Predicting Loss-of-Function Impact of Genetic Mutations: A Machine Learning Approach

Arshmeet Kaur, Dr. Morteza S.

***Abstract*— The innovation of next-generation sequencing (NGS) techniques has significantly reduced the price of genome sequencing, lowering barriers to future medical research; it is now feasible to apply genome sequencing to studies where it would have previously been cost-inefficient. Identifying damaging or pathogenic mutations in vast amounts of complex, high-dimensional genome sequencing data may be of particular interest to researchers. Thus, this paper's aims were to train machine learning models on the attributes of a genetic mutation to predict LoFtool scores (which measure a gene's intolerance to loss-of-function mutations). These attributes included, but were not limited to, the position of a mutation on a chromosome, changes in amino acids, and changes in codons caused by the mutation. Models were built using the univariate feature selection technique f-regression combined with K-nearest neighbors (KNN), Support Vector Machine (SVM), Random Sample Consensus (RANSAC), Decision Trees, Random Forest, and Extreme Gradient Boosting (XGBoost). These models were evaluated using five-fold cross-validated averages of r-squared, mean squared error, root mean squared error, mean absolute error, and explained variance. The findings of this study include the training of multiple models with testing set r-squared values of 0.97.**

***Index Terms*— Machine Learning, Prediction Algorithms, Supervised Learning, Support vector machines, K-Nearest Neighbors, RANSAC, Decision Trees, Random Forest, Genetic mutations, LoFtool, Next Generation Sequencing**

I. Introduction

LAST year, Ultima Genomics announced that it could sequence a human genome for just one hundred dollars per person [1]. The reduced cost of genome sequencing means it may now be possible for research in the medical field to collect "omics" data (i.e., genomics, epigenomics, transcriptomics, epitranscriptomics, proteomics, and metabolomics) where it otherwise would have been too expensive to do so. With the generation of potentially vast amounts of data comes the need to develop informatics tools capable of handling and analyzing it. Machine learning and deep learning pose a solution [2]. Training machine-learning tools that can identify pathogenic variants in a genome sequence is potentially useful to researchers; previous research in the field of prediction of genetic pathogenicity has been focused on developing deep/machine learning models to predict mutations' functional effects. For example, methods like FATHMM-MKL and CADD are designed to predict functional consequences of coding and non-coding variants [3]. MetaRNN (developed in [4]) is a deep learning method that distinguishes between benign and pathogenic rare mutations. Other research has focused on datasets of a specific disease, such as PathoPredictor, an ensemble method made for cardiomyopathy, epilepsy, or RASopathies [5]. Some studies test the generalizability of models by using existing methods on clinical data [6].

This paper's aim was to train machine learning models to predict LoFtool scores. To create the LoFtool gene score, researchers retrieved all high-confident Loss-of-function mutations (defined as those that disrupt protein structure [7]) from Fadista et al.'s 60,706 record Exome Aggregation Consortium dataset [8]. LoFtool provides a score that quantifies how intolerant a certain gene is to loss-of-function variants– in other words, how susceptible a gene is to disease if mutated. It ranks the percentile of intolerance. LoFtool differs from pathogenicity scores such as PolyPhen, SIFT, ENDEAVOR, or Prioritizer because it can extrapolate its measurements to the gene level instead of focusing on a single variant's pathogenicity. It is also possible to calculate the score without prior knowledge of the disease a gene is associated with. The LoFtool score has been used in research for *in silico* experiments. For example, it was used to analyze the pathogenicity of the human SOD1 gene, specifically to get a score for an important non-coding Indel [9]. Or, as done in [10], LoFtool can be used to identify the most variant-intolerant genes or novel genes in a polygenic disease such as Type 2 diabetes. The contribution of our trained machine learning models to get LoFtool scores in a few seconds with high accuracy based on genetic attributes such as chromosome, strand type, gene, feature, exon number, and codon change could be useful to researchers.

II. Methods

A. Original Dataset

In this study, an open-source, public-domain dataset published in 2020 was used [11]. The original dataset, created from ClinVar data, contained genetic mutations from 23 chromosomes (X but not Y chromosome included) and 46 variables quantifying various attributes of the mutation, such as chromosome location or allelic frequency in the general population. To understand the original data in more detail,

Date of submission: Aug 23, 2023

Arshmeet Kaur is with Evergreen Valley College, 3095 Yerba Buena Rd, San Jose, CA 95135(e-mail: Arka7783@stu.evc.edu).

Dr. Morteza Sarmadi is an RD scientists at SiO2 Materials Sciences: 2250 Riley St, Auburn, AL 36832 (e-mail: mortezanear@yahoo.com).

please consult the data card in [11]. To determine whether a variant is classified as pathogenic or benign, geneticists performed manual classification at labs, sorting variants into one of three categories: 1) benign or likely benign, 2) VUS (uncertain or conflicting pathogenicity [12]) or 3) likely pathogenic or pathogenic. If different geneticists at different laboratories assigned different classifications, then CLASS = 1, and otherwise CLASS = 0. The original dataset was created so users could create classification models to predict the CLASS variable. However, in order to use the dataset to train models and predict pathogenicity scores, all rows where CLASS = 1 were deleted and the CLASS variable was dropped, eliminating all conflicting information on pathogenicity.

B. Data Preprocessing

High-dimensional data poses challenges to statistical methods. Oftentimes, high-dimensional data contains redundant information [13]. Thus, the first step of data cleaning was to drop all irrelevant and/or redundant variables, those with very sparse data, and those with very low variance (Table I). These were the final predictor variables: CHROM, POS, REF, ALT, AF_ESP, AF_EXAC, AF_TGP, MC, IMPACT, SYMBOL, Feature, EXON, cDNA_position, CDS_position, Protein_position, Amino_acids, Codons, and STRAND. Several columns (cDNA_position, CDS_position, and Protein_position) contained asterisks, question marks, and dashes in several entries, so these entries were all dropped.

(Insert Table I)

C. Addressing Missing Values and Encoding Categorical Variables

All missing values for the target variable, LoFtool, were dropped (6.23 percent of the data). This incidentally also dropped all null values from other variables. To identify whether the dropping of null values caused low variance in any variables, distributions of all continuous and categorical variables were compared before and after data preprocessing and dropping missing values (see Supplementary Figures 1 and 2). Fortunately, no variables developed low variance and distributions stayed nearly identical. The final dataset contained 37220 entries with 19 variables.

Most of the categorical variables in the dataset were nominal, high-cardinality variables (variables with many possible categorical values). For example, SYMBOL and EXON had over two-thousand unique categories. Because of this, regularized target encoding, which has been shown to outperform other methods of encoding such as leaf, integer, and one-hot or dummy encoding for high-cardinality features [14], was used.

D. Visualizing Relationships Within in the Final Dataset

At this point, relationships between variables were explored, specifically correlation between different predictors (Fig. 1). Studies focusing on machine learning algorithms in genomics have shown that correlations between predictor variables in feature sets should be considered [15]. A Pearson's correlation coefficient above 0.20 is typically considered a weak correlation, above 0.40 is a moderate correlation, and anything over 0.60 is considered a strong correlation [16].

(insert Fig. 1.)

E. Visualizing Skew and Transforming Data

All continuous variables except for LoFtool (AF_ESP, AF_EXAC, AG_TGP, cDNA_position, CDS_position, Protein_position) were heavily right-skewed, which was considered when developing machine learning models [17]. These variables had to be transformed in further data preprocessing. The final, encoded dataset still contained heavily right-skewed variables with dramatic outliers (see Fig. 2): AF_EXAC, AF_ESP, AF_TGP, cDNA_position, CDS_position, and Protein_position variables. The presence of specialized outlier-robust machine learning models such as RANSAC [18], [19] suggests that traditional machine learning models may be thwarted by large proportions of outliers like those present in the cleaned data. RANSAC's key feature is that it is robust to a large amount of outliers in input data. Unlike other algorithms built for the same function, it works by using the smallest amount of entries possible from a dataset and slowly grows the number of entries. Additionally, logarithm and Yeo-Johnson transformations were used to create two new datasets. Logarithm transforming works by putting heavily skewed data on a log scale, which leads to a more normal distribution [20]; however, its validity in biomedical research and data analysis has been questioned [21], [22] and it has been pointed out that it is unique and only applicable for certain cases [23]. Because the validity of the Logarithm transformation has been questioned, I decided to create one dataset that was Yeo-Johnson transformed. The Yeo-Johnson transform is similar to the family of Box-Cox transformations, but it is able to handle negative entries [24], [25]. Even after the transformation, many of the variables still contained significant outliers (Fig. 2).

(Insert Fig. 2)

F. Feature Selection

With the finalized datasets, the next step was feature selection; redundant and low-variance features had already been manually filtered out in data cleaning, but selecting sets of relevant features trains the simplest possible model and helps avoid overfitting. Univariate feature selection techniques are quick, efficient, and good for high-dimensional datasets. In bioinformatics research, one would expect that univariate feature selection would be inferior to other types; however, in practice, univariate methods can yield better results (though it is important to note that researchers have explained this as being a result of limited sample size) [26]. To carry out univariate feature selection, scikit-learn's feature selection module's SelectKBest function was used, which chooses the top k features (k =10 in this case) in each dataset [27]. Since LoFtool was a continuous target variable, and the problem was a regression problem, f-regression was used to select ten out of eighteen variables for use. For all data (df_loftool.csv, df_loftool_log.csv, df_loftool_yj.csv), these features were selected: ['POS', 'cDNA_position', 'CDS_position',

'Protein_position', 'STRAND', 'CHROM', 'SYMBOL', 'Feature', 'EXON', 'Codons']. Seeing the strong correlations between some of these features, some were taken out manually and important findings were added to Tables II, III, and IV.

G. Model Selection

To predict LoFtool, K Nearest Neighbors (KNN) Regressor, Support Vector Regressor (a type of SVM abbreviated SVR), Decision Trees, Random Forest Regressor, Extreme Gradient Boost (XGB), and RANSAC were used. To evaluate the performance of the models used, k-fold cross-validation was used, as this method can test generalization and control overfitting of machine learning models [28]. Performance metrics that were calculated included averaged r-squared, mean squared error, root mean squared error, mean absolute error, and explained variance.

III. Results and Discussion

All of the models were trained and tested on the datasets created. Tables II, III, and IV contain the averaged Cross-Validation scores for k-fold (k=5) cross-validation. Random Forest and XGBoost Regressors performed the best in all three LoFtool datasets, with KNN and Decision Tree Regressor in close second. It did not seem to make a difference if the dataset was transformed or not, as several models achieved an r-squared value of 0.97 regardless of transformation. However, notably, when cDNA_position, CDS_position, and Protein_position, which had significant outliers, and were also very highly correlated with each other, (Figs 1 and 2) were removed along with POS, models tended to perform much better. For example, as seen in Table II, KNN had an r-squared value of 0.44 and 0.95 before and after removal of cDNA_position, CDS_position, Protein_position, and POS and SVR went from an r-squared of -0.32 to 0.92. RANSAC, which is robust to outliers, did not perform the best in any of the datasets. This study shows the potential use of machine learning in analysis of genetic mutations and trains a tool potentially useful to researchers in the fields of sequence analysis and pathogenicity prediction. Future research could further explore how varying data distributions and feature selection techniques affect the performance of models, or test generalizability of the model with larger datasets.

(Insert Tables II,III,IV)

References

[1] (2022) A $100 Genome? New DNA Sequencers Could Be a 'Game Changer' for Biology, Medicine. Science.org. [Online]. Available: https://www.science.org/content/article/100-genome-new-dna-sequencers-could-be-game-changer-biology-medicine

[2] C. Caudai, A. Galizia, F. Geraci, L. Le Pera, V. Morea, E. Salerno, A. Via, and T. Colombo, "Ai applications in functional genomics," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 5762–5790, October, 2021, Accessed on: August, 16, 2023, doi: 10.1016/j.csbj.2021.10.009, [Online].

[3] H. A. Shihab, M. F. Rogers, J. Gough, M. Mort, D. N. Cooper, I. N. Day, T. R. Gaunt, and C. Campbell, "An integrative approach to predicting the functional effects of non-coding and coding sequence variation," *Bioinformatics*, vol. 31, no. 10, pp. 1536–1543, February, 2015, Accessed on: August, 16, 2023, doi: 10.1093/bioinformatics/btv009, [Online].

[4] C. Li, D. Zhi, K. Wang, and X. Liu, "Metarnn: differentiating rare pathogenic and rare benign missense snvs and indels using deep learning," *Genome Medicine*, vol. 14, no. 115, October 2022, Accessed on: August, 16, 2023, doi: 10.1186/s13073-022-01120-z, [Online].

[5] P. Evans, C. Wu, A. Lindy, D. A. McKnight, M. Lebo, M. Sarmady, and A. N. Abou Tayoun, "Genetic variant pathogenicity prediction trained using disease-specific clinical sequencing data sets," *Genome Research*, vol. 29, no. 7, pp. 1144–1151, July, 2019, Accessed on: August, 16, 2023, doi: 10.1101/gr.240994.118, [Online].

[6] A. C. Gunning, V. Fryer, J. Fasham, A. H. Crosby, S. Ellard, E. L. Baple, and C. F. Wright, "Assessing performance of pathogenicity predictors using clinically relevant variant datasets," *Journal of medical genetics*, vol. 58, no. 8, pp. 547–555, August, 2021, Accessed on: August, 16, 2023, doi: 10.1136/jmedgenet-2020-107003, [Online].

[7] L. Gerasimavicius, B. J. Livesey, and J. A. Marsh, "Loss-of-function, gain-of-function and dominant-negative mutations have profoundly different effects on protein structure," *Nature communications*, vol. 13, no. 3895, July, 2022, Accessed on: August, 16, 2023, doi: 10.1038/s41467-022-31686-6, [Online].

[8] J. Fadista, N. Oskolkov, O. Hansson, and L. Groop, "Loftool: a gene intolerance score based on loss-of-function variants in 60 706 individuals," *Bioinformatics*, vol. 33, no. 4, pp. 471–474, August, 2016, Accessed on: August, 16, 2023, doi: 10.1093/bioinformatics/btv602, [Online].

[9] P. Tripathi, S. Agarwal, A. N. Sarangi, S. Tewari, and K. Mandal, "Genetic variation in sod1 gene promoter ins/del and its influence on oxidative stress in beta thalassemia major patients," *International Journal of Hematology-Oncology and Stem Cell Research*, vol. 14, no. 2, pp. 110–117, April, 2020, Accessed on: August, 16, 2023.

[10] J. Taneera, S. Dhaiban, A. K. Mohammed, D. Mukhopadhyay, H. Aljaibeji, N. Sulaiman, J. Fadista, and A. Salehi, "Gnas gene is an important regulator of insulin secretory capacity in pancreatic β-cells," *Gene*, vol. 715, p. 144028, July, 2019, Accessed on: August, 16, 2023, doi: 10.1016/j.gene.2019.144028, [Online].

[11] K. Arvai, 2020. Genetic variant classifications, distributed by Kaggle, doi: 10.34740/KAGGLE/DSV/1030915.

[12] (2023) NCI Dictionary of Genetics Terms. National Cancer Institute. [Online]. Available: https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/vus

[13] C. O. S. Sorzano, J. Vargas, and A. P. Montano, "A survey of dimensionality reduction techniques," March, 2014.

[14] F. Pargent, F. Pfisterer, J. Thomas, and B. Bischl, "Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features," *Computational Statistics*, vol. 37, no. 5, pp. 2671–2692, March, 2022, Accessed on: August, 16, 2023, doi: 10.1007/s00180-022-01207-6, [Online].

[15] K. K. Nicodemus and J. D. Malley, "Predictor correlation impacts machine learning algorithms: implications for genomic studies," *Bioinformatics*, vol. 25, no. 15, pp. 1884–1890, August, 2009, Accessed on: August, 16, 2023, doi: 10.1093/bioinformatics/btp331, [Online].

[16] The BMJ, "11. correlation and regression," https://www.bmj.com/about-bmj/resources-readers/publications/statistics-square-one/11-correlation-and-regression, Oct 2020, accessed August 12, 2023.

[17] J. Raymaekers and P. J. Rousseeuw, "Transforming variables to central normality," *Machine Learning*, pp. 1–23, 2021.

[18] M. Zuliani, "Ransac for dummies," *Vision Research Lab, University of California, Santa Barbara*, October, 2009, Accessed on: August, 16, 2023.

[19] K. G. Derpanis, "Overview of the ransac algorithm," *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, May, 2010, Accessed on: August, 16, 2023.

[20] D. Curran-Everett, "Explorations in statistics: the log transformation," *Advances in physiology education*, vol. 42, no. 2, pp. 343–347, June, 2018, Accessed on: August, 16, 2023, doi: 10.1152/advan.00018.2018, [Online].

[21] F. Changyong, W. Hongyue, L. Naiji, C. Tian, H. Hua, L. Ying *et al.*, "Log-transformation and its implications for data analysis," *Shanghai archives of psychiatry*, vol. 26, no. 2, p. 105, April, 2014, Accessed on: August, 16, 2023, doi: 10.3969/j.issn.1002-0829.2014.02.009, [Online].

[22] C. Feng, H. Wang, N. Lu, and X. M. Tu, "Log transformation: application and interpretation in biomedical research," *Statistics in medicine*, vol. 32, no. 2, pp. 230–239, July, 2012, Accessed on: August, 16, 2023, doi: 10.1002/sim.5486, [Online].

[23] O. N. Keene, "The log transformation is special," *Statistics in medicine*, vol. 14, no. 8, pp. 811–819, April, 1995, Accessed on: August, 16, 2023, doi: 10.1002/sim.4780140810, [Online].

[24] S. Weisberg, "Yeo-johnson power transformations," *Department of Applied Statistics, University of Minnesota. Retrieved June*, vol. 1, p. 2003, October, 2001, Accessed on: August, 16, 2023.

[25] I.-K. Yeo and R. A. Johnson, “A new family of power transformations to improve normality or symmetry,” *Biometrika*, vol. 87, no. 4, pp. 954–959, December, 2000, Accessed on: August, 16, 2023, doi: 10.1093/biomet/87.4.954, [Online].
[26] N. Weizing, ”A review and comparative study on univariate feature selection techniques,” M.S. thesis, Department of Mechanical Engineering, University of Cincinnati, Cincinnati, Ohio, 2012.
[27] Scikit-Learn. (2023) 1.13. feature selection. [Online]. Available: https://scikit-learn.org/stable/modules/feature_selection.html#univariate-feature-selection
[28] D. Berrar, “Cross-validation.” 2019.

TABLE I
VARIABLES DROPPED AND WHY

Variable	Description
CLASS	CLASS = 1 for all rows so it doesn't provide the machine learning model with any important information
Consequence	Redundant to the MC column
CLNDISDB	Storage in different databases is not relevant
CLNDN	ClinVar's name for information already in CLNDISDB column, redundant. Also, storage in ClinVar is not relevant to pathogenicity
CLNVI	Variant's clinical sources are not relevant to pathogenicity
CLNDISDBINCL, CLNDNINCL, CLNSIGINCL, SSR,DISTANCE, MOTIF_NAME, MOTIF_POS, HIGH_INF_POS, MOTIF_SCORE_CHANGE	Sparse Data, 0.20 percent or less of data is non-null
INTRON	Sparse Data, only 13 percent of data is non-null
CADD_RAW	Redundant, an untransformed version of CADD PHRED
BAM_EDIT	Is not relevant whether the file was edited or not
Allele	Redundant to ALT
CLNHGVS	Redundant to ALT and REF columns as well as CHROM and POS
BIOTYPE	Very low amount of variance, 48738 protein_coding and only 11 of any other type
ORIGIN	Contains values not described in the data documentation, also low variance with 47923 in one category
CLNVC	Very few values in categories other than single nucleotide variant
Feature_type	All values are uniform
CADD_PHRED, BLOSUM62,SIFT, PolyPhen	Other gene scores, not relevant to this study (deleted in a later supplementary coding file than others in this table but far before dropping nulls or encoding)

Variables removed due to (i) irrelevance, (ii) redundancy, (iii) low variance, and (iv) sparse data

TABLE II
NON-TRANSFORMED DATASET

Dataset Used	Model Used	Feature Selection	R2	MSE	RMSE	MAE	EV
df_loftool.csv	KNN Regressor	Univariate Feature Selection (f-regression)	0.44	-0.07	-0.26	-0.16	0.45
	KNN Regressor	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.95	-0.01	-0.08	-0.04	0.95
	Decision Tree Regressor	Univariate Feature Selection (f-regression)	0.96	-0.00	-0.07	-0.03	0.96
	Random Forest Regressor	Univariate Feature Selection (f-regression)	0.97	-0.00	-0.06	-0.03	0.98
	XGB	Univariate Feature Selection (f-regression)	0.97	-0.00	-0.06	-0.03	0.97
	SVR	Univariate Feature Selection (f-regression)	-0.32	-0.17	-0.41	-0.32	-0.10
	SVR	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.92	-0.01	-0.10	-0.08	0.92
	RANSAC	Univariate Feature Selection (f-regression)	0.90	-0.01	-0.11	-0.07	0.89
	RANSAC	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.88	-0.01	-0.11	-0.06	0.90

Five-fold cross-validated averages of r-squared, mean squared error, root mean squared error, mean absolute error and explained variance for the dataset that had no transformations applied to it. Univariate Feature Selection (Using F-regression) Feature Set: ['POS', 'cDNA_position', 'CDS_position', 'Protein_position', 'STRAND', 'CHROM', 'SYMBOL', 'Feature', 'EXON', 'Codons']

TABLE III
LOG-TRANSFORMED DATASET

Dataset Used	Model Used	Feature Selection	R2	MSE	RMSE	MAE	EV
df_loftool_log.csv	KNN Regressor	Univariate Feature Selection (f-regression)	0.41	-0.07	-0.27	-0.16	0.44
	KNN Regressor	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.95	-0.01	-0.08	-0.04	0.95
	Decision Tree Regressor	Univariate Feature Selection (f-regression)	0.96	-0.00	-0.07	-0.03	0.96
	Random Forest Regressor	Univariate Feature Selection (f-regression)	0.97	-0.00	-0.06	-0.03	0.97
	XGB	Univariate Feature Selection (f-regression)	0.97	-0.00	-0.06	-0.03	0.97
	SVR	Univariate Feature Selection (f-regression)	-0.32	-0.17	-0.41	-0.32	-0.10
	SVR	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.92	-0.01	-0.10	-0.08	0.92
	RANSAC	Univariate Feature Selection (f-regression)	0.22	-6.99	-0.16	-0.07	0.82
	RANSAC	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.90	-0.01	-0.11	-0.06	0.90

Five-fold cross-validated averages of r-squared, mean squared error, root mean squared error, mean absolute error and explained variance for the dataset that was logarithm transformed. Univariate Feature Selection (Using F-regression) Feature Set: ['POS', 'cDNA_position', 'CDS_position', 'Protein_position', 'STRAND', 'CHROM', 'SYMBOL', 'Feature', 'EXON', 'Codons']

TABLE IV
YEO-JOHNSON TRANSFORMED DATASET

Dataset Used	Model Used	Feature Selection	R2	MSE	RMSE	MAE	EV
df_loftool_yj.csv	KNN Regressor	Univariate Feature Selection (f-regression)	0.42	-0.07	-0.27	-0.16	0.44
	KNN Regressor	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.95	-0.01	-0.08	-0.04	0.95
	Decision Tree Regressor	Univariate Feature Selection (f-regression)	0.96	-0.01	-0.07	-0.03	0.96
	Random Forest Regressor	Univariate Feature Selection (f-regression)	0.97	-0.00	-0.06	-0.03	0.97
	XGB	Univariate Feature Selection (f-regression)	0.97	-0.00	-0.06	-0.03	0.97
	SVR	Univariate Feature Selection (f-regression)	-0.32	-0.17	-0.41	-0.32	-0.10
	SVR	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.92	-0.01	-0.10	-0.08	0.92
	RANSAC	Univariate Feature Selection (f-regression)	-5.2	-0.01	-1.96	-0.07	0.89
	RANSAC	Univariate Feature Selection (f-regression) (POS, cDNA_position, CDS_positon, Protein_positon removed)	0.83	-50.7	-0.11	-0.06	0.83

Five-fold cross-validated averages of r-squared, mean squared error, root mean squared error, mean absolute error and explained variance for the dataset that was Yeo-Johnson transformed. Univariate Feature Selection (Using F-regression) Feature Set: ['POS', 'cDNA_position', 'CDS_position', 'Protein_position', 'STRAND', 'CHROM', 'SYMBOL', 'Feature', 'EXON', 'Codons']

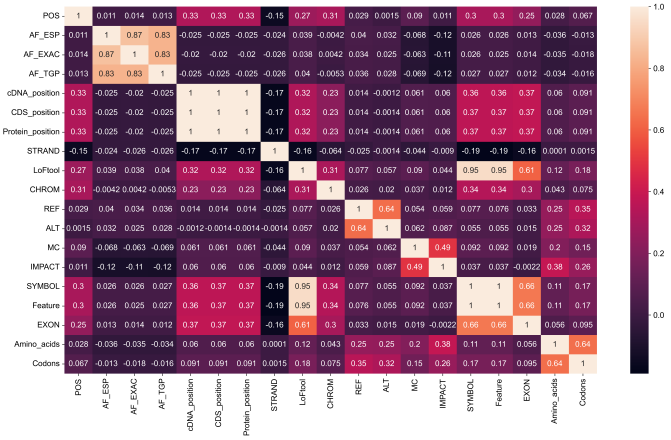
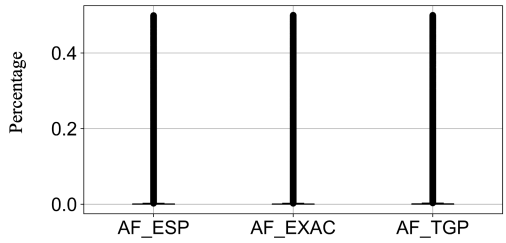
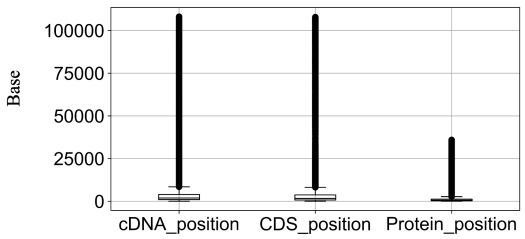


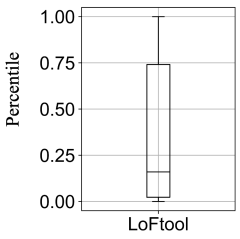
Fig. 1. Correlation Matrix: POS, cDNA position, CDS position, Protein position, CHROM, SYMBOL, Feature, and EXON are correlated with LoFtool. As can be seen above, several of these variables were highly correlated with each other (e.g. cDNA position, CDS position, and Protein position). These variables were kept in mind to drop or add when testing machine learning models. More details are given in Tables II, III, and IV.



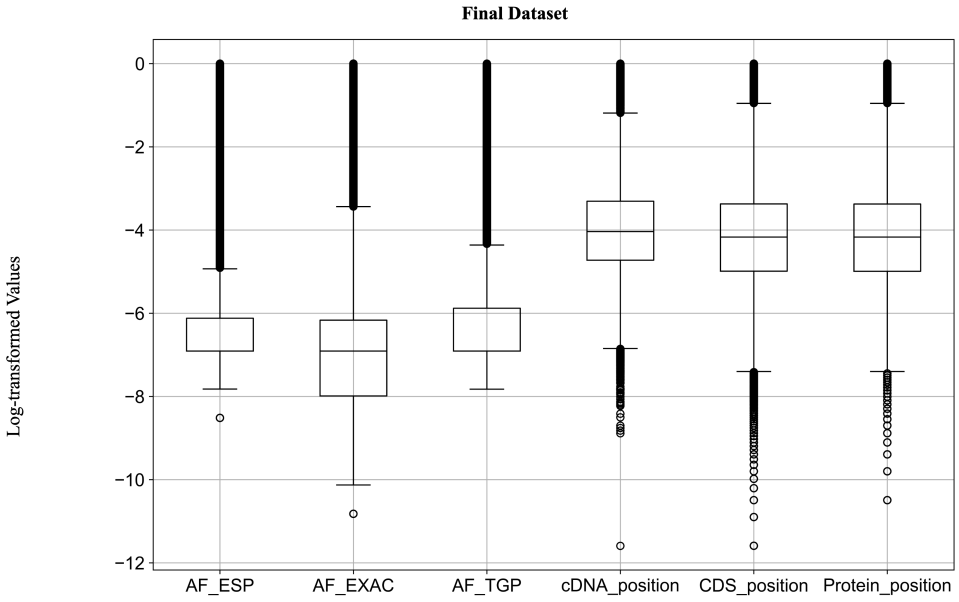
Allele Frequency in Various Populations From the Grand Opportunity Exome Sequencing Project (GO-ESP), Exome Aggregration Consortium (ExAC) and Thousand Genome Project (TGP)

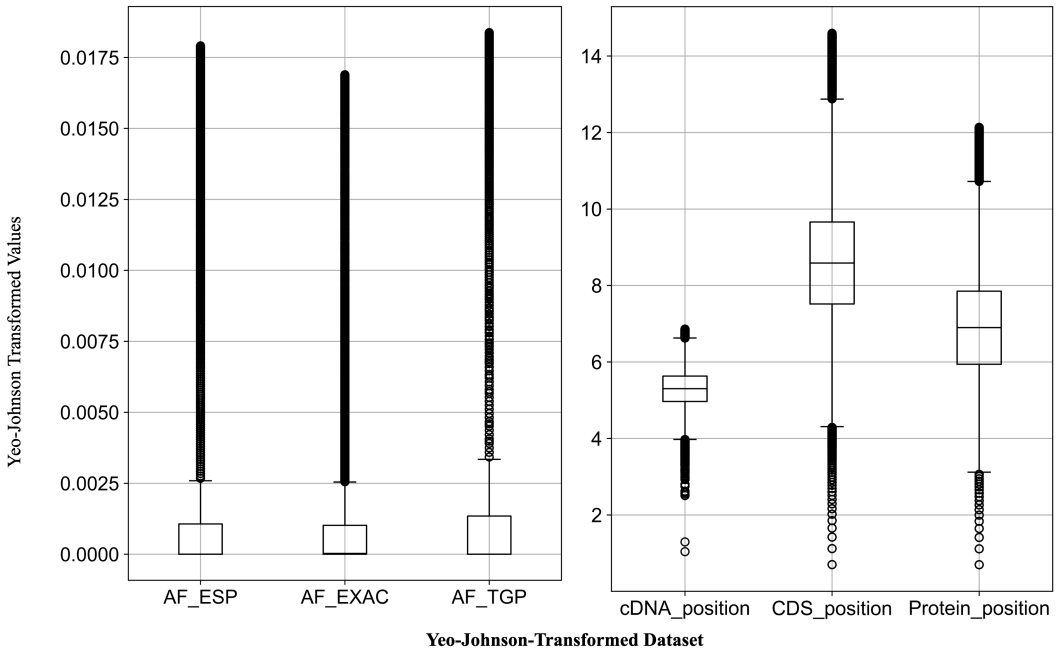


Relative Position of Base in cDNA, Coding or Protein Sequence



Percentile of gene's intolerance to loss-of-function mutation





Left to Right (For Transformed Box Plots): Allele Frequency in Various Populations From the Grand Opportunity Exome Sequencing Project (GO-ESP), Exome Aggregration Consortium (ExAC) and Thousand Genome Project (TGP). Relative Position of Base in cDNA, Coding or Protein Sequence

Fig. 2. Distribution of continuous variables before and after applying transformations. As can be seen in the plots, there were still many outliers left after both transformations. The log transformation normalized allele frequency columns more than the Yeo-Johnson transformation.