Identification of Cancer Essential Genes using GAMs

Abstract

In machine learning often a tradeoff must be made between accuracy and intelligibility. More accurate models such as boosted trees, random forests, and neural nets usually are not intelligible, but more intelligible models such as logistic regression, naive-Bayes, and single decision trees often have significantly worse accuracy.

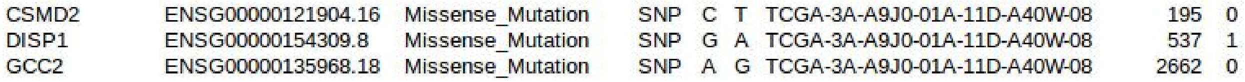
XAI has various applications in the medical industry as if one questions ‘why this gene has been classified as essential, how are you sure that removal of this gene will mitigate cancer cells?’, the availability of an explanation becomes valuable and the users have a higher chance of accepting the answer given by the model.

Also if we have the answer to the question, “based on what features, does the model predict if a gene is essential for a cancer cell”, we will have a direction for the future research on the same. Considering only the features which significantly affect the health of a cancer cell. It can also be used to provide some counterexamples on previously thought important factors affecting the health of the cell.

Classes

The dataset comprised of two classes, whether the gene is essential or not. Our task is a binary classification problem.

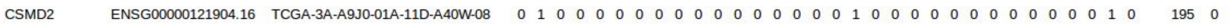
Dataset and PreProcessing



CSMD2	ENSG00000121904.16	Missense_Mutation	SNP	C	T	TCGA-3A-A9J0-01A-11D-A40W-08	195	0
DISP1	ENSG00000154309.8	Missense_Mutation	SNP	G	A	TCGA-3A-A9J0-01A-11D-A40W-08	537	1
GCC2	ENSG00000135968.18	Missense_Mutation	SNP	A	G	TCGA-3A-A9J0-01A-11D-A40W-08	2662	0

The original dataset is shown above, the first two columns being the gene name and gene ID, the third column is the type of mutation the gene underwent, the fourth column shows a substitution mutation. The fifth column contains the name of the nucleotide which was replaced and the sixth column shows the new nucleotide. The seventh column is the patient ID. The eighth column is the Expression Value, and the ninth being the copy number.

The data was one hot encoded to the following format



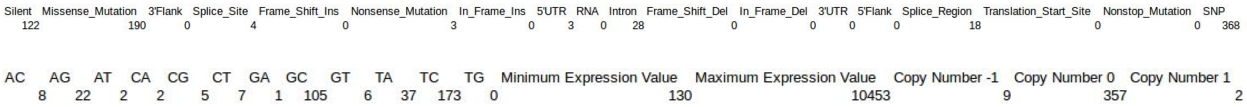
CSMD2 ENSG00000121904.16 TCGA-3A-A9J0-01A-11D-A40W-08 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 195 0

There are 17 types of mutations possible, 12 combinations of nucleotide change and three modes of mutation(SNP, DEL, INS) though this dataset only has Substitution(SNP) mutations.

The data is then compressed by adding all the values of a particular gene to find out the total number of mutations of each type it has undergone.

To find out the range of the expression value, min and max of the expression value of the particular gene are stored.

The data now looks like



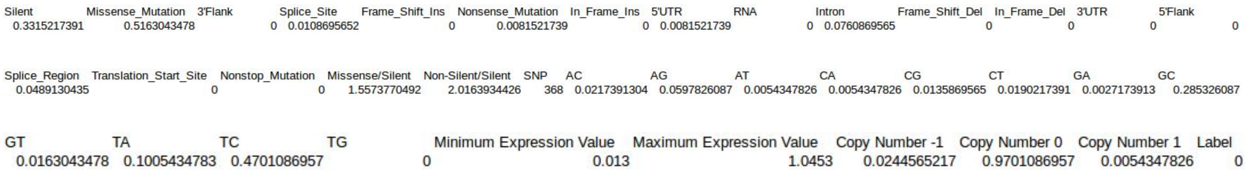
Silent	Missense_Mutation	3'Flank	Splice_Site	Frame_Shift_Ins	Nonsense_Mutation	In_Frame_Ins	5'UTR	RNA	Intron	Frame_Shift_Del	In_Frame_Del	3'UTR	5'Flank	Splice_Region	Translation_Start_Site	Nonstop_Mutation	SNP
122	190	0	4	0	3	0	3	0	28	0	0	0	0	18	0	0	368

AC	AG	AT	CA	CG	CT	GA	GC	GT	TA	TC	TG	Minimum Expression Value	Maximum Expression Value	Copy Number -1	Copy Number 0	Copy Number 1
8	22	2	2	5	7	1	105	6	37	173	0	130	10453	9	357	2

Each row corresponding to a gene.

Now to get the fraction of each type of mutations, it was divided by the total number of mutations for a gene, and the expression values were brought to a scale [0,10]

Now our final data after preprocessing is



Silent	Missense_Mutation	3'Flank	Splice_Site	Frame_Shift_Ins	Nonsense_Mutation	In_Frame_Ins	5'UTR	RNA	Intron	Frame_Shift_Del	In_Frame_Del	3'UTR	5'Flank
0.3315217391	0.5163043478	0	0.0108695652	0	0.0081521739	0	0.0081521739	0	0.0760869565	0	0	0	0

Splice_Region	Translation_Start_Site	Nonstop_Mutation	Missense/Silent	Non-Silent/Silent	SNP	AC	AG	AT	CA	CG	CT	GA	GC
0.0489130435	0	0	1.5573770492	2.0163934426	368	0.0217391304	0.0597826087	0.0054347826	0.0054347826	0.0135869565	0.0190217391	0.0027173913	0.285326087

GT	TA	TC	TG	Minimum Expression Value	Maximum Expression Value	Copy Number -1	Copy Number 0	Copy Number 1	Label
0.0163043478	0.1005434783	0.4701086957	0	0.013	1.0453	0.0244565217	0.9701086957	0.0054347826	0

The dataset is divided, in the following parts

The input feature contains-

The frequency of 17 types of mutations that can happen to the gene (eg Missense Mutation, Silent).

The frequency of the type of Substitution happening for the particular gene (eg A->G, C->T).

The minimum and the maximum expression value the gene exhibits across the different mutations.

The frequency of the Copy Number the gene exhibits across different mutations.

Training Procedure

A Logistic GAM model is trained on the feature space of 33 (Dimension reduced from 37 to 33 as some columns had only 0 in each row). GAM has a higher variance than Linear Regression as it allows the non-linear contribution of the features.

g(E(y|X)) = f1(X1) + f2(X2) ..... + fn(Xn)

In the case of Linear regression, fi and g are Identity functions. g is known as the link function.

There is a link function which is chosen as logit, as it is the standard used for logistic regression.

The train:test split is 60:40

Lam is a hyperparameter which varies the extent of regularization and helps to smoothen the function.

Deciding on the hyperparameters:

There are three main hyperparameters to a GAM model, namely lam, higher lam means higher regularisation and smoothening.

The next parameter is no_splines - It represents the number of splines used to create a function for a parameter.

Constraint - Constraint on the shape of the function.

To decide on getting good value of lam for the model, randomly 10 samples of lam were generated between (exp(-3),exp(3)).

No_splines were varied from 5 to 25.

And constraints were chosen from ['convex', 'concave', 'monotonic_inc', 'monotonic_dec','circular', 'none']

Firstly a grid search was done on no_of splines and lam. The best values were obtained from within the hyperparameters tested.

To save on training time of the GAM, the best constraint was searched for in the next step.

Though searching one after another, and not in the same grid search does not guarantee the best fit among all the combinations of the hyperparameters tested.

It saves on training time and provides a pretty good model as well.

The model with the lowest generalized cross-validation score was considered.

The best lamda values are stated below

[[2.0673611019658185], [1.6550137600896615], [12.104084551199719], [0.48236647723396564], [5.115348389792072], [4.456248763299449], [1.5915248490085596], [0.15640512606843154], [1.6769091678223782], [0.10049025136639023], [3.8926649180569544], [0.09509410374760731], [17.231351990978144], [0.21014185364838445], [5.1741677072860055], [0.7080171857515718], [0.46969785002364106], [0.8385598584081343], [0.20735831235233362], [0.05443705567702603], [5.740018358490613], [14.030656219035649], [0.06277075205253789], [12.912326562913087], [0.21318907544569524], [13.557472712965694], [0.08886022376964244], [0.07027220257600182], [0.06926993974769438], [0.13079151380991574], [0.6504762625196049], [0.7811214639815901], [3.3485558319413253]]

Below is the best no_spline

[20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20]

And the best constaraint was concave

The partial dependence plot was plotted for the various features to get an idea about the variation of the output with the feature.

Results

Accuracy = 78%

Confusion matrix =

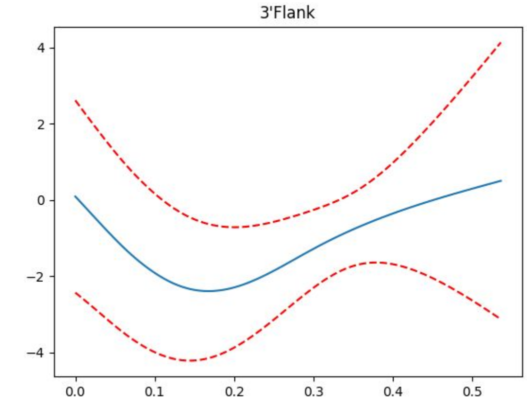
[[9273 2211]

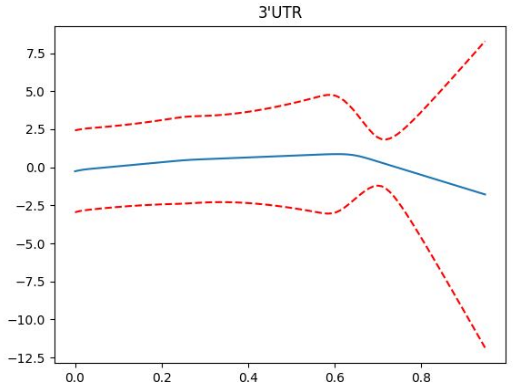
[723 1155]]

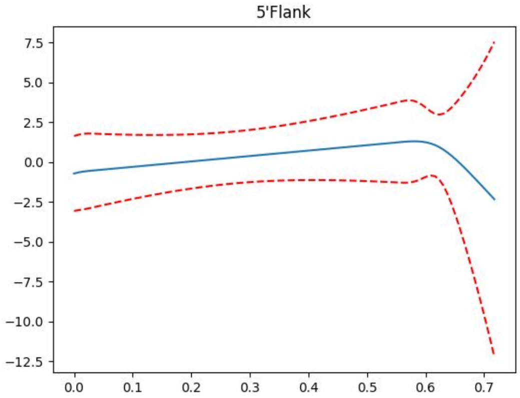
To evaluate how the different features affect the model, we plot the Partial Dependence Plots.

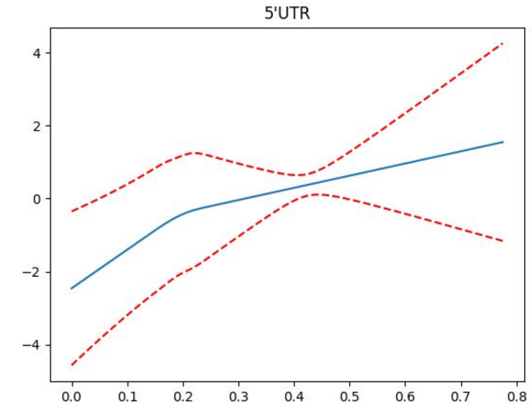
Partial Dependence Plot

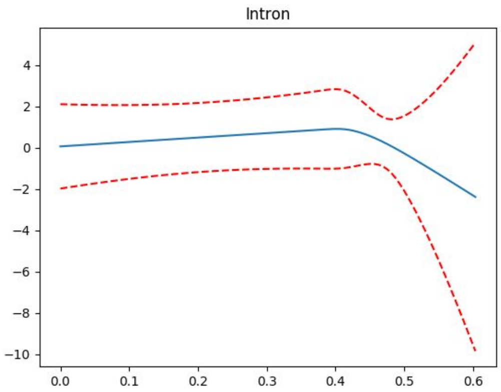
1) Mutation Types

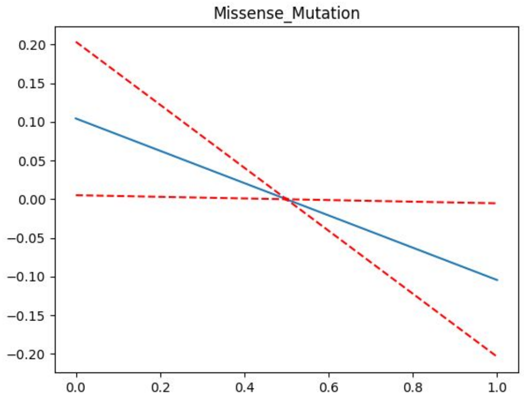


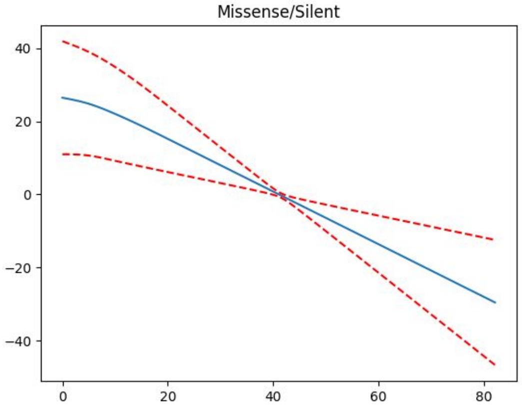


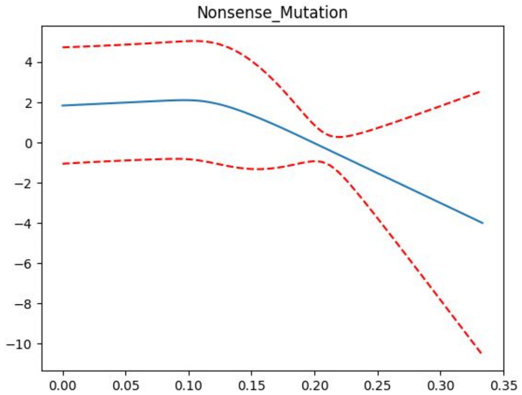


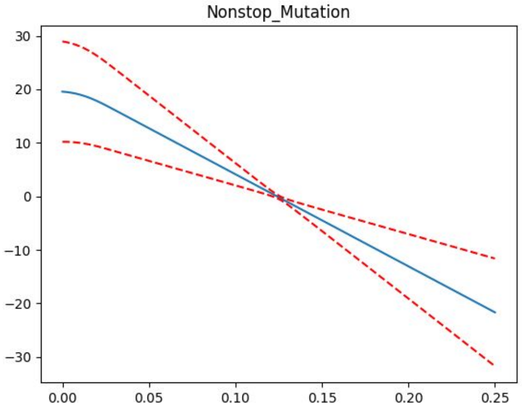


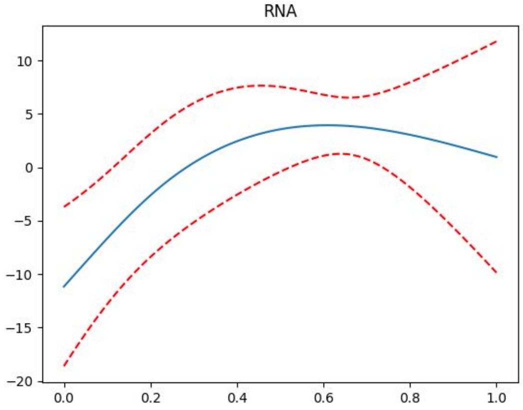


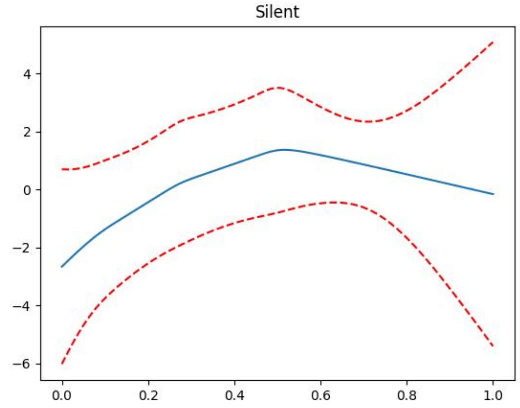


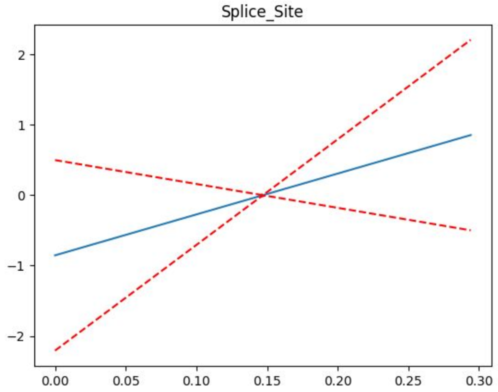

Missense/Silent
40
20
0
−20
−40
0
20
40
60
80

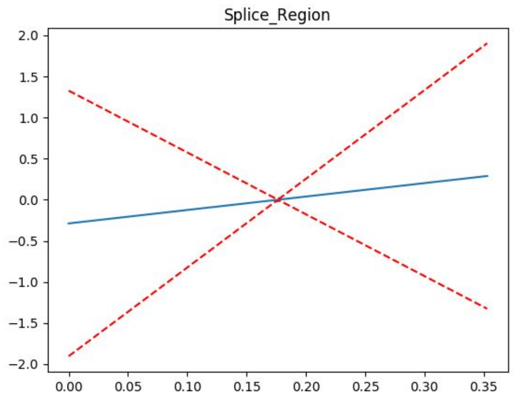

Nonsense_Mutation
4
2
0
−2
−4
−6
−8
−10
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35

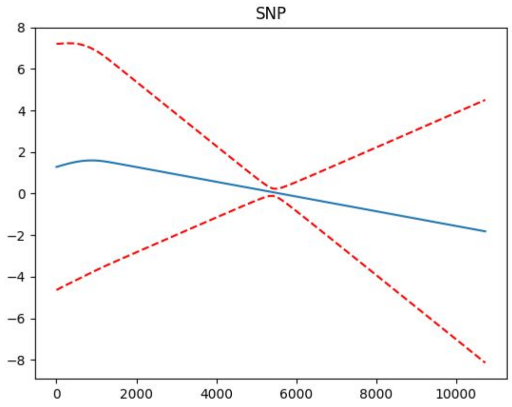

Nonstop_Mutation
30
20
10
0
−10
−20
−30
0.00
0.05
0.10
0.15
0.20
0.25

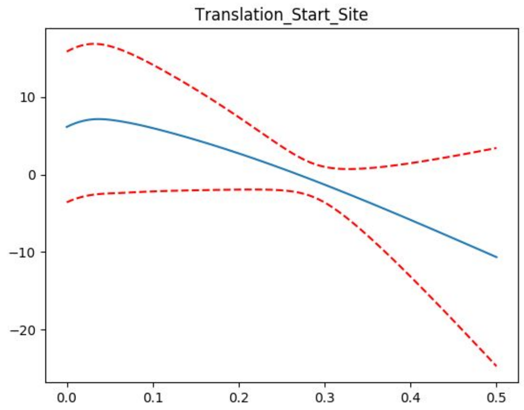

RNA
10
5
0
−5
−10
−15
−20
0.0
0.2
0.4
0.6
0.8
1.0


Silent
4
2
0
−2
−4
−6
0.0
0.2
0.4
0.6
0.8
1.0


Splice_Site
2
1
0
−1
−2
0.00
0.05
0.10
0.15
0.20
0.25
0.30







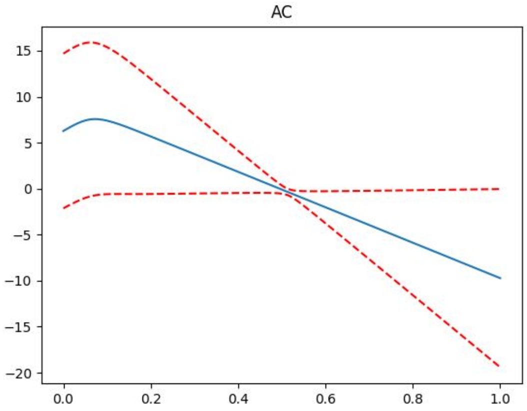
The blue line represents the expected value of g(y) and the red lines represent the boundary of 95% confidence. I:e g(y) will lie in this range with a probability of 95%.

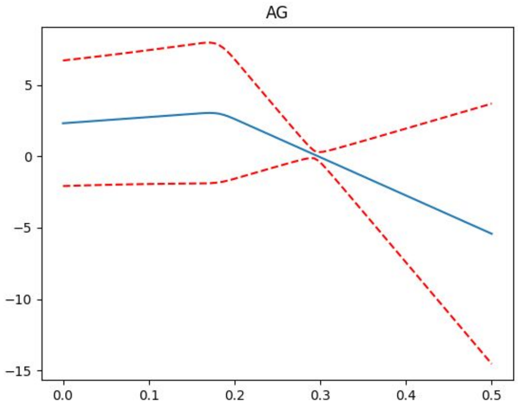
In the case of MIssense/Silent mutation ratio, we observe that the probability of the gene being classified as essential increases this makes sense as the mutation of an essential gene would generally not lead to a silent mutation.

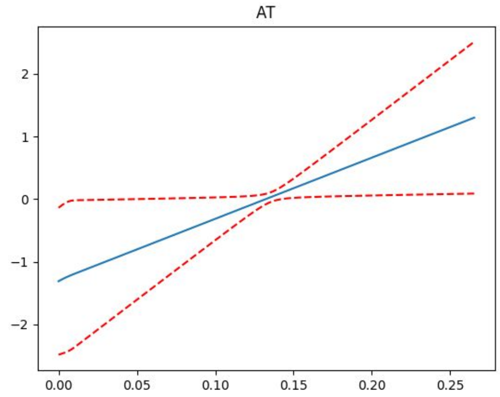
SNP (Substitution Mutations) represents the total number of mutations of the gene in the dataset, as SNP increases it is expected that the gene is not essential as we don't expect an essential gene to undergo mutation as it is required by the cell for it's functioning.

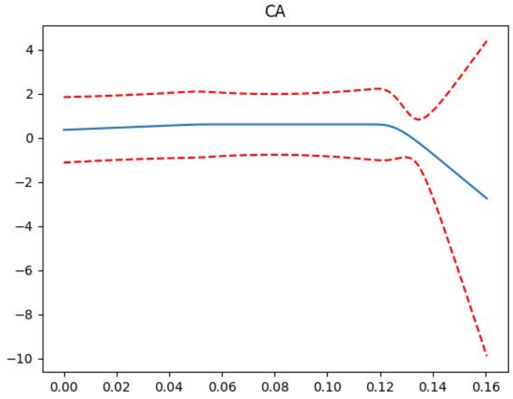
2) Nucleotide Change

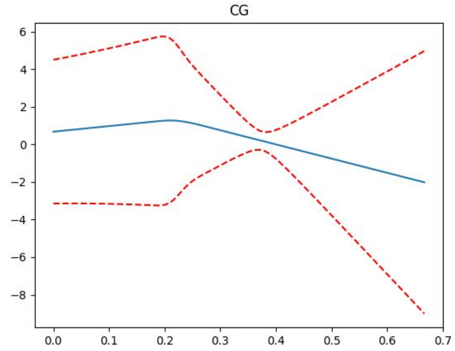
The following plots represent the variation of the type of nucleotide change with the output. Here, for example, AT represents A nucleotide is replaced by T nucleotide.

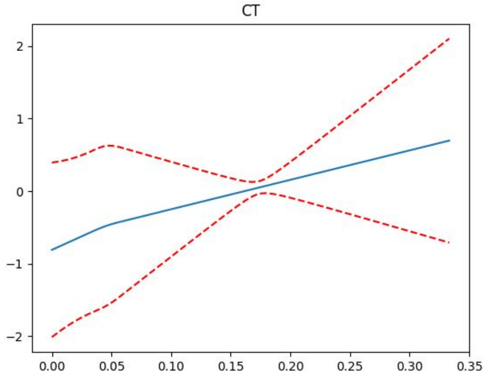

AC
15
10
5
0
−5
−10
−15
−20
0.0
0.2
0.4
0.6
0.8
1.0

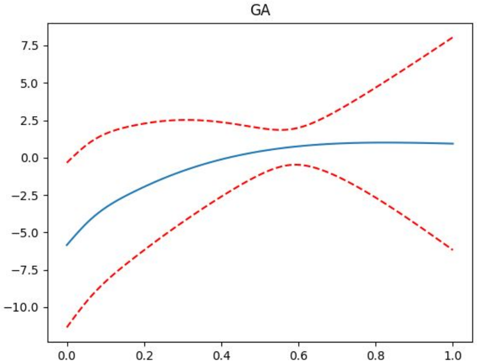

AG
5
0
−5
−10
−15
0.0
0.1
0.2
0.3
0.4
0.5

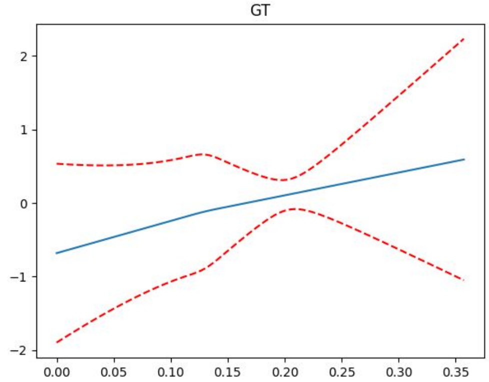

AT
2
1
0
−1
−2
0.00
0.05
0.10
0.15
0.20
0.25

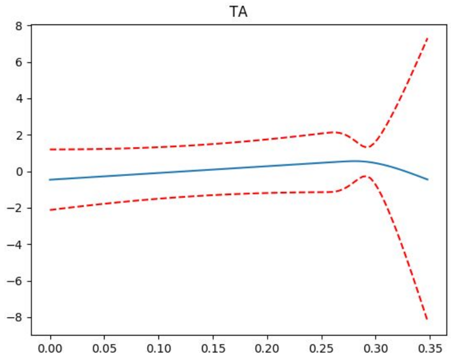

CA
4
2
0
−2
−4
−6
−8
−10
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16

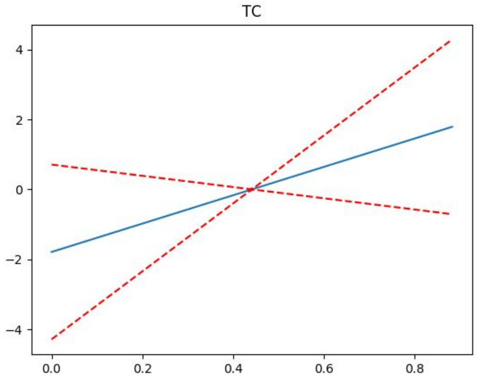

CG
6
4
2
0
−2
−4
−6
−8
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

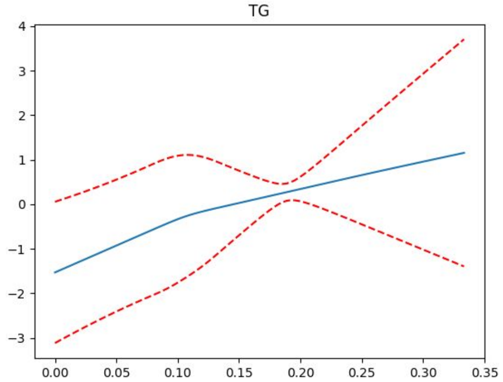

CT
2
1
0
−1
−2
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35


GA
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
0.0
0.2
0.4
0.6
0.8
1.0


GT
2
1
0
−1
−2
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35



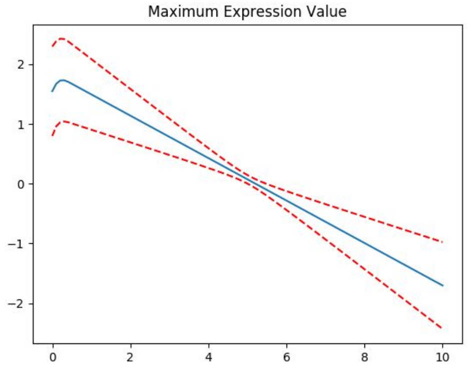


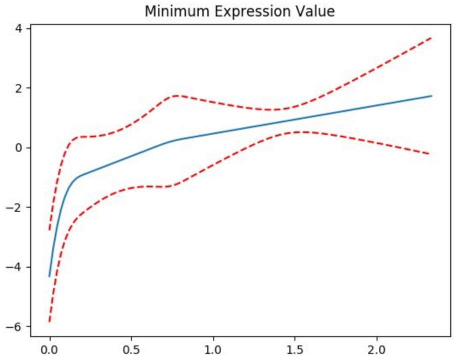


The plots indicate that if it is more likely that a gene will undergo a T->C, T->G or C->T substitution, it has a higher expectation of being an essential gene.

While on the other hand if the substitution of type A->C or A->G has a higher chance of taking place, the gene is more likely to be a non-essential gene.

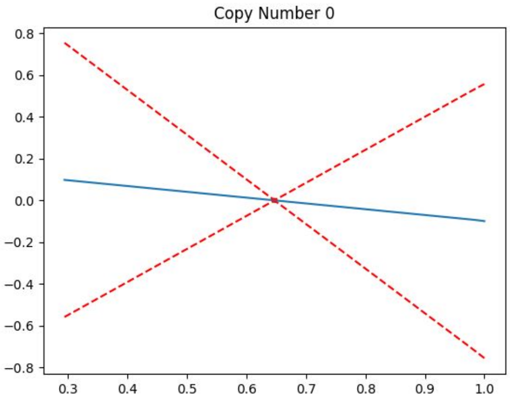
3) Expression Value

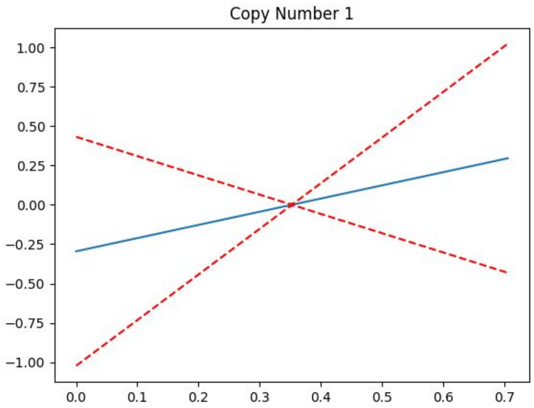


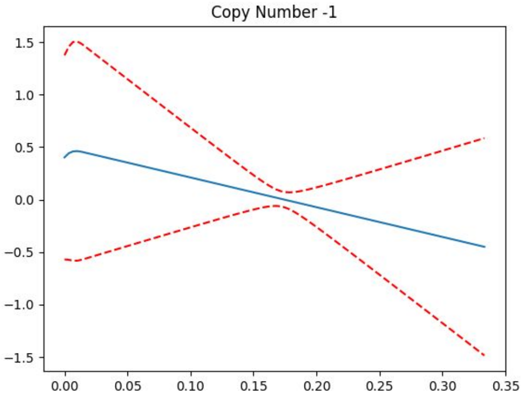


Here we observe that at a lower value of the gene expression data, it is more likely for a gene to be non-essential as it is expected from an essential gene to have a higher expression value as it is a representative of how active is the gene in the cell. Higher maximum expression value indicates a lower probability of the gene being essential.

4) Copy Number fraction







As the copy number increases, the probability of the gene being an essential gene increases

Conclusion

The model seems utilised GAMs to help us discover previously unknown relations between certain features and the probability of a gene being essential, for example, a gene which has a higher probability of undergoing an A->C transition is more likely to be a non-essential gene. It also shows some expected trends like increase in the Minimum Expression value of a gene means a higher probability of the gene is an essential gene.

So, in conclusion, though models with higher variance like Neural Nets might provide better accuracy to study the trends of the effect of different features more transparent models like GAM work well with decent correctness.