Assignment 1.6

Thursday, February 1, 2024 7:15 PM

6. I a) Code in q_6.py * dataset is randomized with set seed before execution*

Accuracy Scores: (# of correct prediction / # of total prediction)

KNN: 99.30%

Decision Tree: 99.10%

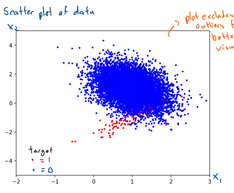
Linear Regression: 37.14%

Logistic Regression: 99.40%

As seen above, accuracy scores for all models are fairly high Except for Linear Regression, which is known for poor performance on classification

b) Because there may be a significantly more data on one class over the others. Models will not be able to accurately predict some classes due to lack of data in training set. Even during validation if most of the data belongs to the class that the model is well trained for, the overall accuracy score of the model will be high even if the model performs poorly for some classes.

c) Scatter plot of data



By plotting out all the data points, it is clear that there are much more of class "0" than class "1". There are 9899 class "0" and only 100 class "1".

d) Accuracy Score: (# of correct prediction per class / # of total prediction per class)

	Class 0	Class 1	
KNN	99.80%	37.5%	As we can see, all models
DT	99.60%	37.5%	has high accuracy for Class 0,
Linear	36.62	100%	and low accuracy for Class 1
Logistic	99.90%	37.5%	Except for Linear Regression

e) Confusion Matrix

	TP	TN	FP	FN
KNN	3	989	2	5
DT	3	987	4	5
Linear	8	363	628	0
Logistic	3	990	1	5

f) Precision and recall are important because they provide a deeper understanding of a model's performance. Especially in the case where the data set is imbalanced and there are much less True Positives than True Negatives, we can compare the number of True Positives to the number of False Positives and Negatives using Precision and Recall, which reveals how the model actually performs when classifying True Positives.

g) F1 score: $\frac{2 \times P \times R}{P + R}$

	Precision	Recall	F1
KNN	$\frac{3}{3+2} = 0.6$	$\frac{3}{3+5} = 0.375$	0.462
DT	$\frac{3}{3+4} = 0.43$	$\frac{3}{3+5} = 0.375$	0.401
Linear	$\frac{8}{8+628} = 0.013$	$\frac{8}{8+0} = 1$	0.026
Logistic	$\frac{3}{3+1} = 0.75$	$\frac{3}{3+5} = 0.375$	0.5

h) Confusion Matrix

	TP	TN	FP	FN
KNN	6	977	14	2
DT	4	987	4	4
Linear	8	261	740	0
Logistic	8	918	75	0

Accuracy Score

	Class 0	Class 1	Overall
KNN	97.59%	75%	97.40%
DT	99.60%	50%	99.20%
Linear	26.34%	100%	26.93%
Logistic	92.65%	100%	92.84%

Precision:
KNN: 0.3
DT: 0.5
Linear: 0.0103
Logistic: 0.099

Although the accuracy of Class 1 improved for all models, this precision drastically got worse. This is due to the models wrongly predicting many Class '0's to be Class '1'. The decision tree model seems to perform the best, as it has decent accuracy and precision. Due to the increase in False positives, oversampling is only ideal in the case where we are tolerant of False negatives while wanting good accuracy on the positives.

i) Confusion Matrix

	TP	TN	FP	FN
KNN	8	913	78	0
DT	9	879	112	0
Linear	8	320	671	0
Logistic	8	785	106	0

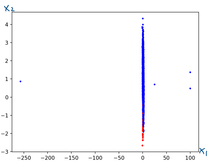
Accuracy Score

	Class 0	Class 1	Overall
KNN	92.13%	100%	92.20%
DT	88.70%	100%	88.74%
Linear	32.3%	100%	32.83%
Logistic	92.93%	100%	92.50%

Precision:
KNN: 0.093
DT: 0.067
Linear: 0.0113
Logistic: 0.0702

None of the classifiers performed well with underfitting as they all have bad precision. KNN performed the best out of all the models with high accuracy and precision higher than other models.
Underfitting in this case would only be ideal if you want high accuracy on the positives while heavily tolerant of false positives. If is also good if you don't want to falsely classify the negatives as none of the models have any false negatives

II a)



Using the scatter plot from part 1.c, we can see that feature x_1 has outliers.
A good approach to dealing with them would just be removing them from the dataset

b) Confusion Matrix

k	TP	TN	FP	FN
1	3	988	5	5
3	3	989	2	5
5	3	989	2	5

It seems like around k=3 is when outliers stop being a problem. They would only be a problem with small k values, as points with features close to outlier features may get classified wrong. As the number of neighbours increase, the impact of outliers decrease as now the majority of neighbours will be of the correct classification

c)

depth	TP	TN	FP	FN
1	3	989	2	5
2	3	990	1	5
3	3	990	1	5

It seems like around max_depth = 2, is when outliers stop being a problem. However, there does not seem to that much of an effect caused by outliers as the number of false positives was small initially and only decreased by 1 as max_depth increased.
In general, since Decision trees make decisions based on splitting features, they are less impacted by a single data point

d) KNN are less imperious to outliers, especially if the number of neighbours is small. This is because if a given point is close to outliers, the outliers might skew the majority vote of the neighbours, resulting in false classification. KNNs are affected by all datapoints, while Decision Trees are only affected by the general patterns of features, making them less affected by outliers

III Some issues:

Feature scaling: Features must be normalized to be on same scale. Some models are sensitive to the magnitude of the feature.

Irrelevant Features: Some feature are irrelevant (no correlation) to the prediction. Leaving them in to train the model might lead to some inaccuracies.

Missing Data: Some data are not easily attainable (ie, difficult experiment, loss of data). Need to either work with small dataset or simulate data.