Genetic Variant Classification Using Classical Machine Learning Models and Bayesian Logistic Regression Model

Predicting whether a variant will have conflicting clinical classification

Background of the Study:

The use of genetic data to prevent diseases is extremely important especially with some deadly diseases such as Breast Cancer. These diseases might occur when there is a genetic variant in the DNA. Genetic variant is defined by the National Cancer Institute as “An alteration in the most common DNA nucleotide sequence. The term variant can be used to describe an alteration that may be benign, pathogenic, or of unknown significance.” Clinical laboratories are interested in finding these genetic variants and determine how likely they will trigger a disease for the patient. Most of the time clinical laboratories manually assign the type of the variant to a specific category: benign, likely benign, uncertain significance (VU), likely pathogenic, pathogenic. Some variants are not classified the same way across different laboratories. For example, a variant might be classified as Likely Benign In laboratory A can be classified as Likely Pathogenic by another laboratory B. This misclassification can affect the patients and sometimes the results can be deadly for them. Therefore, checking with different laboratories is necessary to guarantee the accuracy of the results. In an attempt to resolve the classification conflict issue, ClinVar(public database of genotype-phenotype relationships) and ClinGen(an NIH-funded resource that aims to define the clinical relevance of genes and variants in precision medicine) joined their efforts to gather and compare data of different laboratories as well as other sources in order to achieve optimal accuracy. In this project I tried to predict whether the variant is going to be the same among different laboratories or different.

Related Work:

Some interesting attempts have been done to solve this issue. We can find in Kaggle website two well elaborated analyses. The first one was done by Amandas et al. (2018) and the second one is done by Uddeshya (2018). Both works are great but they didn’t use Bayesian machine learning methods. Bayesian approaches can be used both to tune the parameters of the algorithms and to provide the associated accuracy of each machine learning algorithm applied to the relevant data. Therefore, I believe that by

tuning the parameters and using Bayesian Machine Learning algorithms I can associate with each prediction a certainty interval.

Process and Methods:

A) Preprocessing:

The original data set contained over 65,000 rows and 46 columns. Some columns contained over 90% of NAs, and other columns are just identifiers so I dropped them.

For the other columns, I replaced the NAs with the appropriate values. After searching the meaning of each column, I found that some NAs have actually zero value. For the position columns, I replaced the null values with the mean of the adjacent values. I also needed to parse some columns because they contained different values or characters. Specifically, the Intron and Exon Columns had two values: the length, and the number of the feature. After performing all these steps I used one hot encoding to create dummy variables for the character features, then I split the data into training set and testing set.

B) Conventional Machine Learning:

I used several Machine Learning classification algorithms including Logistic Regression, SGDClassifier, and Gaussian Naive Bayes. Moreover, to increase the accuracy I used Ensemble Machine Learning methods such as Random Forest and XGBoost.

Before doing any prediction, I selected a subset of features using Random Forest. Some algorithms work better with fewer predictors, either because some of the coefficients have a weak correlation with the target variable, or they have no relation at all.

B.1) Random Forest:

Random Forest(RF) classifier is an ensemble method built around decision Decision Tree algorithm. It creates several decision trees from a random subset of the training data. Than, it uses the voting strategy to predict the target variable in the test set.

From root node(top) to leaf node(down), Decision Tree splits at each node by selecting the optimal feature. The process of splitting is guided by the splitting criteria: selecting the node with the minimum impurity. This is done by maximizing the gain $\Delta I(S, T)$. The gain has the following equation:

$$\Delta I(S, T) = I(T) - \sum P(T_b \mid T) * I(T_b) \quad (1)$$

Where T_b is the bth node, $P(T_b|T)$ is the proportion of observations in t that are assigned to T_b. The impurity of a parent node t is defined as I(T). It is estimated by Entropy or Gini index criterion.

B.2) XGBoost:

Similar to Random Forest, XGBoost also built around Decision Tree, but instead of Bagging it uses another method called Boosting(Additive Training). At each step of the boosting a new tree is added to the prediction model. The goal of boosting is to optimise the objective function $J^{(t)}$ by selecting the appropriate tree $F_t(x_i)$ to be added to the prediction $\hat{Y}_i^{(t)}$.

$\hat{Y}_i^{(t)} = \sum F_k(x_i) = \hat{Y}_i^{(t-1)} + F_t(x_i)$ (2)

$J^{(t)} = \sum(Y - (\hat{Y}_i^{(t-1)} + F_t(x_i)))^2 + \sum\Omega(F_i)$ (3)

Where $\hat{Y}_i^{(t)}$ is the prediction i at the step t, F_t is the tree created at the step t, Ω is the regularization term.

B.3) Logistic Regression:

Logistic Regression on the other hand is quite different than the two ensemble methods.It has the following form:

logistic(X) = $\frac{1}{1+e-x}$ where $X = \beta 0 + \sum \beta j Xj$, (4)

Where β0, βj are the coefficients and Xj are the features.

The outcome of logistic regression is a probability between 1 and 0. Generally, if the outcome is bigger than 0.5 than we assign one into the target, but if it is less than 0.5 we assign 0 to our predicted variable.

B.4) SGDClassifier:

Another classifier I used in this project is called SGDClassifier. The main advantage of this model is its good performance on a large dataset. Stochastic gradient descent is used to minimise the cost function of this model. The cost function can be written as follow:

$cost(\theta, (X^{(i)}, Y) = (½) * (H_\theta(X^{(i)}) - Y^{(i)})^2$ (5)

Where θ_j is the jth coefficient, $X^{(i)}$ is the ith features vector, H_θ is the Logistic or the Linear Support Vector method.

The SGDClassifier performs two steps. The first one is shuffling the data. The second one is estimating the parameter θ_j in an iterative process. Where θ_j is being updated at each training sample. The amplitude of the update is dictated by the parameter α:

$\theta_j := \theta_j - \alpha * (1/m) \sum (H_\theta(X^{(i)} - Y^{(i)}) X_j^{(i)}$ (6)

B.5) Gaussian Naive Bayes:

The last model I used in this project is the Gaussian Naive Bayes. This model uses a probabilistic approach to estimate the target variable $\hat{Y}$. The model estimates the target variable by multiplying the frequency of Y in the data P(Y), with the likelihood $P(X_j|Y)$.

$\hat{Y} = \arg\max P(Y) \prod P(X_j|Y)$ (7)

The likelihood function in this model is assumed to be normally distributed.

C) Bayesian Logistic Regression:

The outcome of a bayesian model is a distribution. If the distribution is narrow that means we have a good prediction, but if it is wide that means the prediction is uncertain. In the bayesian approach we can include our prior knowledge about the parameters in order to get a better estimation. We can also apply a hierarchical model when dealing with data that can be split into groups. The hierarchical model helps to share information between groups. In this project, the variables INTRON and EXON are parts of the chromosome, so instead of encoding the CHROM feature into dummy variables, I used it for the hierarchical model. This will help share information between subsets of the data that is grouped by CHROM feature.

These are the models I used in this project:

m[i] := X[i]* beta[chrom[i]] + alpha[chrom[i]] (8)

mu := m + X2*beta2 + alpha2 (9)

Where i is the row number, m is a vector of prediction of length N (number of samples)), X is a vector of 2 which contains the INTRON and EXON variables, beta[chrom] and alpha[chrom] are the group coefficients, X2 is a matrix contains the other variables, beta2 and alpha2 are the coefficients of X2 variables.

The distribution associated with these two models are:

beta[chrom] ~ normal(mu_beta,sigma_beta);

alpha[chrom] ~ normal(mu_alpha, sigma_alpha);

beta2 ~ normal(0 , 10);

alpha2 ~ cauchy(0, 10);

mu_alpha ~ normal(0, 10);

sigma_alpha ~ cauchy(0, 10);

mu_beta ~ normal(0, 10);

sigma_beta ~ cauchy(0, 10);

y ~ bernoulli_logit(mu)

We can notice that beta[chrom] and alpha[chrom] have non constant parameters, because they are part of the hierarchical model.

D) Prediction Metrics:

In this project I assessed the accuracy of the models in order to compare between them and I assessed the the false negatives. Accuracy is the portion of correct predictions the model made, but the False negatives appears when the actual target value is one but the model predicts zero. That means, the classifier predicts that a genetic variant is “non-conflicting” while in fact it is conflicting. Therefore, the ROC curve is the appropriate measure to use in the context of this project.

Results and Discussion:

A) Feature Selection:

Before selecting features using random forest, I estimated the correlation between the features. Some features are strongly correlated with each other, either because the features have the same information collected from different sources(in the case of AF_ESP, AF_EXACT, AF_TGP), or because they have similar information(in the case of positions). After dropping the correlated features I selected a subset of important features to use for prediction. As we can see in fig.2 Allele Frequency from ExAC is the most important feature. The second important feature is POS which is the position in the chromosome the variant is located in. CADD_PHRED which is a tool of scoring the deleteriousness of the SNV(single nucleotide variant) is the third most important feature.

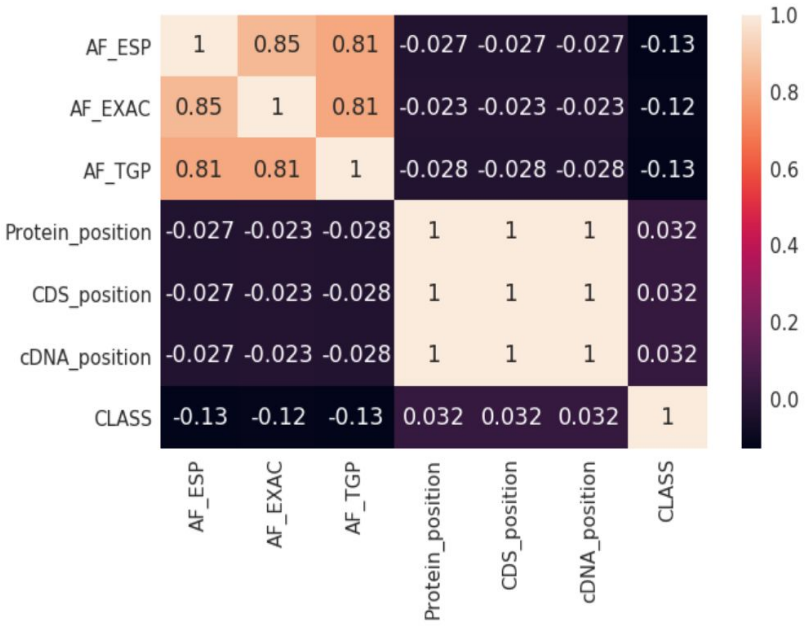

Fig.1 This figure shows a strong correlation between some features. Specifically position variables are strongly correlated

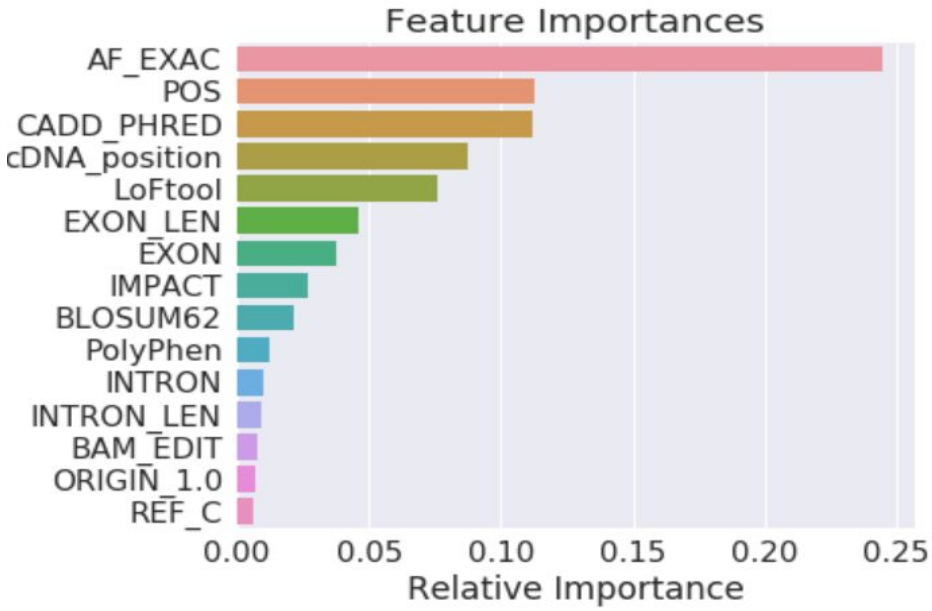

Fig.2 This figure shows the feature importance of the top 15 variables

B) model performance:

Random Forest classifier outperformed the other models in terms of accuracy, and ROC curve. Table.1 shows the performance of each model.

Model	XGBoost	Random Forest	Logistic Regression	GaussianNB	SGDClassifier
Accuracy	0.7565	0.7574	0.7461	0.675	0.7459
ROC curve	0.55	0.58	0.5	0.55	0.5

Table.1 Accuracy, ROC, and False Negative of the 5 classifiers used in this project

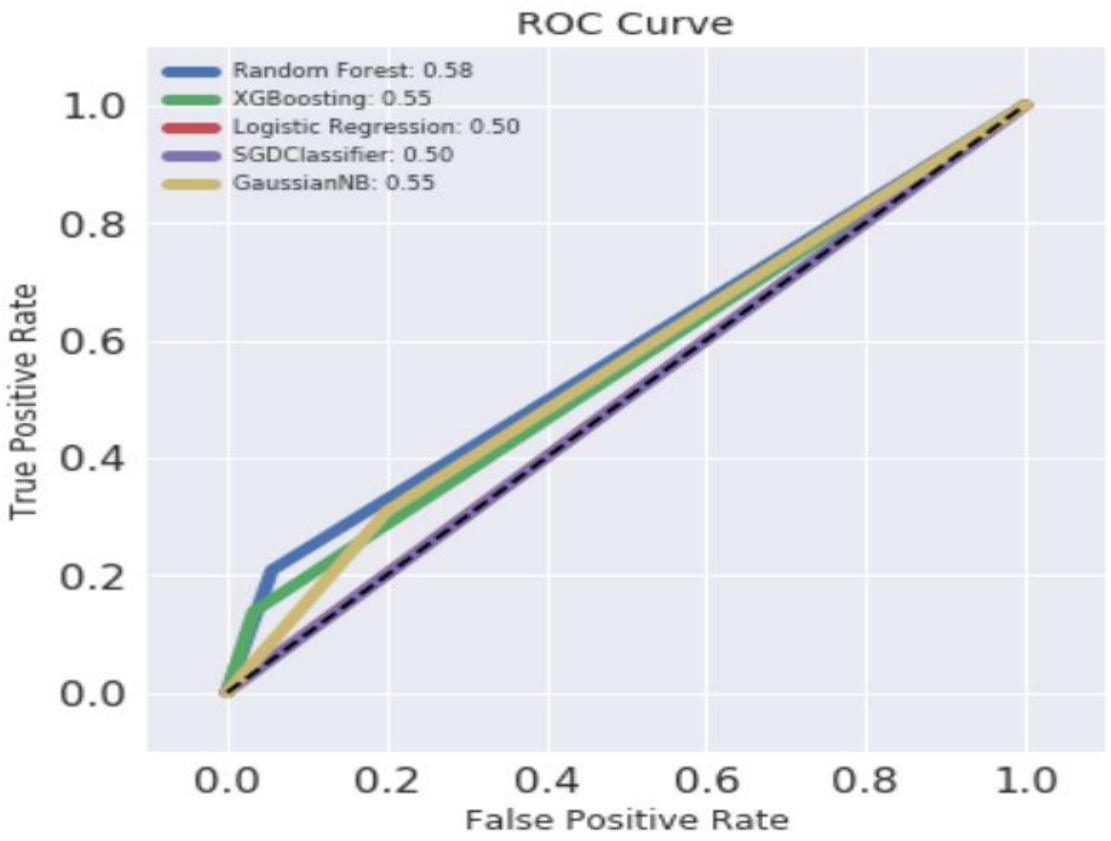

Fig.3 Random Forest performed better the the other models

Even Though I used grid search to tune the parameters, the overall performance of the five models is low. This could be due to the lack of important other features or these models do not reflect the hierarchy present in the data

Conclusion:

Allele frequency is the most important feature to predict the whether a variant will have a conflicting clinical classification. The other features such as the position in the chromosome the variant is located in and the deleteriousness score of the SNV(single nucleotide variant) are also important.

The models in this project have poor performance, but I believe that with a more sophisticated models and more relevant features the performance can be improved substantially. Therefore, there is a possibility to solve this problem with machine learning algorithms.

The Bayesian model I proposed in this project can be extended beyond Intron and exon to include more levels, because many features in this data set are actually part of the other features.

References:

Armbruester, A., Billah, M., Bocicariov, V., Luedemann, N., Salzwedel, R., & Taneja, R. (2018). ClinVar - Identifying Conflicting Genetic Variant https://www.kaggle.com/vasilyb/clinvar-identifying-conflicting-genetic-variants

Human Genetic Variants. (n.d). Retrieved from ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh37/

Kaggle. “Genetic Variant Classifications.” 2017. https://www.kaggle.com/kevinarvai/clinvar-conflicting

NCI Dictionary of Genetics Terms. (n.d). Retrieved from https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/genetic-variant

Uddeshya, S. (2018). Conflicting Result Classifications. Posted in Genetic Variant Classifications https://www.kaggle.com/uds5501/conflicting-result-classifications