ML Project Presentation

Personalized Medicine: Redefining Cancer

Group 54
-Dhruva Sahrawat (2015026)
-Aditya Adhikary (2015007)

Introduction

- In this project, we have an expert-annotated knowledge base where researchers have manually annotated thousands of mutations in genes.
- These mutations have been classified into 9 classes, some contributing to tumor growth (drivers) from the neutral mutations (passengers).
- Our goal is to find a machine learning algorithm that, when given the text, automatically classifies these genetic variations.
- The performance of machine learning algorithm is ranked on the basis of log-loss which is calculated through unknown test data on the Kaggle Challenge.

Initial Approach and Data Preprocessing

- The training set consists of 3321 samples and the test set, 968 samples.
- The training set consists of columns ID, Gene, Variation, and Class, which we have to predict.

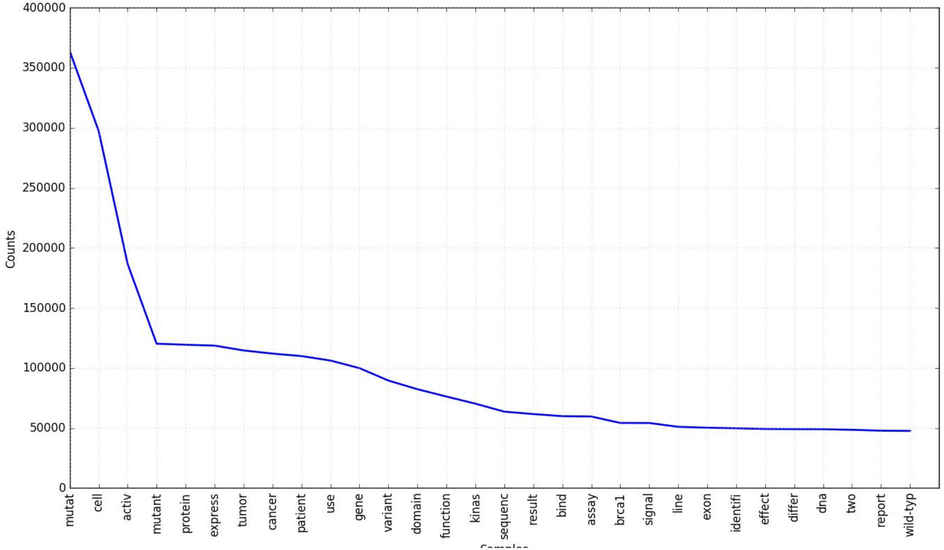
We have used the *nltk* library to

- Tokenize the documents to words and remove punctuations
- Stem the words to their roots
- Remove commonly occurring stop words as well as common words found in literature, such as “et”, “al”, “study”, “figure”,”result”, ”conclusion”, ”author” etc
- We do the same step on the test dataset before prediction.

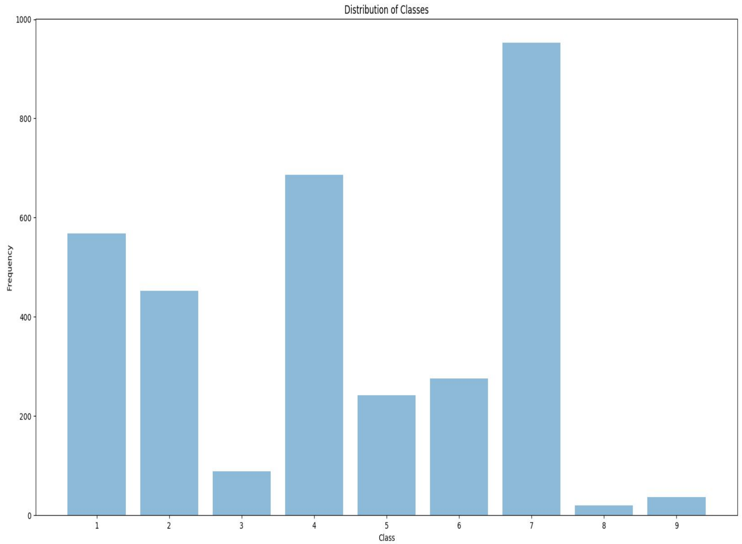
- We also found that the training set was unbalanced in favour of a particular class(7). So we also tried upsampling for the classes with lower number of labels associated with them.

Initial Approach and Data Preprocessing

- We also used TruncatedSVD on the tf-idf vectors, which uses SVD on them to perform dimensionality reduction.
- In TruncatedSVD, a rank-reduced, singular value decomposition is performed on the feature matrix obtained from tfidf to determine patterns in the relationships between the terms and concepts contained in the text. This preserves the most important semantic information in the text while reducing noise.
- We also encoded the ‘Gene’ and ‘Variation’ columns to our feature vector for better classification by using a one hot encoding and then latter using truncated SVD to reduce dimensionality.



Left: Frequency distribution of the most commonly occurring words in the text (after pre-processing)



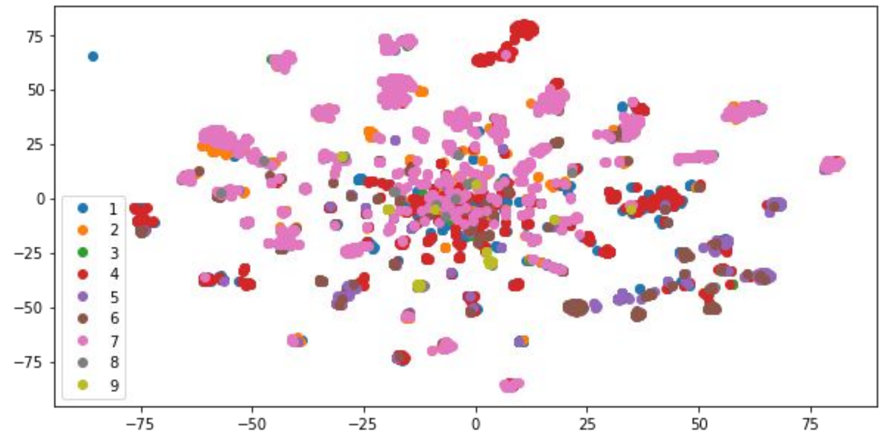
Right: Frequency distribution of the classes

Feature Extraction and Evaluation Metrics

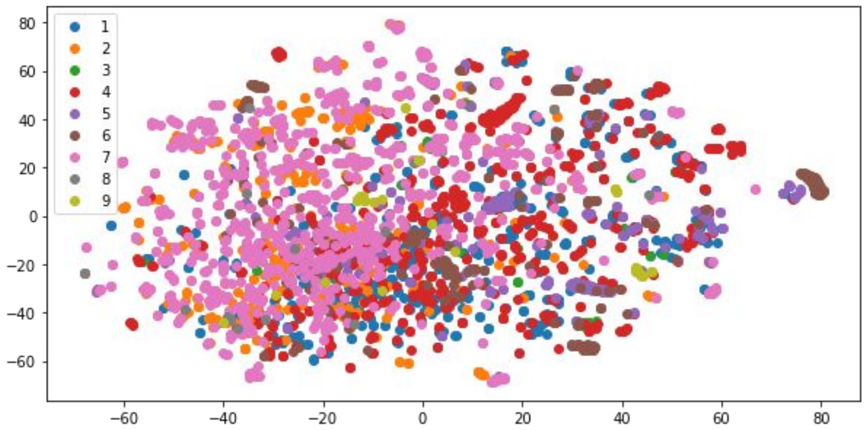
- We have used the Bag-of-Words and Bigram/Trigram models by using the *CountVectorizer* and *Tfidf* libraries to convert variable length text documents to sparse vectors before running any machine learning algorithm on them.
- *StratifiedKFold* has been used to cross-validate and *cross_val_predict* to fit a model and predict on the training dataset.
- We plotted confusion matrices, found the prediction accuracies and multi class log losses for each model under consideration.

Feature Extraction and Evaluation Metrics - Contd

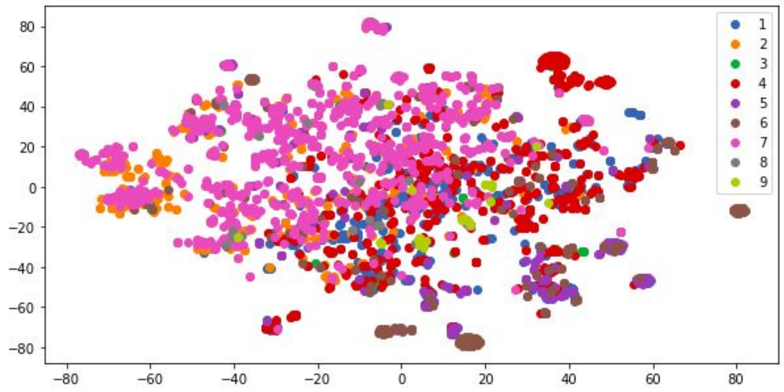
- We also used Precision (Positive Predictive Value) and Recall (True Positive Rate/Sensitivity) and F-Score as evaluation metrics too since the classes were highly unbalanced.
- We also used Word2Vec and Doc2Vec, which consist of shallow, two-layer neural network models that are trained to reconstruct linguistic contexts of words. Word2Vec contains two distinct models, Continuous Bag-of-Words and Skip-gram. The first is to predict the probability of a word given a context(another set of words), and the other vice versa.
- We select the model with the best performance during cross-validation, i.e the best combination of log-loss and accuracy scores, having a balanced confusion matrix, and high precision and recall.



We used TruncatedSVD and TSNE to plot the data points whose features are simple sparse matrices from TfIdf.



After we use doc2vec and add gene and variation to our feature pool, we get the following representation of data points through TSNE.



<- Similarly, for Word2vec. As we can see the data points are not linearly separable for any of them. The doc2vec/word2vec vectors have less no of features and give better accuracy compared with normal Tfidf vectors.

Analysis and Learning Techniques

Learning Techniques:-

- Multinomial Naive Bayes classifier, since it works well as a benchmark and is suited for sparse matrices of the type the vectorizers output for the samples.
- Logistic Regression, as it uses a one vs rest strategy for classification and serves as another essential base model.
- SVC - due to its ability to handle non-linearly separable datasets with the help of kernels.
- RandomForest - since it is an ensemble approach and can overcome the overfitting problem associated with Decision Trees,
- AdaBoost - as this is a standard boosting algorithm which controls both the aspects of bias & variance, and is considered to be more effective. New models are created that predict the residuals or errors of prior models and then added together to make the final prediction.

Analysis and Learning Techniques (contd)

Then, it uses a gradient descent algorithm to minimize the loss when adding new models.

- XGBoost, an advanced implementation of gradient boosting algorithm, which has a much better execution speed and is highly flexible and portable. Xgboost and other models follows the principle of gradient boosting, but the difference in modeling details, as it uses a more regularized model formalization to control overfitting.
- Neural Network using Keras with loss as cross entropy, and metrics as accuracy, with a validation split of 0.1.

Analysis and Learning Techniques (contd)

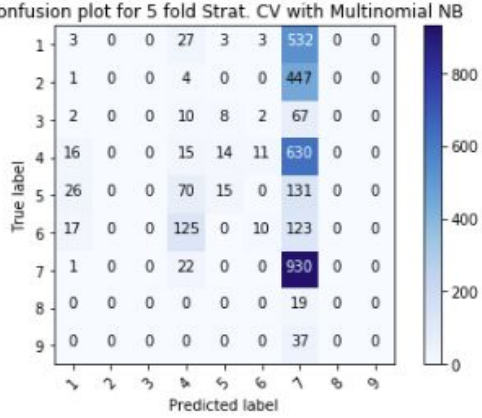
Hyperparameter Optimization:-

- We have used GridSearch- Using sklearn- GridSearchCV to go through optimum parameters
- We have used simple parameters for a few selected better-performing models like RandomForest - ’n_estimators’ and ’max_features’, for SVC - ‘C’, ‘decision_function_shape’, for XGB - ‘learning rate’ and ‘max_depth’, and for ADABoost - ‘learning_rate’ and ‘n_estimators’.

Results

Log loss: 2.74217844806
Accuracy: 0.292984040952

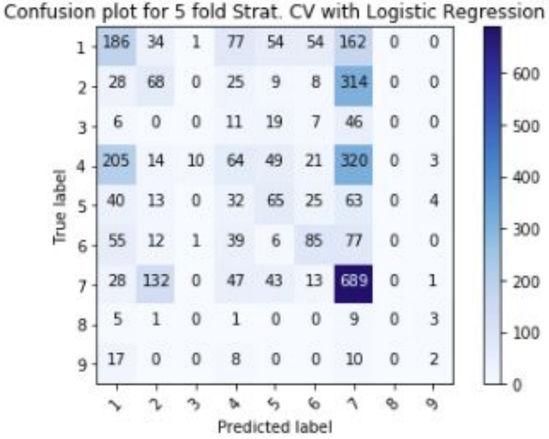
	precision	recall	f1-score	support
1	0.05	0.01	0.01	568
2	0.00	0.00	0.00	452
3	0.00	0.00	0.00	89
4	0.05	0.02	0.03	686
5	0.38	0.06	0.11	242
6	0.38	0.04	0.07	275
7	0.32	0.98	0.48	953
8	0.00	0.00	0.00	19
9	0.00	0.00	0.00	37
avg / total	0.17	0.29	0.16	3321



['multinomial_nb.pkl']

Log loss: 1.76259870904
Accuracy: 0.34899126769

	precision	recall	f1-score	support
1	0.33	0.33	0.33	568
2	0.25	0.15	0.19	452
3	0.00	0.00	0.00	89
4	0.21	0.09	0.13	686
5	0.27	0.27	0.27	242
6	0.40	0.31	0.35	275
7	0.41	0.72	0.52	953
8	0.00	0.00	0.00	19
9	0.15	0.05	0.08	37
avg / total	0.30	0.35	0.31	3321

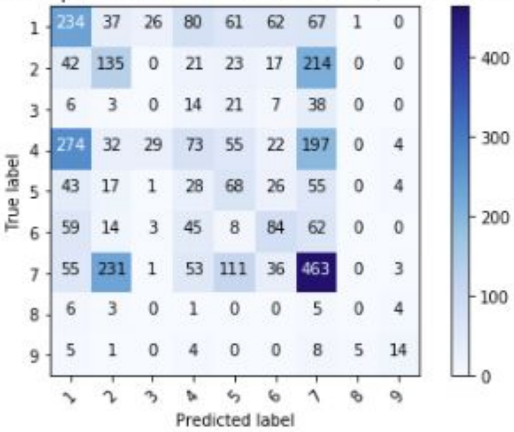


['logistic_regr.pkl']

Few preliminary models with tfidf

```
Log loss:  1.95761040555
Accuracy:  0.322493224932
             precision    recall  f1-score   support

          1       0.32      0.41      0.36       568
          2       0.29      0.30      0.29       452
          3       0.00      0.00      0.00        89
          4       0.23      0.11      0.15       686
          5       0.20      0.28      0.23       242
          6       0.33      0.31      0.32       275
          7       0.42      0.49      0.45       953
          8       0.00      0.00      0.00        19
          9       0.48      0.38      0.42        37

avg / total       0.31      0.32      0.31      3321
```

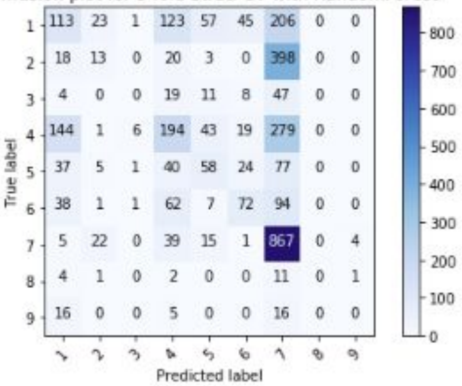
Confusion plot for 5 fold Strat. CV with SVM, linear kernel




```
Fitting model!

['svm linear.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation...

[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed: 32.0min finished

Log loss:  1.72326575758
Accuracy:  0.396567299006
             precision    recall  f1-score   support

          1       0.30      0.20      0.24       568
          2       0.20      0.03      0.05       452
          3       0.00      0.00      0.00        89
          4       0.38      0.28      0.33       686
          5       0.30      0.24      0.27       242
          6       0.43      0.26      0.32       275
          7       0.43      0.91      0.59       953
          8       0.00      0.00      0.00        19
          9       0.00      0.00      0.00        37

avg / total       0.34      0.40      0.33      3321

/home/aditya15007/anaconda3/lib/python3.6/site-packages/sklearn/metrics/c
ng: Precision and F-score are ill-defined and being set to 0.0 in labels
  'precision', 'predicted', average, warn_for)
```

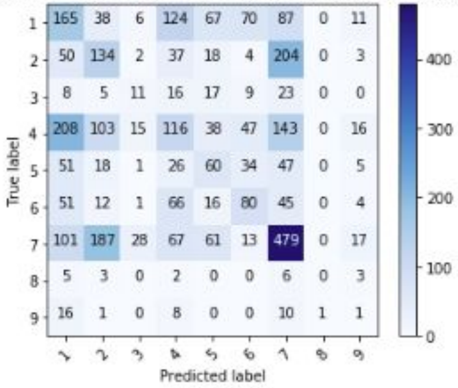
Confusion plot for 5 fold Strat. CV with RandomForest




```
['rand_for.pkl']
```

```
Finished transforming data in  172.34492421150208  secs
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   12.4s finished
Log loss:  2.42271174732
Accuracy:  0.314965371876
             precision    recall  f1-score   support

          1       0.25      0.29      0.27       568
          2       0.27      0.30      0.28       452
          3       0.17      0.12      0.14        89
          4       0.25      0.17      0.20       686
          5       0.22      0.25      0.23       242
          6       0.31      0.29      0.30       275
          7       0.46      0.50      0.48       953
          8       0.00      0.00      0.00        19
          9       0.02      0.03      0.02        37

avg / total       0.31      0.31      0.31      3321
```

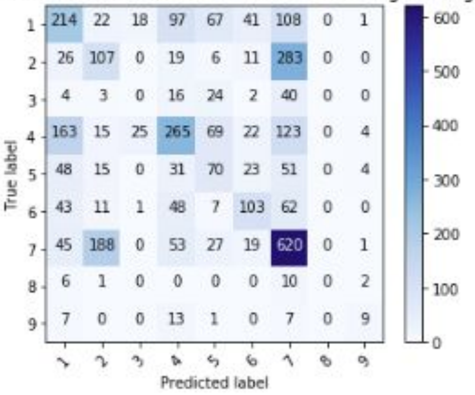
Confusion plot for 5 fold Strat. CV with Truncated SVD and RandomForest




```
['rand_for_trunc_tfidf.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:    1.6s finished
Log loss:  1.65313998161
Accuracy:  0.417946401686
             precision    recall  f1-score   support

          1       0.38      0.38      0.38       568
          2       0.30      0.24      0.26       452
          3       0.00      0.00      0.00        89
          4       0.49      0.39      0.43       686
          5       0.26      0.29      0.27       242
          6       0.47      0.37      0.42       275
          7       0.48      0.65      0.55       953
          8       0.00      0.00      0.00        19
          9       0.43      0.24      0.31        37

avg / total       0.41      0.42      0.41      3321

/home/aditya15007/anaconda3/lib/python3.6/site-packages/sklearn/metrics/
ng: Precision and F-score are ill-defined and being set to 0.0 in labels
  'precision', 'predicted', average, warn_for)
```

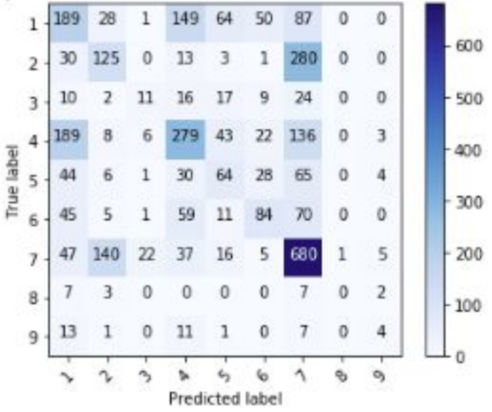
Confusion plot for 5 fold Strat. CV with Word2Vec and Logistic Regression




```
['logistic_w2v.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   31.5s finished
Log loss:  2.02685740982
Accuracy:  0.432399879554
             precision    recall  f1-score   support

          1       0.33      0.33      0.33       568
          2       0.39      0.28      0.32       452
          3       0.26      0.12      0.17        89
          4       0.47      0.41      0.44       686
          5       0.29      0.26      0.28       242
          6       0.42      0.31      0.35       275
          7       0.50      0.71      0.59       953
          8       0.00      0.00      0.00        19
          9       0.22      0.11      0.15        37

avg / total       0.42      0.43      0.42      3321
```

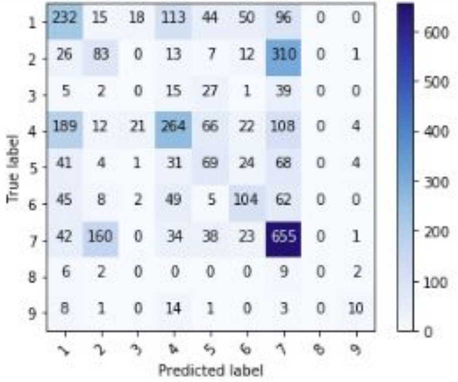
Confusion plot for 5 fold Strat. CV with Word2Vec and RandomForest




```
['rand_for_w2v.pkl']
```

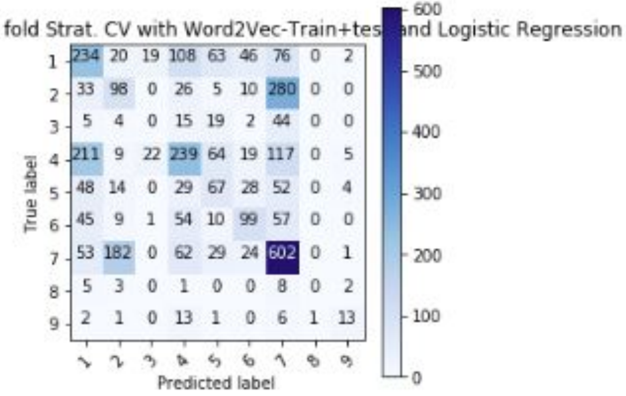
```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:    7.8s finished
Log loss:  1.58455459641
Accuracy:  0.426678711232
             precision    recall  f1-score   support

          1       0.39      0.41      0.40       568
          2       0.29      0.18      0.22       452
          3       0.00      0.00      0.00        89
          4       0.50      0.38      0.43       686
          5       0.27      0.29      0.28       242
          6       0.44      0.38      0.41       275
          7       0.49      0.69      0.57       953
          8       0.00      0.00      0.00        19
          9       0.45      0.27      0.34        37

avg / total       0.41      0.43      0.41      3321

/home/aditya15007/anaconda3/lib/python3.6/site-packages/sklearn/metrics/cl
ng: Precision and F-score are ill-defined and being set to 0.0 in labels w
  'precision', 'predicted', average, warn_for)
```

Confusion plot for 5 fold Strat. CV with Word2Vec and SVM linear kernel



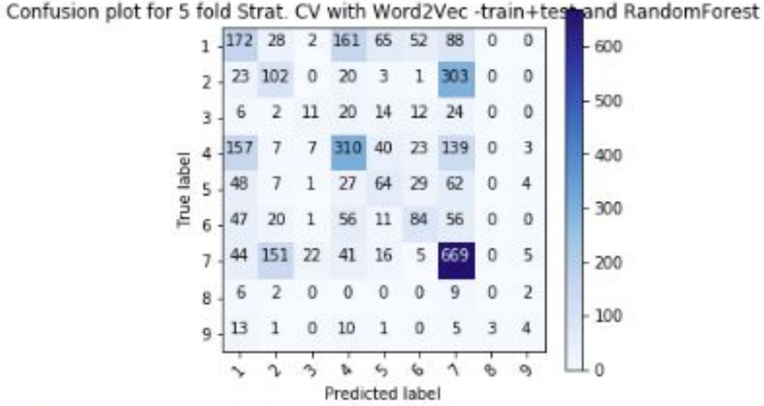

```
['svm_linear_w2v.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation..
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:    8.2s fir
Log loss:  1.66819606622
Accuracy:  0.407106293285
             precision    recall  f1-score   support

          1       0.37      0.41      0.39       568
          2       0.29      0.22      0.25       452
          3       0.00      0.00      0.00        89
          4       0.44      0.35      0.39       686
          5       0.26      0.28      0.27       242
          6       0.43      0.36      0.39       275
          7       0.48      0.63      0.55       953
          8       0.00      0.00      0.00        19
          9       0.48      0.35      0.41        37

avg / total       0.39      0.41      0.39      3321
```




```
['logistic_w2v_trtest.pkl']
```

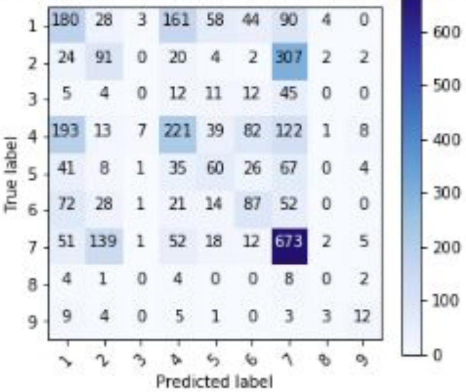
```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   29.6s finished
Log loss:  1.97700589683
Accuracy:  0.426377597109
             precision    recall  f1-score   support

          1       0.33      0.30      0.32       568
          2       0.32      0.23      0.26       452
          3       0.25      0.12      0.17        89
          4       0.48      0.45      0.47       686
          5       0.30      0.26      0.28       242
          6       0.41      0.31      0.35       275
          7       0.49      0.70      0.58       953
          8       0.00      0.00      0.00        19
          9       0.22      0.11      0.15        37

avg / total       0.41      0.43      0.41      3321
```




```
['rand_for_w2v_trtest.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:  1.5min finis
Log loss:  1.6045817989
Accuracy:  0.398675097862
             precision    recall  f1-score   support

          1       0.31      0.32      0.31       568
          2       0.29      0.20      0.24       452
          3       0.00      0.00      0.00        89
          4       0.42      0.32      0.36       686
          5       0.29      0.25      0.27       242
          6       0.33      0.32      0.32       275
          7       0.49      0.71      0.58       953
          8       0.00      0.00      0.00        19
          9       0.36      0.32      0.34        37

avg / total       0.37      0.40      0.38      3321
```

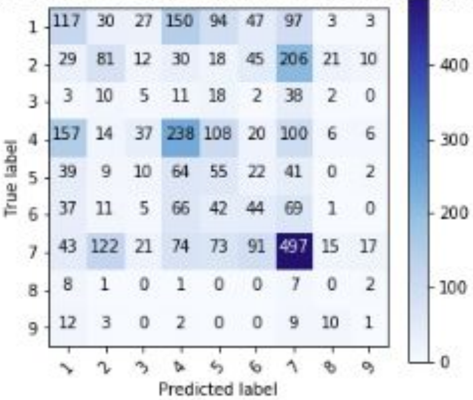
Confusion plot for 5 fold Strat. CV with Word2Vec -train+test and XGBClassifier




```
['xgb_w2v_trtest.pkl']
```

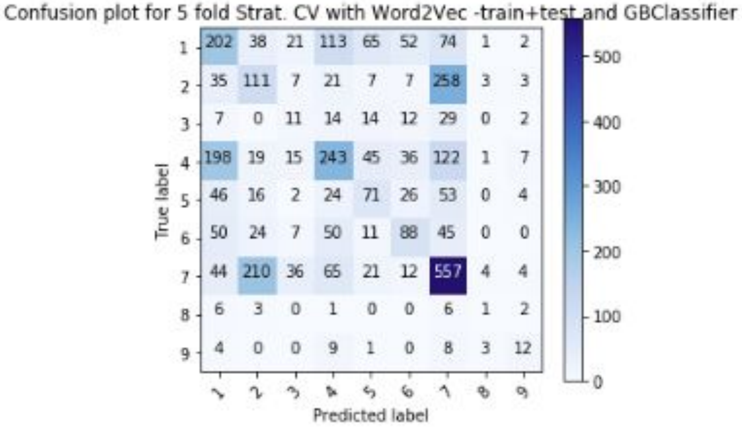
```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   51.5s finished
Log loss:  1.96729029737
Accuracy:  0.312556458898
             precision    recall  f1-score   support

          1       0.26      0.21      0.23       568
          2       0.29      0.18      0.22       452
          3       0.04      0.06      0.05        89
          4       0.37      0.35      0.36       686
          5       0.13      0.23      0.17       242
          6       0.16      0.16      0.16       275
          7       0.47      0.52      0.49       953
          8       0.00      0.00      0.00        19
          9       0.02      0.03      0.03        37

avg / total       0.32      0.31      0.31      3321
```

Confusion plot for 5 fold Strat. CV with Word2Vec -train+test and XGBClassifier




```
['ada_w2v_trtest.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:  3.9min finished
Log loss:  2.91272899578
Accuracy:  0.390243902439
             precision    recall  f1-score   support

          1       0.34      0.36      0.35       568
          2       0.26      0.25      0.25       452
          3       0.11      0.12      0.12        89
          4       0.45      0.35      0.40       686
          5       0.30      0.29      0.30       242
          6       0.38      0.32      0.35       275
          7       0.48      0.58      0.53       953
          8       0.08      0.05      0.06        19
          9       0.33      0.32      0.33        37

avg / total       0.39      0.39      0.39      3321
```

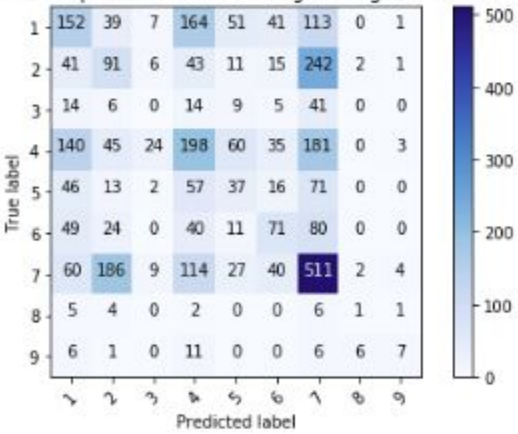



```
['gbc_w2v_trtest.pkl']
```

Hence, we tried out different models and discovered that Random Forest with the Word2Vec model performed the best amongst the rest in the cross-validation step.

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:    5.1s finished
Log loss:  2.00292969257
Accuracy:  0.321589882565
             precision    recall  f1-score   support

          1       0.30      0.27      0.28       568
          2       0.22      0.20      0.21       452
          3       0.00      0.00      0.00        89
          4       0.31      0.29      0.30       686
          5       0.18      0.15      0.17       242
          6       0.32      0.26      0.29       275
          7       0.41      0.54      0.46       953
          8       0.09      0.05      0.07        19
          9       0.41      0.19      0.26        37

avg / total       0.31      0.32      0.31      3321
```

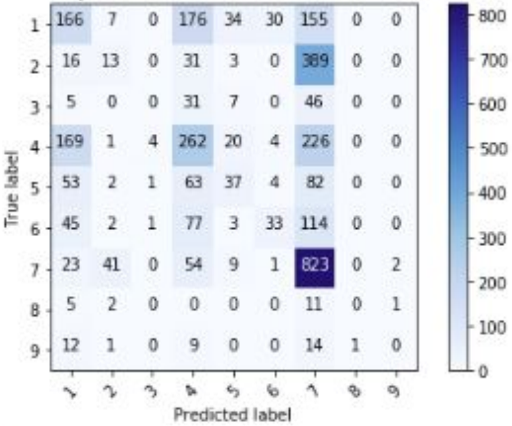
Confusion plot for 5 fold Strat. logistic regression d2v




```
['logistic_regression_d2v.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   19.6s finished
Log loss:  1.63484242934
Accuracy:  0.401686239085
             precision    recall  f1-score   support

          1       0.34      0.29      0.31       568
          2       0.19      0.03      0.05       452
          3       0.00      0.00      0.00        89
          4       0.37      0.38      0.38       686
          5       0.33      0.15      0.21       242
          6       0.46      0.12      0.19       275
          7       0.44      0.86      0.59       953
          8       0.00      0.00      0.00        19
          9       0.00      0.00      0.00        37

avg / total       0.35      0.40      0.34      3321
```

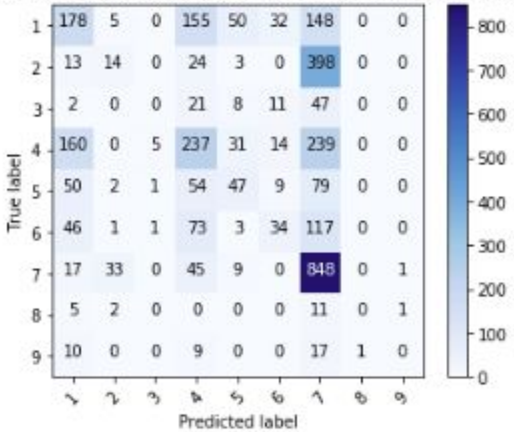
Confusion plot for 5 fold Strat. CV with RandomForest




```
['d2vrand_for.pkl']
```

```
Cross validating model using 5-fold Stratified cross validation
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   17.1s f
Log loss:  1.63364285836
Accuracy:  0.408912978019
             precision    recall  f1-score   support

          1       0.37      0.31      0.34       568
          2       0.25      0.03      0.06       452
          3       0.00      0.00      0.00        89
          4       0.38      0.35      0.36       686
          5       0.31      0.19      0.24       242
          6       0.34      0.12      0.18       275
          7       0.45      0.89      0.59       953
          8       0.00      0.00      0.00        19
          9       0.00      0.00      0.00        37

avg / total       0.35      0.41      0.34      3321
```

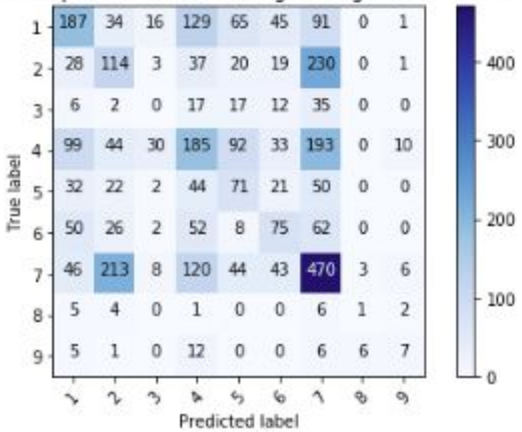
Confusion plot for 5 fold Strat. CV with RandomForest Gene Variation d2v




```
['d2v_with_gene_variation_rand_for.pkl']
```

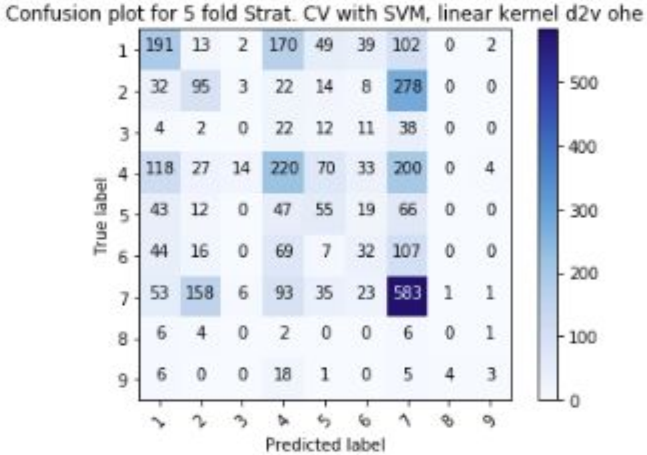
```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:    5.7s finished
Log loss:  2.11160969912
Accuracy:  0.3342366757
             precision    recall  f1-score   support

          1       0.41      0.33      0.36       568
          2       0.25      0.25      0.25       452
          3       0.00      0.00      0.00        89
          4       0.31      0.27      0.29       686
          5       0.22      0.29      0.25       242
          6       0.30      0.27      0.29       275
          7       0.41      0.49      0.45       953
          8       0.10      0.05      0.07        19
          9       0.26      0.19      0.22        37

avg / total       0.33      0.33      0.33      3321
```

Confusion plot for 5 fold Strat. logistic regression d2v ohe

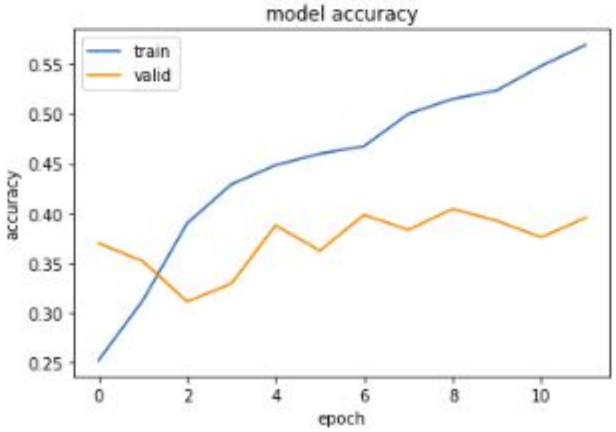



```
['logistic_regression_d2vohe.pkl']
```

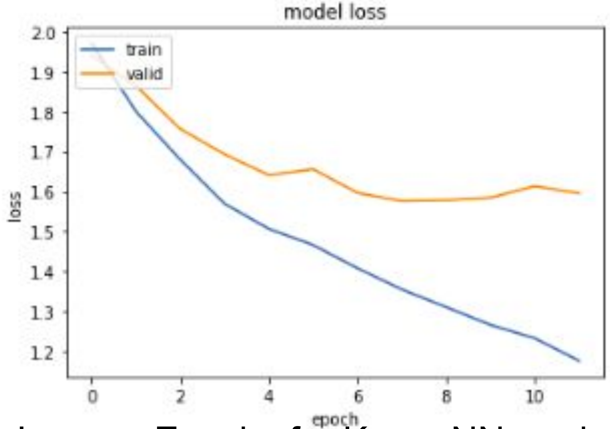
```
Cross validating model using 5-fold Stratified cross validation..
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   11.1s fin
Log loss:  1.74345471442
Accuracy:  0.355013550136
             precision    recall  f1-score   support

          1       0.38      0.34      0.36       568
          2       0.29      0.21      0.24       452
          3       0.00      0.00      0.00        89
          4       0.33      0.32      0.33       686
          5       0.23      0.23      0.23       242
          6       0.19      0.12      0.15       275
          7       0.42      0.61      0.50       953
          8       0.00      0.00      0.00        19
          9       0.27      0.08      0.12        37

avg / total       0.33      0.36      0.33      3321
```




```
['svm_lineard2vohe.pkl']
```

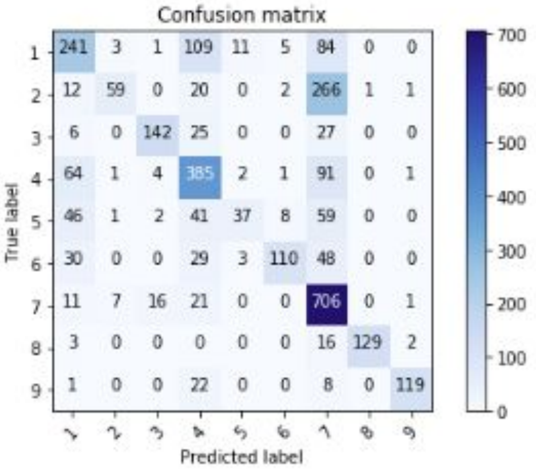


Accuracy vs Epochs for Keras NN model



Loss vs Epochs for Keras NN model

```
Cross validating model using 5-fold Stratified cross validation...
[Parallel(n_jobs=-1)]: Done   3 out of   3 | elapsed:   15.8s finished
Log loss:  1.21845721556
Accuracy:  0.634210526316
             precision    recall  f1-score   support

          1       0.58      0.53      0.56       454
          2       0.83      0.16      0.27       361
          3       0.86      0.71      0.78       200
          4       0.59      0.70      0.64       549
          5       0.70      0.19      0.30       194
          6       0.87      0.50      0.64       220
          7       0.54      0.93      0.68       762
          8       0.99      0.86      0.92       150
          9       0.96      0.79      0.87       150

avg / total       0.69      0.63      0.61      3040
```



On testing the performance of different models with doc2vec, we found that the performance did not improve with a great difference as such. Even the Keras Neural Networks model did not show satisfactory improvement. We plotted epoch vs training loss and validation loss, epoch vs training accuracy and validation accuracy to check if it was overfitting, but the graphs show that after 10 epochs validation log loss starts increasing.

However, oversampling on some of the classes helped.

Analysis and Conclusion

- Upsampling helped some of the classes with repeated sampling. So with random forest and upsampling ,we got the best results (with doc2vec as preprocessing technique) for cross-validation. [Accuracy: 0.63 and Log loss: 1.2, with Average Precision as 0.69 and Recall 0.63]
- This was because the initial distribution of classes was very skewed and required additional data for training.
- In conclusion, we tried an array of different models for text-preprocessing and performance evaluation, and found that the Random Forest Model (which uses bagging), is an efficient technique to avoid overfitting, Doc2Vec is a reasonable method for creating word embeddings of dense biological literature, and Oversampling is essential in the case of imbalanced classes with low number of samples for a few classes.
- However, the oversampling method we used is not very trustable since we have randomly chosen samples to create multiple copies, introducing high variance and overfitting (low bias).

References

-https://www.kaggle.com/c/msk-redefining-cancer-treatment

-http://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction

-https://www.kaggle.com/headsortails/personalised-medicine-eda-with-tidy-r

-https://www.kaggle.com/reiinakano/basic-nlp-bag-of-words-tf-idf-word2vec-lstm

-https://www.kaggle.com/dextrousjinx/brief-insight-on-genetic-variations

- Wikipedia, scikit-learn documentation of different models.

- Scikit Learn documentation
- http://linanqiu.github.io/2015/10/07/word2vec-sentiment/
- https://www.kaggle.com/c/word2vec-nlp-tutorial/discussion/12287
- https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
- https://towardsdatascience.com/a-gentle-introduction-to-doc2vec-db3e8c0cce5e
- https://rare-technologies.com/word2vec-in-python-part-two-optimizing/

- https://rare-technologies.com/doc2vec-tutorial/
- http://scikit-learn.org/stable/modules/ensemble.htm
- https://elitedatascience.com/imbalanced-classes
- https://www.analyticsvidhya.com/blog/2017/03/imbalanced-classification-problem/#
- https://www.kaggle.com/alyosama/doc2vec-with-keras-0-77