SWE 642: Extra Credit Assignment

Name: Rashika Koul

Problem description: Classify the patient as diabetic (class label 1) or not diabetic (class label 0).

Dataset: diabetes.csv (https://www.kaggle.com/saurabh00007/diabetescsv)

The data set consists of 763 records with the following attributes:

1. Number_pregnant: Number of times pregnant
2. Glucose_concentration: Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Blood_pressure: Diastolic blood pressure (mm Hg)
4. Triceps: Triceps skin fold thickness (mm)
5. Insulin: 2-Hour serum insulin (mu U/ml)
6. BMI: Body mass index (weight in kg/(height in m)^2)
7. Pedigree: Diabetes pedigree function
8. Age: Age (years)

	Number_pregnant	Glucose_concentration	Blood_pressure	Triceps	Insulin	BMI	Pedigree	Age	Class
0	6	0.743719	0.590164	0.353535	0.000000	0.500745	0.234415	50	1
1	1	0.427136	0.540984	0.292929	0.000000	0.396423	0.116567	31	0
2	8	0.919598	0.524590	0.000000	0.000000	0.347243	0.253629	32	1
3	1	0.447236	0.540984	0.232323	0.111111	0.418778	0.038002	21	0
4	0	0.688442	0.327869	0.353535	0.198582	0.642325	0.943638	33	1

Methodology:

1. Data Normalisation: To make sure that all the features/attributes in the diabetes.csv dataset are on a similar scale, mean normalisation is performed.

$$x = x - \{\min(x) / [\max(x) - \min(x)]\} \qquad \textit{where x is a feature value}$$

Note: To make feature columns work correctly with the tensorflow's estimator API, numeric column is created for every feature column.

2. Splitting training and test set: To split the dataset randomly into training and test set, sklearn library's train_test_split() method is used with test size = 0.33. Thus the number of records in the test set is 254 and that in the training set is 514.

3. Classify Using a Linear Classifier:

3.1. Train the linear classifier: The instance of the linear classifier (from the tensorflow's estimator API) is trained using the train() method for 1000 epochs and a batch size of 10 on the training set.

3.2. Predict the class label for test set: The predict() method is used on the test set to predict the class labels of the test set as either 1 (tested positive for diabetes) or 0 (tested negative for diabetes).

3.3. Evaluate performance: The evaluate() method is used to calculate the performance and accuracy of the linear classifier.

Results:

Obtained the following performance measures:

accuracy: 73.23%
accuracy_baseline: 0.65748036
auc: 0.78742516
auc_precision_recall: 0.6267114
average_loss: 0.53191453
label/mean: 0.34251967
loss: 5.196396
precision: 0.6233766
prediction/mean: 0.36624974
recall: 0.55172414
global_step: 1000