Sarthak Dhingra
1532881
December 20th, 2021

CMPUT 466 Mini-Project

INTRODUCTION

The task of this project is to predict whether or not a person has diabetes based on several physical attributes. This project uses the PIMA Indians diabetes dataset from the National Institute of Diabetes and Digestive and Kidney Diseases to accomplish this task. The full dataset can be found in **data/diabetes.csv** in the repository. More information about the dataset can be found here: https://www.kaggle.com/uciml/pima-indians-diabetes-database.

The dataset contains 768 data samples. There are 8 input fields and one output field. The input fields are: number of pregnancies, plasma glucose concentration, diastolic blood pressure, skin thickness, insulin level, body mass index, diabetes pedigree function, and age. The output, Outcome, is a binary value where 1 indicates that the person has diabetes, and 0 indicates the person does not have diabetes. This is a supervised binary classification task where the goal is to use machine learning algorithms to predict the Outcome, based on these 8 input attributes. Below is an example of one data sample:

Pregnancies	Glucose	Blood Pressure	Skin Thickness	Insulin	BMI	Diabetes Pedigree Function	Age	Outcome
1	89	66	23	94	28.1	0.167	21	0

Diabetes is a condition in which the human body has difficulty producing or using insulin that negatively impacts millions of people. Some consequences of diabetes are blindness, kidney failure, and heart disease. As such, being able to predict whether someone has the disease is incredibly important.

This project compares the performance of 3 machine learning algorithms on the PIMA Indians diabetes dataset. Logistic regression, k-nearest neighbours, and a neural network are all used to predict whether someone has diabetes from these 8 input attributes.

APPROACH

The code for this project made use of pandas, numpy, matplotlib, sklearn, and karas for data processing + visualization and implementation of the machine learning algorithms. All algorithms made use of the train-validation-test framework, in which a subset of the dataset is used to train the machine learning model, another subset is used to validate its performance in training, and another is used to test the model after training has been completed. The evaluation metric is accuracy, namely the percentage of correct predictions. For example an accuracy of 0.68 means 68% of the models predictions were correct.

Data Processing

The data contains values of 0 in many fields where it does not make sense to have a value of 0. A value of 0 does not make sense in the categories of Glucose, BloodPressure, SkinThickness, Insulin, and BMI. This occurs because 0 was recorded for missing data samples when the data was collected. To improve performance, we will impute the mean category value (not including the 0s) for values of 0 in the mentioned categories. The data was also normalized to improve performance so that each category had a mean of 0 and a standard deviation of 1. All experiments were performed on the same normalized and imputed data.The training, validation, and test framework used a 60%, 20%, and 20% split respectively.

Baseline - Majority Guess

The trivial baseline algorithm for this task is a majority guess. In a majority guess you predict the majority class 100% of the time. In this task 500/768 of the samples have an outcome of 0. As such the majority class is 0, a person not having diabetes.

K-Nearest Neighbours

The k-nearest neighbours algorithm (KNN) is an algorithm that calculates the “distance” between one data sample and all other data samples. It then sorts the collection, and predicts a given data sample's output as the mode output of the K samples with the nearest distances. The hyperparameters considered in this algorithm were the value of K and the distance function.

K-Values from 1 to 100 were considered, as well as Euclidean, Manhattan and Chebyshev distances. The Euclidean distance is the square root of the sum of the square difference between all input fields between two data samples. The Manhattan distance is the sum of the absolute differences between all input fields between two data samples. The Chebyshev distance is the maximum difference between all input fields between two samples. These distances were chosen as they are commonly used for KNN and in machine learning.

All unique combinations of K and distance were considered. Each unique model of K and distance was fit on the training data, and had its accuracy calculated using on the validation datasets. The model/hyperparameters with the best validation accuracy was chosen.

Neural Network

The architecture of the neural network is as follows:

- Input layer with 8 nodes (one for each feature)
- Dense hidden layer with N nodes using activation function A
- Dense Output layer of 1 node using sigmoid activation

The values of N and A were tuned hyperparameters. Values from 2 to 100 increasing by 4 (2, 6, 10 etc.) were considered for N, and the Relu, Sigmoid, and Tanh activation functions were considered for the hidden layer. Additionally, the neural network uses a cross entropy loss function and an ADAM optimizer. ADAM initializes the learning rate to 0.001, more information can be found here about the ADAM optimizer used in keras: https://keras.io/api/optimizers/adam/.

All unique combinations of N and the A were considered. Each unique model of N and A was trained for 100 epochs using stochastic gradient descent with a batch size of 10. Within training, the model was validated after every epoch on the validation dataset, and the model from the epoch with the best validation accuracy was selected from training. N and A were selected from the unique model that performed best on the validation dataset.

Logistic Regression

Logistic regression is a generalized linear model that applies linear regression to a non-linear sigmoid function. For this algorithm, the tuned hyperparameter was the learning rate, or alpha, that was used in gradient descent. Each alpha value was trained for 100 epochs using stochastic gradient descent with a batch size of 10. Within training, the model was validated after every epoch on the validation dataset, and the model from the epoch with the best validation accuracy was selected from training. The alpha value was chosen based upon the value that yielded the highest validation accuracy. Additionally, this model used a binary cross entropy loss function.

EXPERIMENT

The table below shows the different models with their best hyperparameters and the highest validation accuracy that yielded these hyperparameters.

ML Algorithm	Best Hyperparameters	Validation Accuracy
KNN	K=5, distance=chebyshev	0.784
Logistic Regression	alpha=0.3	0.771
Neural Network	N=2, activation=relu	0.758

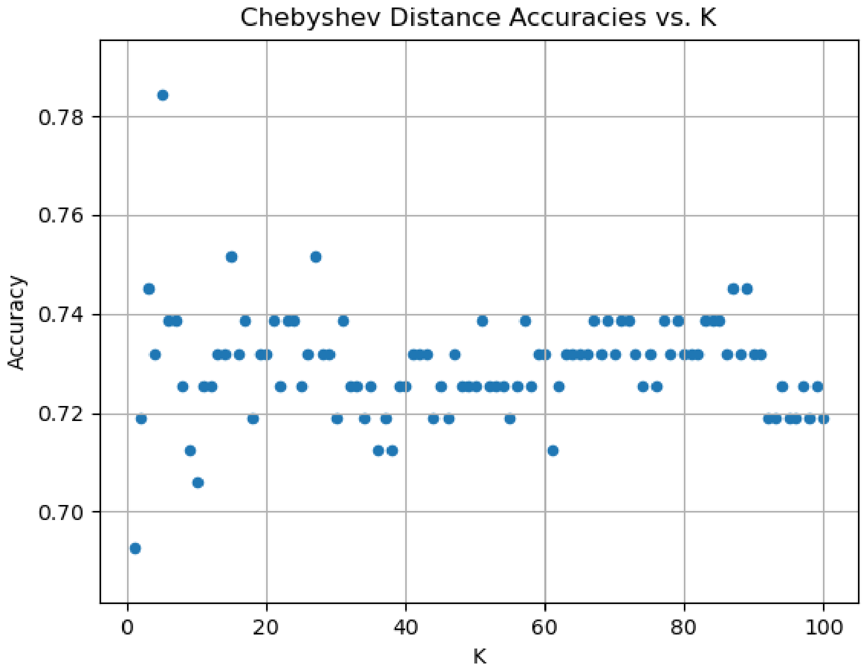
The table below shows each model using their best hyperparameters and their accuracies on the test dataset:

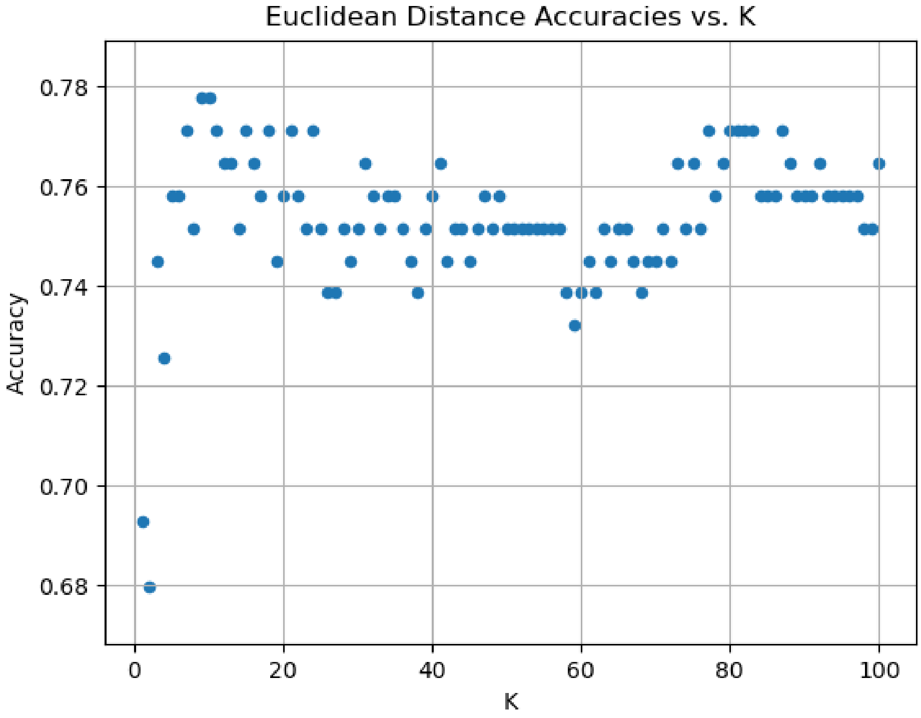
Model	Test Accuracy
Majority Guess	0.696
KNN	0.723
Logistic Regression	0.768
Neural Network	0.761

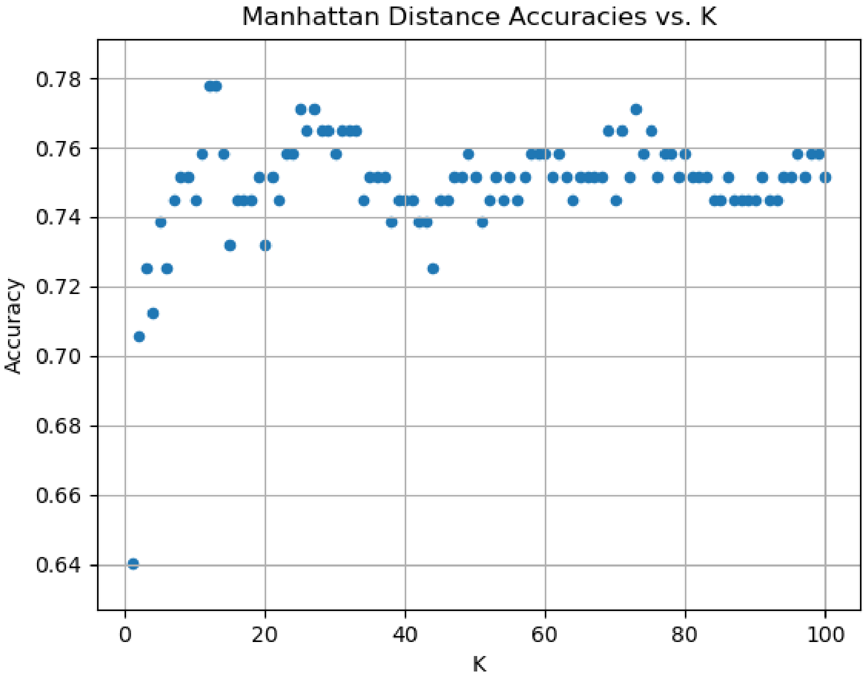
As such we can see that the logistic regression model performed best on the test dataset by a margin of roughly 7%. It is worth noting that all models perform better than the majority guess baseline, meaning that our results are likely meaningful and that our models are not just guessing. Notably, even though KNN performed best on the validation dataset, it performed worst on the test dataset. Next we will look at the visualization of some results

KNN Validation Accuracies

Since there is no gradient for KNN, there is no need to calculate the loss. Instead of graphing the loss, we graph the validation accuracies vs. K-value for the three distances used.



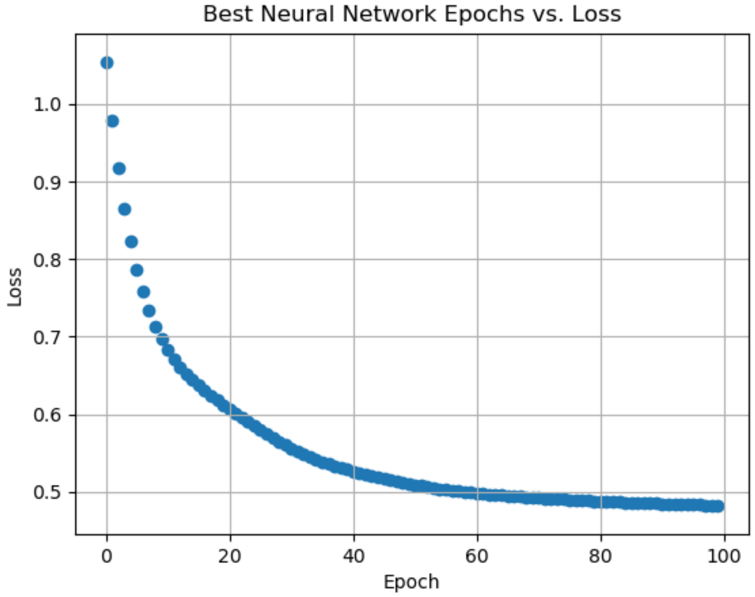




We can see for all distances that the validation accuracies fluctuate throughout all values of k. As such it is unclear whether performance has peaked. In the future a wider range of k-values can be considered.

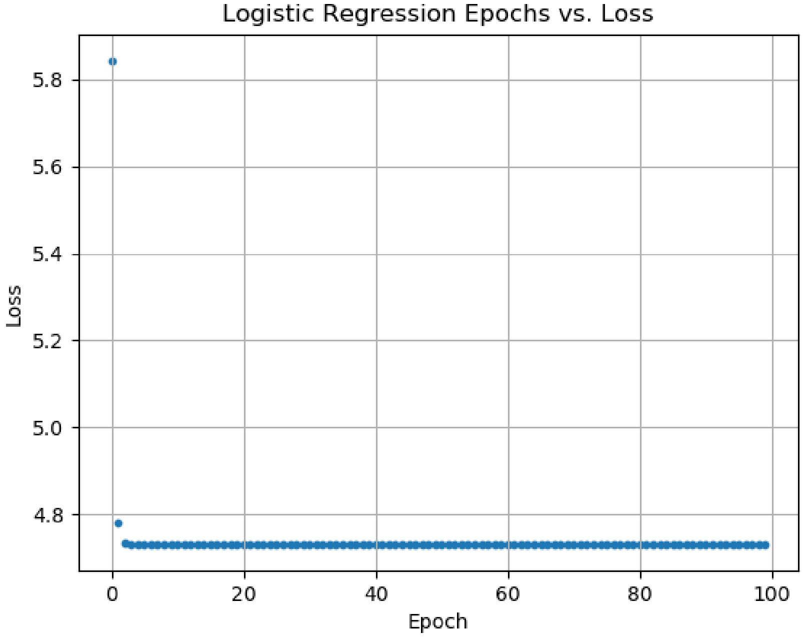
Neural Network Loss

Below is the learning curve for the best neural network model. We can see that performance has likely peaked since the loss plateaus in the latter half of the graph.



Logistic Regression Loss

Below is the learning curve for the best logistic regression model. We can see that performance remains relatively constant over all epochs, meaning that performance has likely peaked.



CONCLUSION

The task of this project was to predict whether someone had diabetes based on 8 physical attributes. We trained and tuned hyperparameters for 3 machine learning algorithms for this task, specifically K-nearest neighbours, a neural network, and logistic regression. When compared, we found that all algorithms performed better than a majority guess, meaning the results are likely meaningful and not just lucky guesses. Additionally, we found that logistic regression performed the best out of all models on the test dataset with an accuracy of roughly 77%, and the performance had likely peaked for logistic regression and the neural network after training.

REFERENCES

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

I made reference to the following websites:

- stackoverflow.com
- towardsdatascience.com
- machinelearningmastery.com
- medium.com

The code also made use of some of the default code given in coding assignments 1 and 2.