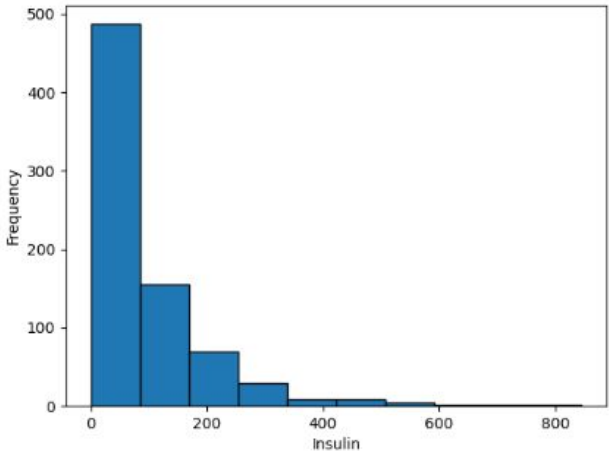

Figure 6. Insulin distribution.

Aspect	Value
Min	0
Max	846
Mean	79.8
Median	32
# of NaN	0

Table 5. Insulin statistics.

Once again this distribution has a high positive skew, with a steep drop after the first bar.

6. Body Mass Index (x_6)

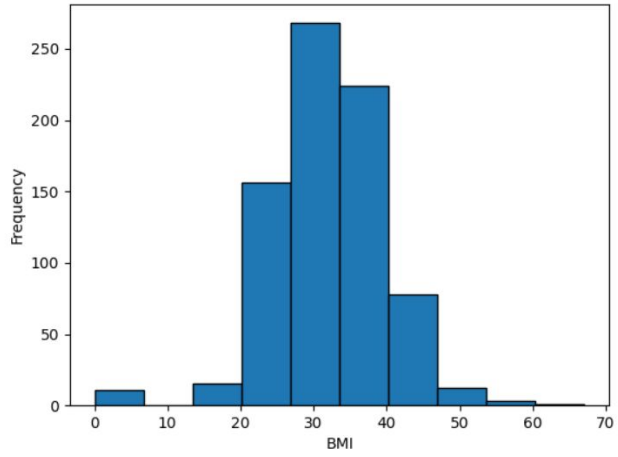

Figure 7. BMI distribution.

Aspect	Value
Min	0
Max	67.1
Mean	32
Median	32
# of NaN	32.2

Table 6. BMI statistics.

BMI seems to be normally distributed, as it is to be expected.

7. Diabetes pedigree function (x_7)

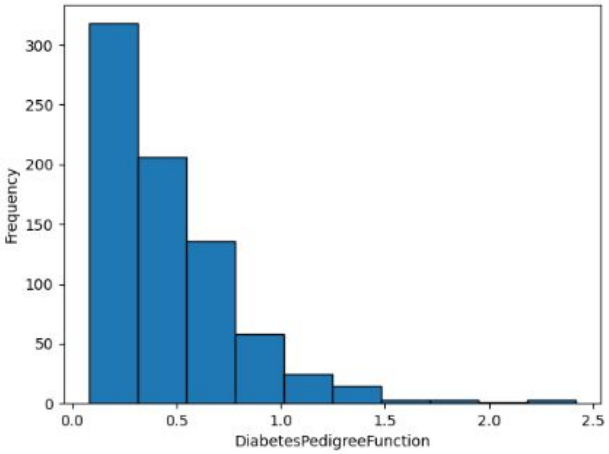

Figure 8. DiabetesPedigree distribution.

Aspect	Value
Min	0.08
Max	2.42
Mean	0.47
Median	0.37
# of NaN	0

Table 7. DiabetesPedigree statistics.

This feature also shows a skew to the right, but the drop is relatively gradual compared to the serum distribution.

8. Age (x_8)

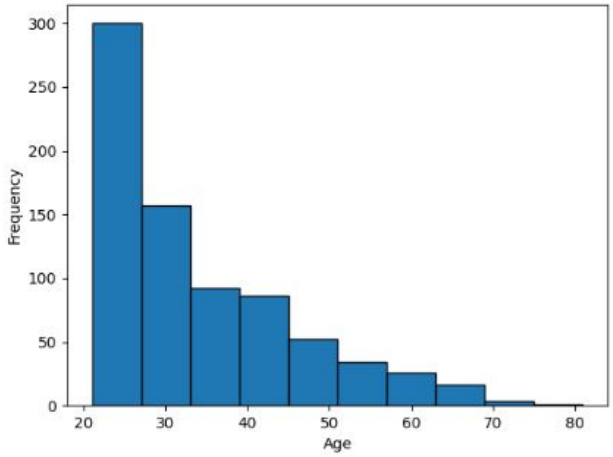

Figure 9. Age distribution.

Aspect	Value
Min	21
Max	81
Mean	33.2
Median	29
# of NaN	0

Table 8. Age statistics.

One more time, this feature also shows a positive skew, but this time the drop seems to be between that of the serum distribution and that of the diabetes pedigree function.

In regards to the labels, a sample is either labelled with a 0, which represents an absence of diabetes; or with a 1, which represents that the patient suffers diabetes. Just like the 8 features shown previously, here's the numeric information about the labels:

9. Outcome (y)

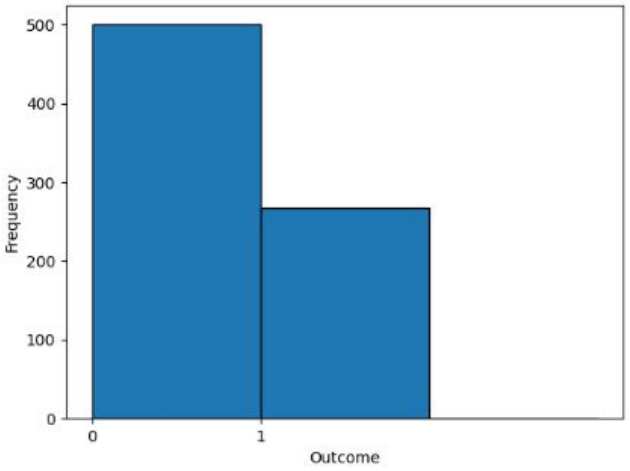

Figure 10. Outcome distribution.

Aspect	Value
Min	0
Max	1
Mean	0.35
Median	0
# of NaN	0

Table 9. Outcome statistics.

As it can be observed by the previous distribution of the labels, this dataset is not fairly balanced, consisting of 268 (about 34.896% of the dataset) negative classes; and 500 (about 65.104% of the dataset) positive classes.

3.2. Particular Exploratory Data Analysis

First, the dataset is read from a .csv file, that is, a comma-separated values file. The first line of the file contains the titles of each of the attributes, each separated by a comma, while the remaining lines are the dataset samples. Again, each of the lines corresponding to the samples contains the value for each feature separated by a comma. This .csv file is read, and afterwards it is randomly shuffled. Then, it is split: leaving the first 95% of the dataset to be used as the training dataset, and the other 5% for the testing dataset.

To get an idea about the samples included in the dataset, 10 samples from the training dataset are randomly selected and visualized. For example, the following 10 samples:

Training Dataset									
Sample	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	y
1	8	196	76	29	280	37.5	0.61	57	1
2	14	175	62	30	0	33.6	0.21	38	1
3	13	152	90	33	29	26.8	0.73	43	1
4	2	93	64	32	160	38	0.67	23	1
5	1	114	66	36	200	38.1	0.29	21	0
6	8	126	74	38	75	25.9	0.16	39	0
7	1	71	62	0	0	21.8	0.42	26	0
8	6	114	88	0	0	27.8	0.25	66	0
9	5	109	75	26	0	36	0.55	60	0
10	8	188	78	0	0	47.9	0.14	43	1

Table 10. Training dataset randomly selected samples.

Just like with the training dataset, another 10 samples were visualized, but this time they were randomly selected from the testing dataset:

Testing Dataset									
Sample	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	y
1	5	136	82	0	0	0	0.64	69	0
2	0	84	82	31	125	38.2	0.23	23	0
3	1	97	70	15	0	18.2	0.15	21	0
4	7	142	90	24	480	30.4	0.13	43	1
5	4	114	65	0	0	21.9	0.43	37	0
6	3	132	80	0	0	34.4	0.4	44	1
7	4	131	68	21	166	33.1	0.16	28	0
8	5	166	72	19	175	25.8	0.59	51	1
9	1	139	46	19	83	28.7	0.65	22	0
10	4	114	65	0	0	21.9	0.43	37	0

Table 11. Testing dataset randomly selected samples.

Additionally, some statistics are computed about each of the 8 features, both for the training dataset and for the testing dataset. The results are included in the following tables:

Training Dataset								
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Min	0	0	0	0	0	0	0.08	21
Max	17	199	122	99	846	67.1	2.42	81
Mean	3.84	120.7	69.1	20.6	78.7	32.1	0.47	33.2
Median	3	117	72	23	32	32.2	0.38	29

Table 12. Training dataset statistics before normalization.

Testing Dataset								
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
Min	0	73	0	0	0	0	0.13	21
Max	12	195	106	49	480	45	1.73	69
Mean	3.9	124.2	69.4	19.6	99.6	29.1	0.36	12.1
Median	4	119	70	21	0	30.4	0.34	33

Table 13. Training dataset statistics before normalization.

3.3 Design of the Model

The first step of the process is to normalize the training dataset, as well as the testing dataset. To achieve this, the method which involves the mean and the variance was used, as requested by the client. After normalization, the data values for each feature fall between a much more limited range (they are closer together) and are distributed with a mean located at 0, which is great for numeric stability. Normalization is particularly important in this case, because the glucose feature has a value as big as 199, while the diabetes pedigree function has a value as small as 0.08. To achieve this normalization the following formula was used:

$$x \leftarrow \frac{x - \mu}{\sqrt{\sigma^2 + 10^{-8}}}$$

Once the data is normalized, the euclidean distances are calculated. The euclidean distance is a way to calculate how close or similar to samples are. A smaller value for the euclidean distance between two samples means that they are more similar to one another, compared to the case where the euclidean distance provides a higher value. An euclidean distance of 0 would mean that two samples have the exact same values for each or their attributes. Therefore, this distance is not limited to 2D space, which would be the distance between two points, but rather the euclidean distance can perfectly work for samples with N dimensions. In this case, there are 8 attributes or features.

The euclidean distance between two samples is defined as the square root of the sum of the squared differences between each of the values of their attributes. That is, the square root of: the squared difference between the value of the first attribute of sample A and the value of the first attribute of sample B, plus the squared difference between the value of the second attribute of sample A and the value of the second attribute of sample B, and so on until the last attribute.

More specifically, for each of the testing samples the euclidean distance to all of the training samples is calculated and stored in a matrix. To compute the conditional probabilities, for each testing sample the calculated euclidean distances are sorted in increasing order, and it is looked at the first K distances, which represent the K closest training points to the testing point. From these K distances, the number of training samples of each class is counted. Based on these two counts (number of negative classes and number of positive classes), the conditional probabilities are calculated by dividing the count of each class by K.

After the conditional probabilities are calculated, a prediction can be made for each of the testing samples based on which of the two probabilities is higher, the one corresponding to the negative class, or the one corresponding to the positive class. The testing samples are predicted to be of the class as the one corresponding to the higher conditional probability. This means that the testing samples are predicted to be of the same class as the majority of its K nearest neighbours.

In the case where the conditional probabilities are the same, which means that there were an equal number of samples of each class in the K-nearest neighbours, the prediction has to be made based on additional information. For this reason, besides the conditional probabilities, the sum of the distances of the K-nearest neighbours of each class is passed to the method which performs the predictions. If the probabilities are the same, the class corresponding to the smaller sum would be the prediction, which means that if there were an equal amount of training samples of each class, it is looked at which were closer to the testing sample.

4. Experimental Results: Testing

After the predictions were made for each of the testing samples, it is important to compare those predictions to the actual classes of the samples, in order to obtain the confusion matrix. With the confusion matrix, several metrics can be calculated, which indicate the performance of the classifier. In this project, those metrics were calculated for different values of K, to be able to observe how this parameter plays a role in the classification results.

k-nearest neighbours	Accuracy	Precision	Recall	Specificity	F1-Score
k = 3	0.77	0.67	0.71	0.8	0.69
k = 5	0.77	0.69	0.64	0.84	0.67
k = 10	0.85	0.79	0.79	0.88	0.79
k = 15	0.85	0.79	0.79	0.88	0.79
k = 20	0.85	0.83	0.71	0.92	0.77

Table 14. Training dataset statistics before normalization.

From the previous table, perhaps the first aspect that can be observed is that in general, as k increases, the classifier tends to perform better, as can be seen by the increase of the various metrics. For example, accuracy went from 0.77 to 0.85, and precision from 0.67 to 0.69 to 0.79 to 0.83. A similar increase occurred with the specificity. With those three metrics, a higher k always showed better or equal performance than the previous k value. The reason for this might be the fact that looking at more neighbours gives a more accurate result by not limiting the view with too few samples. An analogy could be made to statistics, where a small sample has a higher chance of not being representative of the population.

However, with the Recall and the F1-Score, the last row, where k is equal to 20, showed a decreased performance compared to the previous k. This could be attributed to the fact that possibly for these specific samples, when looking at the nearest neighbours for a sample which is actually positive, increasing k led to including some negative training samples which resulted in the prediction also being negative, and therefore said positive

sample not being predicted as such. This would decrease the recall metric, which would then decrease the F1-Score, as it is included in its calculation.

Another reason for the decrease in the final metrics might be that looking at too many neighbours goes beyond what is necessary for making a correct prediction, and brings into consideration samples which are not closely related enough to the testing sample.

Based on the previous table, going forward with this project for the remaining discussions and visualization of the results, k being 10 will be chosen. This was due to various reasons. First, it is important to note that the F1-Score metric is more useful than accuracy for datasets which are not balanced, which was the case with this project. Additionally, recall is possibly the most important metric for this problem of diabetes classification, as with health conditions in general, it is of great importance to identify when someone does have the disease. If they are diagnosed, they can receive proper medical attention and adjust their life to better deal with the condition. The values of k where the F1-Score and recall were the highest were 10 and 15, and 10 was chosen due to the fact that it is lower than 15 and, even if by a negligible margin, therefore computationally faster to calculate the results with it.

The confusion matrix for k = 15, which was the chosen parameter as described on the previous paragraph, is shown below:

		Predicted Class	
		Has diabetes (1)	No diabetes (0)
Actual Class	Has diabetes (1)	True Positive 11	False Negative 3
	No diabetes (0)	False Positive 3	True Negative 22

Table 15. Confusion matrix obtained from the testing dataset.

From the previous confusion matrix, several metrics were obtained, as shown below. Said metrics will be discussed next.

Metric	Value
Accuracy	0.85
Precision	0.79
Recall / Sensitivity	0.79
Specificity	0.88
F1-score	0.79

Table 16. Testing metrics calculated from the confusion matrix.

First, looking at the accuracy, the classifier seems to perform well, at least when working with the samples gathered for the testing dataset. Perhaps it is not extremely accurate, as the correct predictions were only 85% and this problem is related to patients' health, but considering the fact that the dataset was fairly small and the process takes extremely little time to compute, it is not a bad result for the classifier.

Moving on to precision, which indicates what fraction of positive predictions is actually positive, we can see once again that although it is lower than precision, it is still relatively good. For the diabetes prediction problem, this metric can be related to how many patients which were diagnosed with diabetes actually had it. This is important, as it is not great to tell someone who doesn't have diabetes that they suffer from it, as they could get depressed and feel bad about the news. However, this can be mitigated by always getting second opinions when referring to the medical field.

Recall will be discussed at the end, as it plays a special role on the diabetes classification problem. Advancing with specificity, which was also good (0.88), is related to the previous point, as it indicates, out of all the negative classes, the proportion that was predicted as negative by the KNN-based classifier.

Regarding the F1-Score, it is important to remember that this metric is useful when either precision is high and recall is low, or the opposite. However, in this case both of those metrics are 0.79, so it is not as useful. It makes sense then why the F1-Score was also 0.79.

Finally, recall will be discussed. For the problem that this project is dealing with, that is, identifying someone as suffering from diabetes or not, recall, which resulted in 0.79, is a very important metric. Recall or sensitivity measures the fraction of samples correctly predicted as positive out of all of the samples that were actually positive. That is, out of all the patients who suffered from diabetes, how many of them were actually predicted to suffer from diabetes.

If an individual actually has diabetes but they're told that they're perfectly fine (false negative), there can be some consequences. For example, perhaps the patient could have received medical treatment or made adjustments to their routine to deal with the diabetes, but as they were told that they don't suffer from it, they went on with life without any adjustments, and therefore later their health could become complicated. On the other end, if they were correctly diagnosed with the condition, they could be treated for it and improve their quality of life.

5. Conclusions

For every problem it is important to first understand where the data comes from, as well as what are the most critical aspects of the results after testing what was developed. In this case, as mentioned before, recall is a good metric for health classification problems, and with this KNN-based classifier its result was relatively good, with a value of 0.79.

In general, this binary classifier performed relatively well, but there is ample room for improvement, as none of the metrics went above 0.95, and with health-related problems and dealing with the lives of people, such as when talking about diagnosing diabetes, the accuracy needs to be as high as possible.

A possible use for this classifier, as it is not extremely accurate but it still had a relatively good performance, would be for it to be implemented on a website, where people could submit their relevant diagnostic measurements, and the classifier would tell them whether they have a high probability of suffering from diabetes or not. However, with the presence of any serious indicators or concerns patients would still need to go to the doctor, as with any other health-related issue, in order to get it properly checked out and to receive a

more accurate diagnosis based on professional medical advice and more in-depth analysis on the individual.

Another important point to highlight is that this dataset consisted only of females over 20 years old with Pima Indian heritage. This means that the classifier would probably not be as good for other populations. It could still perform well, as diabetes can affect many different people and perhaps is not as dependent on certain factors as other biologic aspects such as race and melanin being related to sunburns and vitamin-D absorption. But for example, the attribute of the number of pregnancies could not be obtained from males, so that would need to be adjusted. The website idea mentioned earlier could be improved by letting people update if they were later diagnosed with diabetes, and use their data as another training sample, which would over time generate a dataset from a wider and more varied population.

As a final note, it was a great surprise that the classifier performed as well as it did with such a little dataset and with the process not being computationally intensive. In a way it makes perfect sense, as looking at similar samples in order to make a prediction, which is the core idea behind the K-Nearest Neighbours algorithms, clearly seems like an effective way to make guesses. People do something that closely resembles that in everyday life, where we make internal predictions for new scenarios based on previous similar experiences.

6. References

[Figure 1] Retrieved from: https://www.diabetes.co.uk/wp-content/uploads/2019/01/iStock-987670332.jpg

American Diabetes Association. (2018). *Statistics About Diabetes*. Retrieved from: https://www.diabetes.org/resources/statistics/statistics-about-diabetes

National Institute of Digestive and Kidney Diseases. (2016). *What is Diabetes?*. Retrieved from: https://www.heartresearch.com.au/heart-disease/what-is-heart-disease/

Hernández, A. (2020). *K-Nearest Neighbours* [Slides].

UCI Machine Learning (Kaggle user). (2016). *Pima Indians Diabetes Database*. Retrieved from: https://www.niddk.nih.gov/health-information/diabetes/overview/what-is-diabetes