Credit : Lauren Solomon, Broad Communications

Human genetic variant classifications

Mid Eum Lee, PhD

May, 2020

Table of contents

1. Introduction

1.1. Objective

We now have tools and resources available for identifying and analyzing human genetic variants that are mysterious but critical in disease diagnosis. It has been known that people with high risk of diseases are based just on their genetic variation. ClinVar is a free repository for the collection of data from genetic tests and its associated diseases. When clinicians and laboratory experts enter the genetic testing results and phenotypes, classification results from different labs show conflicting classification errors. These clinical classification results range from benign to pathogenic, and the errors are based on the perceived classifications by clinicians. To reduce such a conflict due to human errors, we can identify critical features for accurate phenotypic predictions using machine learning models.

1.2. Significance

The model-based analysis provided by this study will provide high confidence on datasets and eventually help advancing disease diagnosis with genetic variants and translating genetic information into medicine. The appropriate classification of disease-related genes and its variants can open the development of the early diagnosis and treatments including gene therapy. This forward-looking strategy can reduce the error rate that impedes drug discovery for decades.

1.3. Data

This dataset is originated from Clinvar(https://www.ncbi.nlm.nih.gov/clinvar/) from NIH, and uploaded in the Kaggle (https://www.kaggle.com/kevinarvai/clinvar-conflicting). This dataset contains 65188 entries and total 46 columns. All features available from this dataset can be found in the Appendix.

The classification with conflicting results is categorized in CLASS column. 0 represents classification with consistent results and 1 represents conflicting classification.
There are three categories that clinicians assign based on variants.

1) Likely Benign or Benign
2) VUS (A variation in a genetic sequence for which the association with disease risk is unclear)[1]
3) Likely Pathogenic or Pathogenic

[1] "Definition of VUS - NCI Dictionary of Genetics Terms ...."
https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/vus.

The classification with conflict (CLASS : 1) is when two of any of the following three categories are present for one variant, two submissions of one category are not considered conflicting. The raw variant call format (vcf) file was downloaded here: ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh37/clinvar.vcf.gz The exact dataset used in this study is modified using the python script[2].

2. Data Exploration

2.1. Data cleaning

In the dataset, there are many features with null values. Before the data analysis, all columns with more than 50% of null values (NaN) were dropped and 31 columns were selected. Data types in most columns are "object". If we look at data closely, Protein_position, CDS_position, and cDNA_position are numbers but it says object. Some numeric values were separated by '_' so this was cleaned and converted into the first number. These changed values are saved in the new "KEY" column in the dataset.

In CHROM column, it contains numeric values and some alphabetic (e.g. 22 vs. '22'). All values are converted to integers and the extra entries, 'X' and 'MT' are assigned numbers 23 and 24 respectively.

The columns REF, ALT, and Allele contain base-pair sequences. The term REF (reference allele) refers to the base that is found in the reference genome. This may not be the major allele since the reference is just from someone's genome. The alternative allele (ALT) is any base that is found at that locus, other than the reference. This allele can be, or can not be linked to the phenotype and not necessarily the minor allele. It may not be useful to use allele information for predictions because there are only 4 nucleotides and most values are just one nucleotide. Since there are no clear patterns with a limited number of variables, these features were removed for the test.

'Amino_acids' and 'Codons' columns are removed since these are the translated genetic information (i.e. information for proteins). 'EXON' columns consist of fractions so the denominator and numerator are separated using the function. Exons are features of genes that map sequences of nucleotides that encode functional parts of DNA. Genes have different numbers of exons, some have few, some have many.

LabelEncoder was used to convert the rest of columns with object type into numeric values.

---

[2] "clinvar-kaggle/process_clinvar.py at master · arvkevi ... - GitHub." https://github.com/arvkevi/clinvar-kaggle/blob/master/process_clinvar.py.

2.2. Exploratory Data Analysis (EDA)

This dataset is highly skewed to 0 class feature (figure 1), meaning that there are less variants that result in conflicting submissions.

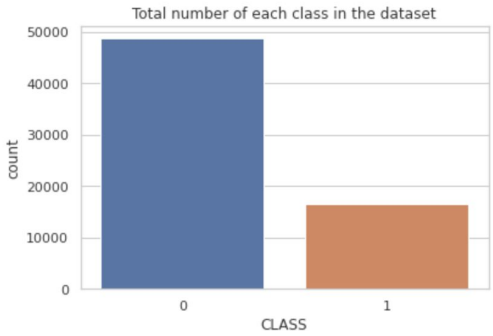

Figure 1. Class distribution of the dataset. Value counts for 0 is 48754, and for 1 is 16434.

To select important features for running the classification models, I explored each column feature categorized by "CLASS" feature and plotted values from class 1 as it is the class we are interested in.

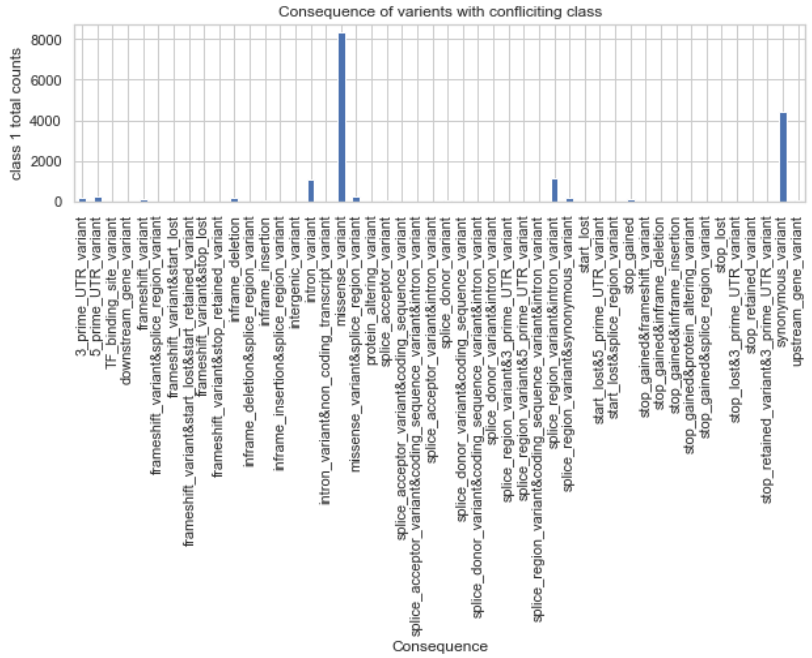

Figure 2. Consequence of variants. The consequence values plotted in X axis and the total counts of class 1 are plotted in Y axis.

The "Consequence" column indicates that most of the conflicting classifications are in missense_variant and synonymous_variant (Figure 2).

Missense variant is a nonsynonymous variant that changes one or more bases, resulting in a different amino acid sequence but where the length is preserved. Synonymous variant is a silent variant where there is no resulting change to the encoded amino acid. If we only compare these two categories, the missense variant is almost twice more than the synonymous variant.

In “CHROM” feature, most conflicting variants are enriched in chromosome 2 (Figure 3). In the “CLNVC” feature, single_nucleotide_variant is the dominant feature (Figure 4). This feature is also referred to as single nucleotide polymorphisms (SNP). SNPs are the most common type of genetic variants among people.

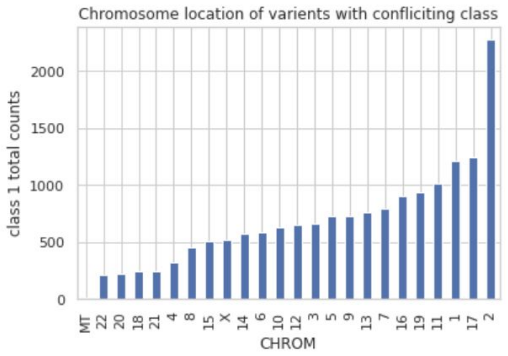

Figure 3. Bar plot of chromosome location of variants with conflicting class1.

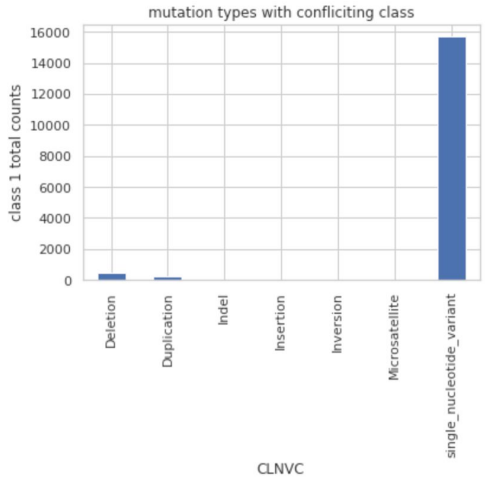

Figure 4. Bar plot of mutation types with the conflicting class 1.

2.3. Correlation analysis

Correlation is measuring the relationships between variables measured on a -1 to 1 scale. If the correlation value is close to -1 or 1, there is a stronger negative or positive relationship. If the value is closer to 0, there is a weaker relationship. It measures how change in one variable is associated with change in another variable.

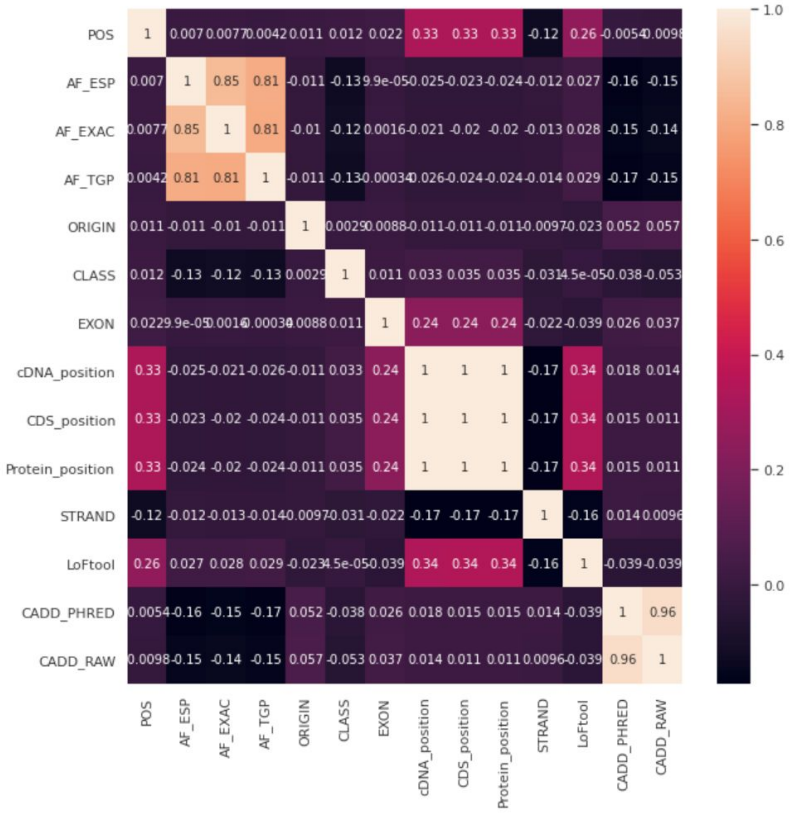

Figure 5. Correlation matrix of features.

As you can see in figure 5, the correlation matrix shows that AF_ESP, AF_EXAC,AF_TGP are highly correlated. These features are showing allele frequencies from different project sources. The term variant refers to a specific region of the genome that is different between two genomes. Different versions of the same variant are defined as alleles. cDNA_position, CDC_position, Protein_position are perfectly correlated so we can just use one feature among these features.

3. Modeling

3.1. Data pre-processing

Many machine learning algorithms cannot operate on label data directly so all input variables and output variables need to be numeric. 20 features that are cleaned and pre-processed were selected, and the data type of each feature was changed into numeric values using LabelEncoder. When we collect data and perform feature extractions, each feature is presented in a different scale. In order to run these modeling, we should perform feature scaling and mean normalization for reducing the impact of large valued features and allowing small valued features to contribute equally. Standardscaler is used for scaling. Next, data was splitted into two groups: training data (90%), testing data (10%). Scaled data was used in modelings.

3.2. Resampling methods for imbalanced classification

For modeling two resampling methods were applied to this imbalanced dataset: 1) undersampling, and 2) oversampling. Undersampling is to balance the class distribution for a classification dataset that has a skewed class distribution. Undersampling removes data from the training dataset that is in the majority class in order to better balance the class distribution. After undersampling it to 1000 samples, the skewed dataset was reduced almost into a 1:1 class distribution (Figure 6).

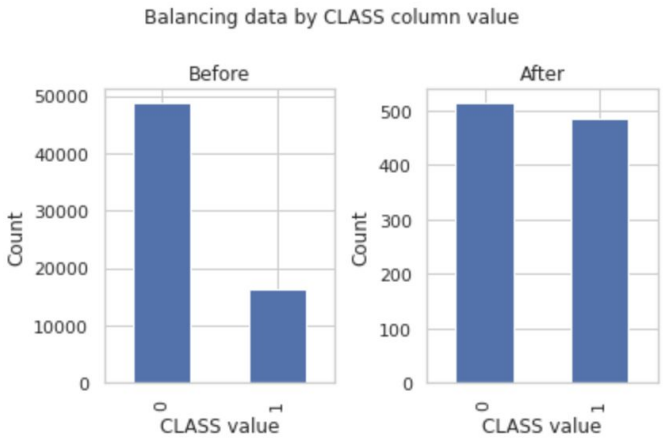

Figure 6. Bar plots of total class counts before balancing data (left), and after balancing data (right).

In the imbalanced classification data, there are too few examples of the minority class for a model to effectively learn the decision boundary so oversampling the examples in the minority class can be the solution. In this study, Synthetic Minority Oversampling Technique, or SMOTE was used for oversampling. SMOTE works with a random oversampling from the minority class. Then k of the nearest neighbors for that example are found (typically k=5). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space[3]. This resampling was only applied to the training set.

3.3. Logistic regression

Logistic regression is the typical choice of modeling for the binary classification problem. It is a linear method, but the predictions are transformed using the logistic function. Table 1 shows all metrics from the logistic regression classifier after resampling methods.

Table 1: logistic regression classifier - Resampling techniques were compared to No resampling

		Values			
No resampling	Class	Precision	Recall	F1-score	Accuracy
	0	0.75	1.00	0.85	0.74
	1	0.42	0.01	0.02	
UnderSampler	0	0.91	0.19	0.31	0.38
	1	0.28	0.94	0.44	
SMOTE	0	0.82	0.48	0.60	0.53
	1	0.31	0.70	0.43	

The model performance was compared to the results from other modelings. No resampling gave the highest accuracy (Table 1).

[3] "Imbalanced Learning | Wiley Online Books." 10 Jun. 2013, https://onlinelibrary.wiley.com/doi/book/10.1002/9781118646106.

3.4. Decision tree

Decision tree is a supervised learning technique that predicts outcomes by learning decision rules derived from features. It is a flowchart-like tree structure where an internal node represents features, the branch represents a decision rule, and each leaf node represents the outcome. Figure 7 shows the outcome of modeling with the datasets after undersampling.

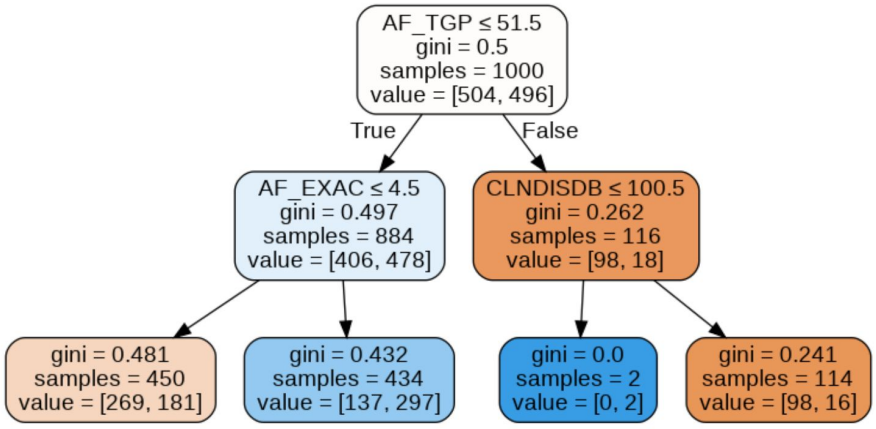

Figure 7. Graphics of decision tree from datasets after undersampling.

The Gini Index is calculated by subtracting the sum of the squared probabilities of each class from one. A gini score of zero means that within that node only a single class of samples exist. The value tells you how many samples at the given node fall into each category. Decision Tree is easy to interpret and it doesn't require any normalization. Since tree-based models do not require scaling the data, the decision tree is built again using data without scaling. All performance scores were the same as the results from standard scaling.

Table 2: Decision tree - Resampling techniques were compared to No resampling

		Values			
No resampling	Class	Precision	Recall	F1-score	Accuracy
	0	0.75	1.00	0.85	0.75
	1	0.75	0.00	0.00	
UnderSampler	0	0.75	1.00	0.85	0.75
	1	0.00	0.00	0.00	

SMOTE	0	0.00	0.00	0.00	0.25
	1	0.25	1.00	0.41	

The model performance was compared to the results from other modelings. No resampling and UnderSampler result in the same accuracy (Table 2).

3.5. Random Forest

Random forest is a supervised learning model. This model uses labeled data to "learn" how to classify unlabeled data. It is an ensemble method based on bagging (bootstrap aggregation) and consists of many decision trees. This modeling enables each individual tree to randomly sample from the dataset, resulting in different trees. In a random forest classifier, the number of trees in the forest was set at n_estimators=500. Here are the results of model performance.

Table 3: Random Forest - Resampling techniques were compared to No resampling

		Values			
No resampling	Class	Precision	Recall	F1-score	Accuracy
	0	0.80	0.93	0.86	0.78
	1	0.61	0.34	0.44	
UnderSampler	0	0.00	0.00	0.00	0.25
	1	0.25	1.00	0.41	
SMOTE	0	0.84	0.85	0.84	0.77
	1	0.54	0.52	0.53	

The model performance was compared to the results from other modelings. No resampling gave the highest accuracy (Table 3).

3.6. Gradient Boosting

Gradient Boosting is a boosting algorithm which produces a prediction model in the form of an ensemble of weak prediction models. By using gradient descent and updating predictions, we can find the values where loss function (MSE) is minimum.

Table 4: Gradient Boosting - Resampling techniques were compared to No resampling

		Values			
No resampling	Class	Precision	Recall	F1-score	Accuracy
	0	0.77	0.98	0.86	0.77
	1	0.69	0.15	0.25	
UnderSampler	0	0.00	0.00	0.00	0.25
	1	0.25	1.00	0.41	
SMOTE	0	0.85	0.76	0.80	0.72
	1	0.46	0.60	0.52	

The model performance was compared to the results from other modelings. No resampling gave the highest accuracy (Table 4).

3.7. XGBoost

XGBoost stands for eXtreme Gradient Boosting. The XGBoost implements the gradient boosting decision tree algorithm. Boosting is an ensemble technique where new models are added to correct the errors made by existing models.

Table 5: XGBoost - Resampling techniques were compared to No resampling

		Values			
No resampling	Class	Precision	Recall	F1-score	Accuracy
	0	0.77	0.97	0.86	0.76
	1	0.65	0.16	0.26	
UnderSampler	0	0.00	0.00	0.00	0.25
	1	0.25	1.00	0.41	
SMOTE	0	0.85	0.76	0.80	0.72
	1	0.46	0.60	0.52	

The model performance was compared to the results from other modelings. No resampling gave the highest accuracy (Table 5).

4. Model performance comparison

Machine learning classifiers tested in this study exhibited strong performance at discriminating binary classes. Table 6 shows the accuracy scores from all modelings. The dataset was fairly imbalanced (nearly 75% of entries are class 0) so both undersampling and oversampling (SMOTE) were applied and used for model training .

Table 6: The summary of Accuracy scores from Machine learning models.

	Before balancing data	After balancing data	
		UnderSampler	SMOTE
Logistic regression	0.74	0.38	0.53
Random Forest	0.78	0.25	0.77
Decision tree	0.75	0.75	0.25
Gradient Boosting	0.77	0.25	0.72
XGBoost Model	0.76	0.25	0.72

Based on the accuracy scores, random forest gave the highest accuracy score. After applying SMOTE, the random forest still gave better and accurate models compared to others.

5. Conclusion\future scope

We've known that there are people at high risk for disease based just on their overall genetic variation. In this study, some critical features from the dataset were selected by EDA analysis, and these selected attributes were used for machine learning modeling to predict conflicting variant classes. Based on decision tree modeling, AF_TGP feature (Allele frequencies from the 1000 genomes project) seems to be one of the critical attributes. So now we can narrow features or combinations thereof, which are common among conflicting variants so biologists could study these types of variants in more detail. By predicting conflicting variant classes, clinicians can be also cautious when they identify features. We conclude that machine learning can be used to build a practical classification tool to identify the critical mutations relevant for disease diagnosis for individual genetic variants.

6. Appendix

Feature descriptions:

- CHROM: Chromosome the variant is located on
- POS: Position on the chromosome the variant is located on.
- REF: Reference Allele
- ALT: Alternate Allele
- AF_ESP: Allele frequencies from GO-ESP
- AF_EXAC: Allele frequencies from ExAC
- AF_TGP: Allele frequencies from the 1000 genomes project
- CLNDISDB: Tag-value pairs of disease database name and identifier, e.g. OMIM:NNNNNN
- CLNDISDBINCL: For included Variant: Tag-value pairs of disease database name and identifier, e.g. OMIM:NNNNNN
- CLNDN: ClinVar's preferred disease name for the concept specified by disease identifiers in CLNDISDB
- CLNDNINCL: For included Variant : ClinVar's preferred disease name for the concept specified by disease identifiers in CLNDISDB
- CLNHGVS: Top-level (primary assembly, alt, or patch) HGVS expression.
- CLNSIGINCL: Clinical significance for a haplotype or genotype that includes this variant. Reported as pairs of VariationID:clinical significance.
- CLNVC: Variant Type
- CLNVI: the variant's clinical sources reported as tag-value pairs of database and variant identifier
- MC: comma separated list of molecular consequence in the form of Sequence Ontology ID|molecular_consequence
- ORIGIN: Allele origin. One or more of the following values may be added: 0 - unknown; 1 - germline; 2 - somatic; 4 - inherited; 8 - paternal; 16 - maternal; 32 - de-novo; 64 - biparental; 128 - uniparental; 256 - not-tested; 512 - tested-inconclusive; 1073741824 - other
- SSR: Variant Suspect Reason Codes. One or more of the following values may be added: 0 - unspecified, 1 - Paralog, 2 - byEST, 4 - oldAlign, 8 - Para_EST, 16 - 1kg_failed, 1024 - other
- CLASS: The binary representation of the target class. 0 represents no conflicting submissions and 1 represents conflicting submissions.
- Allele: the variant allele used to calculate the consequence
- Consequence: Type of consequence: https://useast.ensembl.org/info/genome/variation/prediction/predicted_data.html#consequences

- IMPACT: the impact modifier for the consequence type
- SYMBOL: Gene Name
- Feature_type: type of feature. Currently one of Transcript, RegulatoryFeature, MotifFeature.
- Feature: Ensembl stable ID of feature
- BIOTYPE: Biotype of transcript or regulatory feature
- EXON: the exon number (out of total number)
- INTRON: the intron number (out of total number)
- cDNA_position: relative position of base pair in cDNA sequence
- CDS_position: relative position of base pair in coding sequence
- Protein_position: relative position of amino acid in protein
- Amino_acids: only given if the variant affects the protein-coding sequence
- Codons: the alternative codons with the variant base in upper case
- DISTANCE: Shortest distance from variant to transcript
- STRAND: defined as + (forward) or - (reverse).
- BAM_EDIT: Indicates success or failure of edit using BAM file
- SIFT: the SIFT prediction and/or score, with both given as prediction(score)
- PolyPhen: the PolyPhen prediction and/or score
- MOTIF_NAME: the source and identifier of a transcription factor binding profile aligned at this position
- MOTIF_POS: The relative position of the variation in the aligned TFBP
- HIGH_INF_POS: a flag indicating if the variant falls in a high information position of a transcription factor binding profile (TFBP)
- MOTIF_SCORE_CHANGE: The difference in motif score of the reference and variant sequences for the TFBP
- LoFtool: Loss of Function tolerance score for loss of function variants: https://github.com/konradjk/loftee
- CADD_PHRED: Phred-scaled CADD score.
- CADD_RAW: Score of the deleteriousness of variants: http://cadd.gs.washington.edu/
- BLOSUM62See: http://rosalind.info/glossary/blosum62/