The Problem

Domain	Context	Problem Statement
Healthcare	Vaccine Adverse Event Reporting System (VAERS)	What might be the adverse effect post vaccination?

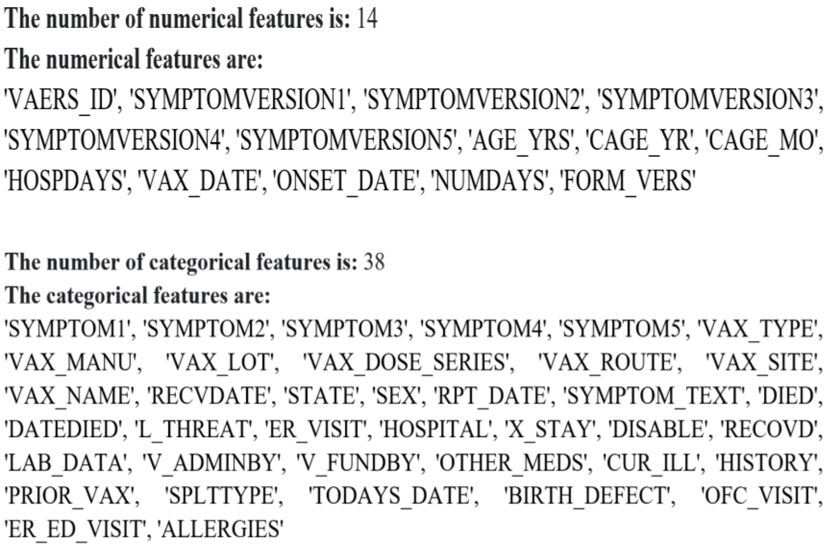
Data Dictionary

Features in VAERSVAX
VAERS_ID
VAX_TYPE
VAX_MANU
VAX_LOT
VAX_DOSE_SERIES
VAX_ROUTE
VAX_SITE
VAX_NAME

Features in VAERSSYMPTOMS
VAERS_ID
SYMPTOM1
SYMPTOMVERSION1
SYMPTOM2
SYMPTOMVERSION2
SYMPTOM3
SYMPTOMVERSION3
SYMPTOM4
SYMPTOMVERSION4
SYMPTOM5
SYMPTOMVERSION5

Features in VAERSDATA
DIED
DATEDIED
L_THREAT
ER_VISIT
HOSPITAL
HOSPDAYS
X_STAY
DISABLE
RECOVD
VAX_DATE
ONSET_DATE
NUMDAYS
LAB_DATA
V_ADMINBY
V_FUNDBY
OTHER_MEDS
CUR_ILL
HISTORY
PRIOR_VAX
SPLTTYPE
FORM_VERS
TODAYS_DATE
BIRTH_DEFECT
OFC_VISIT
ER_ED_VISIT
ALLERGIES

Merging data & Removal of duplicates



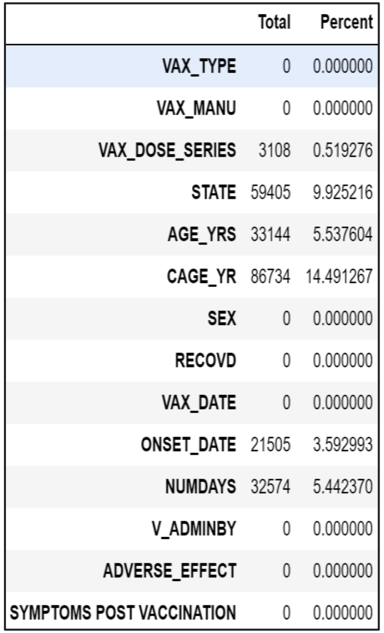
```
The number of numerical features is: 14
The numerical features are:
'VAERS_ID', 'SYMPTOMVERSION1', 'SYMPTOMVERSION2', 'SYMPTOMVERSION3',
'SYMPTOMVERSION4', 'SYMPTOMVERSION5', 'AGE_YRS', 'CAGE_YR', 'CAGE_MO',
'HOSPDAYS', 'VAX_DATE', 'ONSET_DATE', 'NUMDAYS', 'FORM_VERS'

The number of categorical features is: 38
The categorical features are:
'SYMPTOM1', 'SYMPTOM2', 'SYMPTOM3', 'SYMPTOM4', 'SYMPTOM5', 'VAX_TYPE',
'VAX_MANU', 'VAX_LOT', 'VAX_DOSE_SERIES', 'VAX_ROUTE', 'VAX_SITE',
'VAX_NAME', 'RECVDATE', 'STATE', 'SEX', 'RPT_DATE', 'SYMPTOM_TEXT', 'DIED',
'DATEDIED', 'L_THREAT', 'ER_VISIT', 'HOSPITAL', 'X_STAY', 'DISABLE', 'RECOVD',
'LAB_DATA', 'V_ADMINBY', 'V_FUNDBY', 'OTHER_MEDS', 'CUR_ILL', 'HISTORY',
'PRIOR_VAX', 'SPLTTYPE', 'TODAYS_DATE', 'BIRTH_DEFECT', 'OFC_VISIT',
'ER_ED_VISIT', 'ALLERGIES'
```

Dropping redundant features

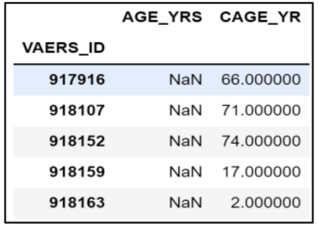
Feature Engineering

Missing Value Analysis & Treatment

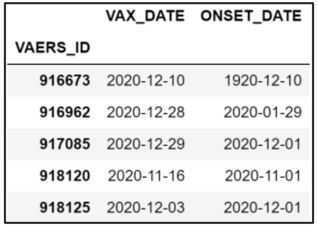


	Total	Percent
VAX_TYPE	0	0.000000
VAX_MANU	0	0.000000
VAX_DOSE_SERIES	3108	0.519276
STATE	59405	9.925216
AGE_YRS	33144	5.537604
CAGE_YR	86734	14.491267
SEX	0	0.000000
RECOVD	0	0.000000
VAX_DATE	0	0.000000
ONSET_DATE	21505	3.592993
NUMDAYS	32574	5.442370
V_ADMINBY	0	0.000000
ADVERSE_EFFECT	0	0.000000
SYMPTOMS POST VACCINATION	0	0.000000

Replacing missing values in 'AGE_YRS' with corresponding values in 'CAGE_YR'



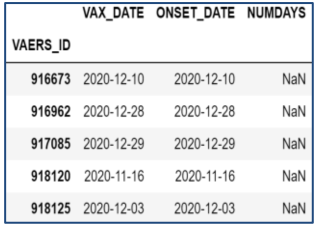
	AGE_YRS	CAGE_YR
VAERS_ID		
917916	NaN	66.000000
918107	NaN	71.000000
918152	NaN	74.000000
918159	NaN	17.000000
918163	NaN	2.000000



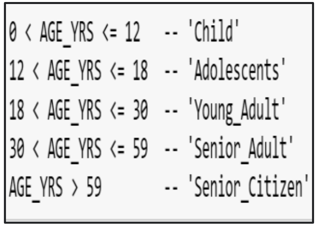
	VAX_DATE	ONSET_DATE
VAERS_ID		
916673	2020-12-10	1920-12-10
916962	2020-12-28	2020-01-29
917085	2020-12-29	2020-12-01
918120	2020-11-16	2020-11-01
918125	2020-12-03	2020-12-01

Replacing ONSET_DATE lesser than VAX_DATE with VAX_DATE

Imputing Missing Values in 'NUMDAYS' based on 'VAX_DATE' & 'ONSET_DATE'

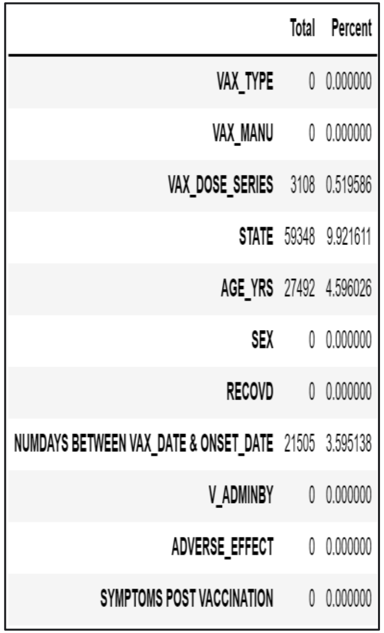


	VAX_DATE	ONSET_DATE	NUMDAYS
VAERS_ID			
916673	2020-12-10	2020-12-10	NaN
916962	2020-12-28	2020-12-28	NaN
917085	2020-12-29	2020-12-29	NaN
918120	2020-11-16	2020-11-16	NaN
918125	2020-12-03	2020-12-03	NaN



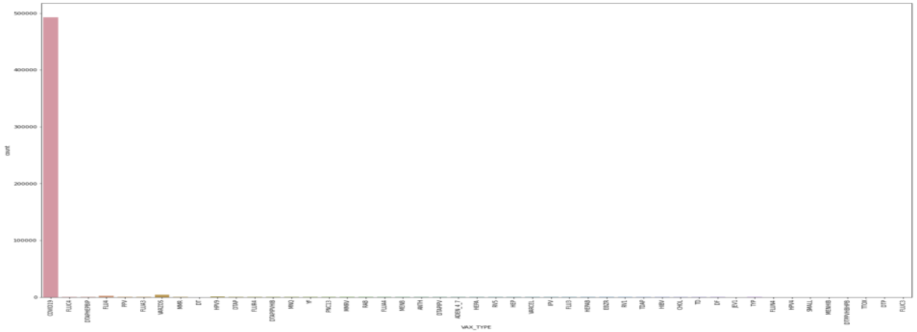
```
0 < AGE_YRS <= 12   -- 'Child'
12 < AGE_YRS <= 18  -- 'Adolescents'
18 < AGE_YRS <= 30  -- 'Young_Adult'
30 < AGE_YRS <= 59  -- 'Senior_Adult'
AGE_YRS > 59        -- 'Senior_Citizen'
```

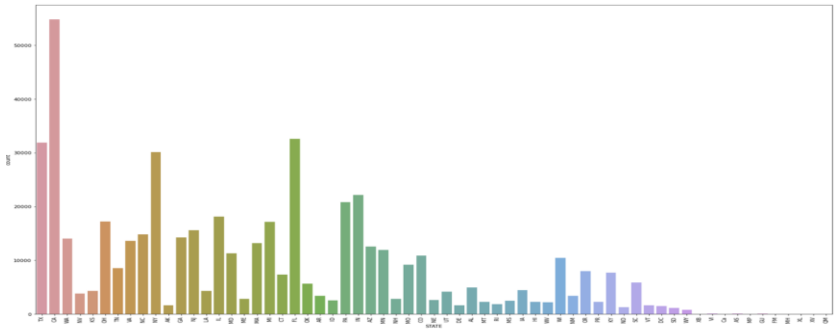
Bucketing of 'AGE_YRS' variable into 'AGE_GROUP'

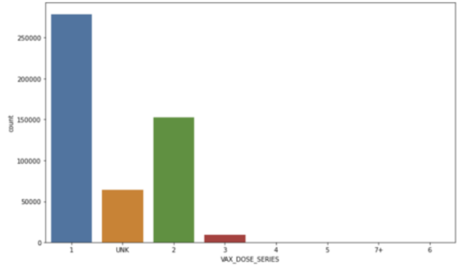


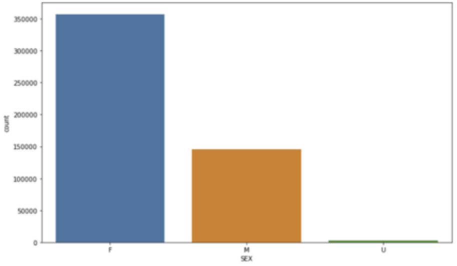
	Total	Percent
VAX_TYPE	0	0.000000
VAX_MANU	0	0.000000
VAX_DOSE_SERIES	3108	0.519586
STATE	59348	9.921611
AGE_YRS	27492	4.596026
SEX	0	0.000000
RECOVD	0	0.000000
NUMDAYS BETWEEN VAX_DATE & ONSET_DATE	21505	3.595138
V_ADMINBY	0	0.000000
ADVERSE_EFFECT	0	0.000000
SYMPTOMS POST VACCINATION	0	0.000000

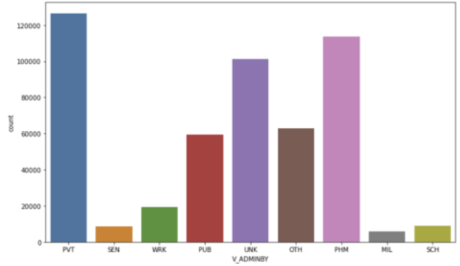
UniVariate Analysis

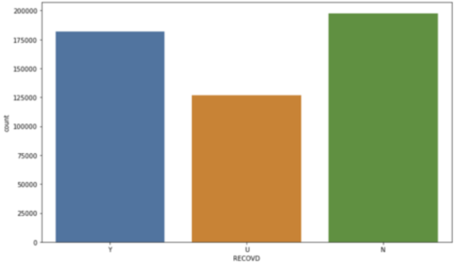


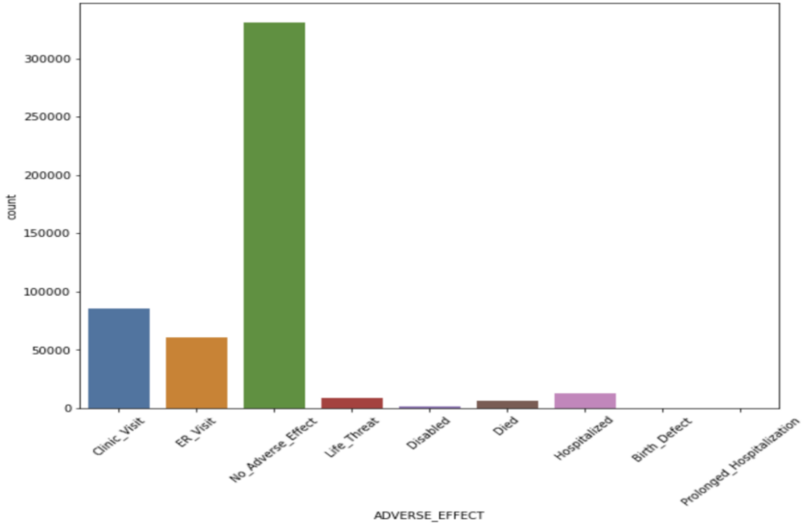




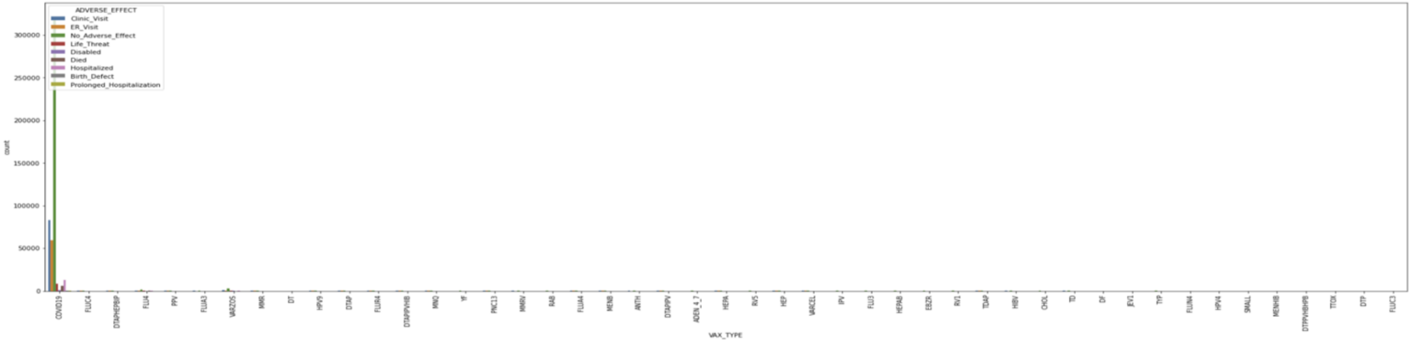


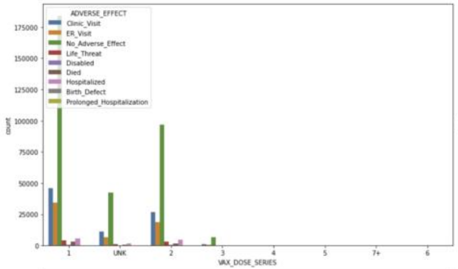


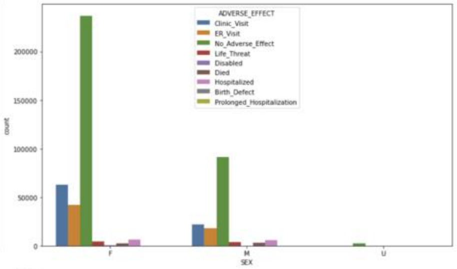


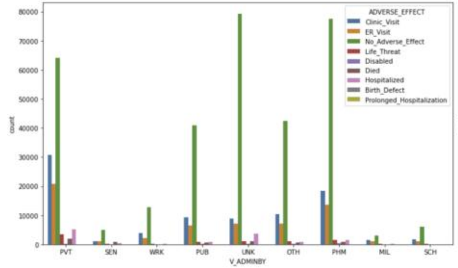


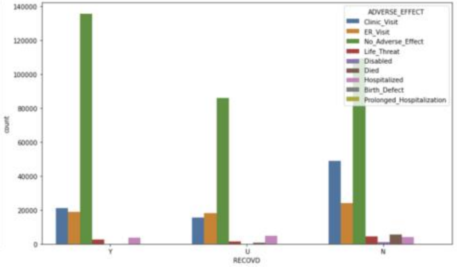
BiVariate Analysis



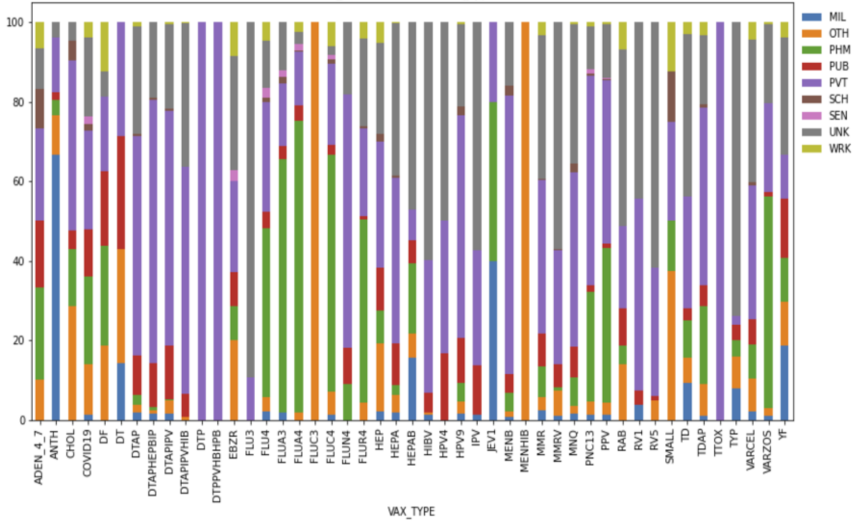








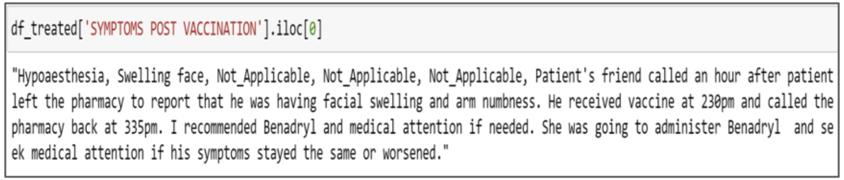
V_ADMINBY Vs Vaccine_Type



Natural Language Processing

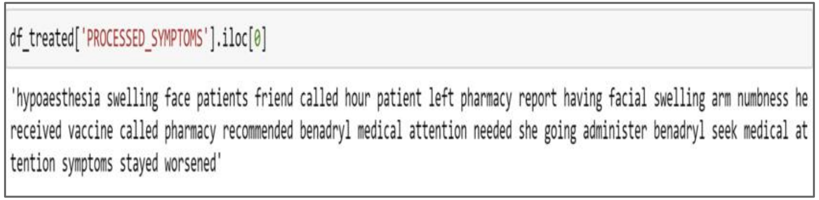
NLP on 'SYMPTOMS POST VACCINATION' feature:

The data in this feature was in an unstructured format which cannot be used while building the model. Hence, the data in the feature has to be pre processed.



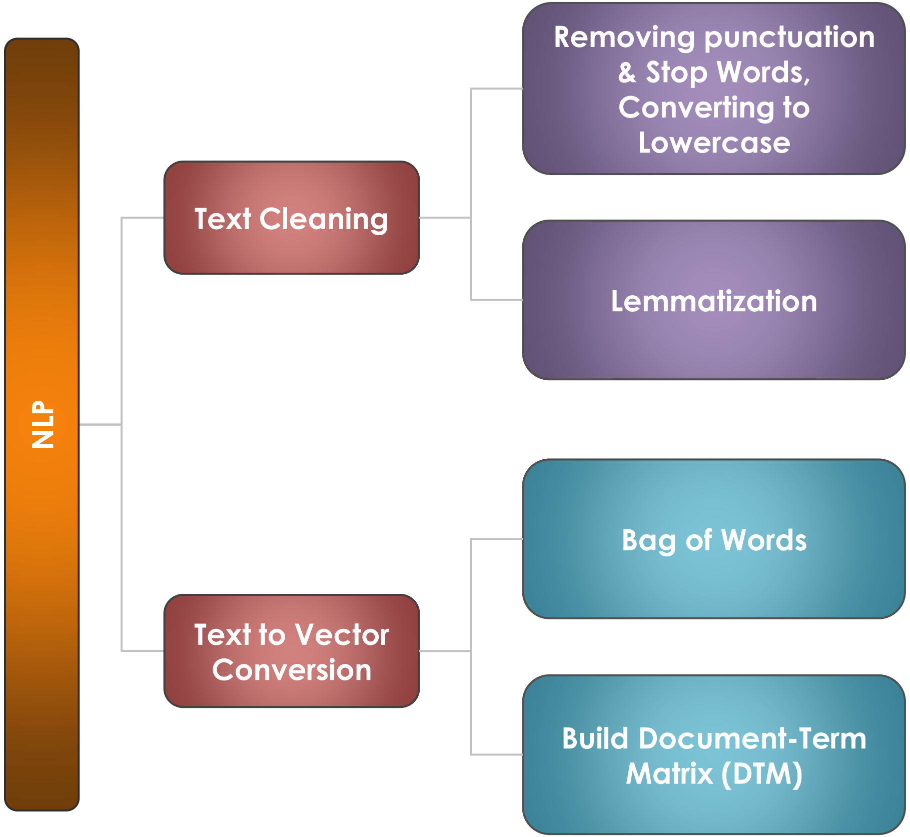
```
df_treated['SYMPTOMS POST VACCINATION'].iloc[0]
```

```
"Hypoaesthesia, Swelling face, Not_Applicable, Not_Applicable, Not_Applicable, Patient's friend called an hour after patient
left the pharmacy to report that he was having facial swelling and arm numbness. He received vaccine at 230pm and called the
pharmacy back at 335pm. I recommended Benadryl and medical attention if needed. She was going to administer Benadryl  and se
ek medical attention if his symptoms stayed the same or worsened."
```



```
df_treated['PROCESSED_SYMPTOMS'].iloc[0]
```

```
'hypoaesthesia swelling face patients friend called hour patient left pharmacy report having facial swelling arm numbness he
received vaccine called pharmacy recommended benadryl medical attention needed she going administer benadryl seek medical at
tention symptoms stayed worsened'
```



NLP

Text Cleaning

Removing punctuation & Stop Words, Converting to Lowercase

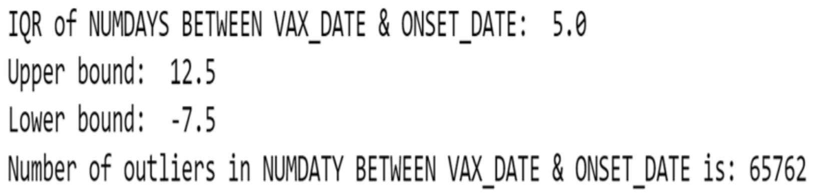
Lemmatization

Text to Vector Conversion

Bag of Words

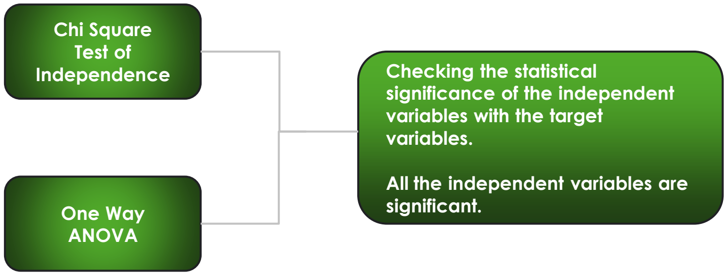
Build Document-Term Matrix (DTM)

Outlier Analysis



```
IQR of NUMDAYS BETWEEN VAX_DATE & ONSET_DATE:  5.0
Upper bound:  12.5
Lower bound:  -7.5
Number of outliers in NUMDATY BETWEEN VAX_DATE & ONSET_DATE is: 65762
```

Statistical Significance



Chi Square Test of Independence

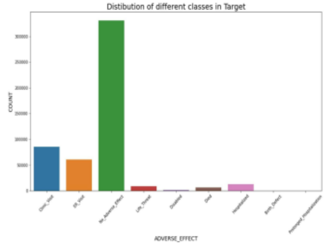
One Way ANOVA

Checking the statistical significance of the independent variables with the target variables.

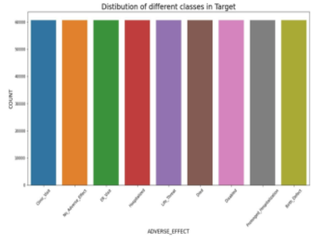
All the independent variables are significant.

Class Imbalance Treatment

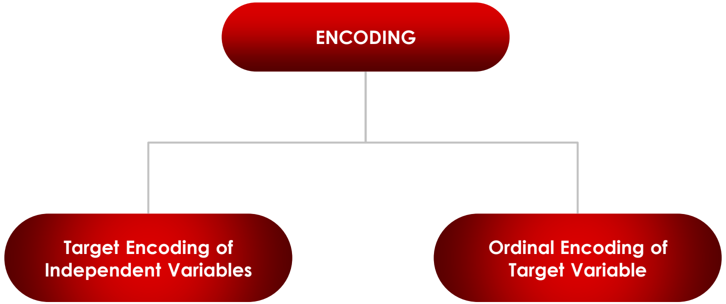
Before Sampling



After Sampling



Encoding

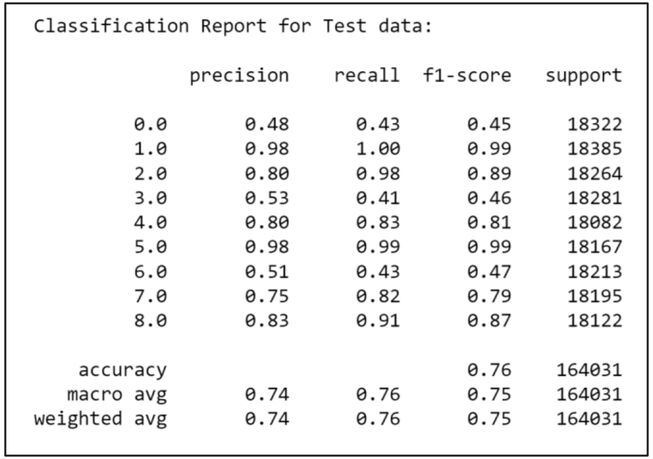


ENCODING

Target Encoding of Independent Variables

Ordinal Encoding of Target Variable

Base Model Analysis without Text Feature



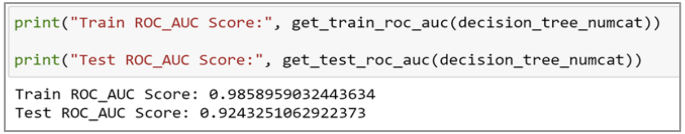
```
Classification Report for Test data:

              precision    recall  f1-score   support

         0.0       0.48      0.43      0.45     18322
         1.0       0.98      1.00      0.99     18385
         2.0       0.80      0.98      0.89     18264
         3.0       0.53      0.41      0.46     18281
         4.0       0.80      0.83      0.81     18082
         5.0       0.98      0.99      0.99     18167
         6.0       0.51      0.43      0.47     18213
         7.0       0.75      0.82      0.79     18195
         8.0       0.83      0.91      0.87     18122

    accuracy                           0.76    164031
   macro avg       0.74      0.76      0.75    164031
weighted avg       0.74      0.76      0.75    164031
```

ROC_AUC score:

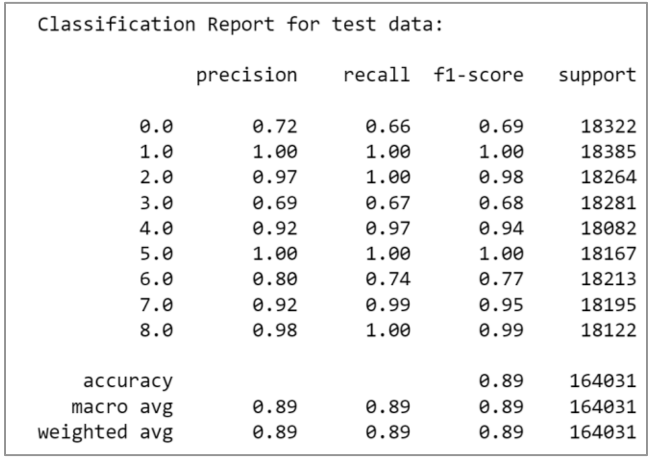


```
print("Train ROC_AUC Score:", get_train_roc_auc(decision_tree_numcat))

print("Test ROC_AUC Score:", get_test_roc_auc(decision_tree_numcat))
```

```
Train ROC_AUC Score: 0.9858959032443634
Test ROC_AUC Score: 0.9243251062922373
```

Base Model Analysis for Text Feature



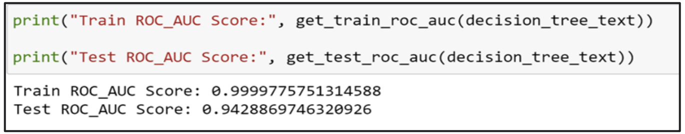
```
Classification Report for test data:

              precision    recall  f1-score   support

         0.0       0.72      0.66      0.69     18322
         1.0       1.00      1.00      1.00     18385
         2.0       0.97      1.00      0.98     18264
         3.0       0.69      0.67      0.68     18281
         4.0       0.92      0.97      0.94     18082
         5.0       1.00      1.00      1.00     18167
         6.0       0.80      0.74      0.77     18213
         7.0       0.92      0.99      0.95     18195
         8.0       0.98      1.00      0.99     18122

    accuracy                           0.89    164031
   macro avg       0.89      0.89      0.89    164031
weighted avg       0.89      0.89      0.89    164031
```

ROC_AUC score:

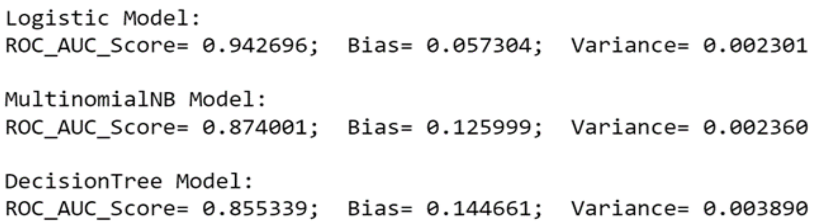


```
print("Train ROC_AUC Score:", get_train_roc_auc(decision_tree_text))

print("Test ROC_AUC Score:", get_test_roc_auc(decision_tree_text))
```

```
Train ROC_AUC Score: 0.9999775751314588
Test ROC_AUC Score: 0.9428869746320926
```

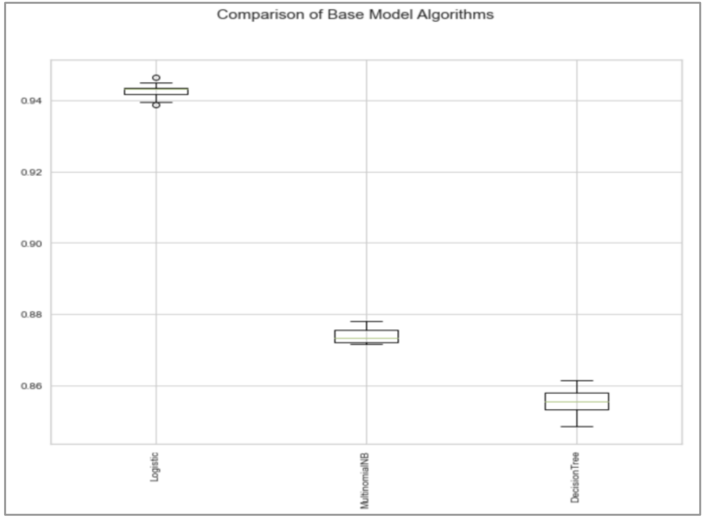
Comparative Analysis Of Base Model Algorithms



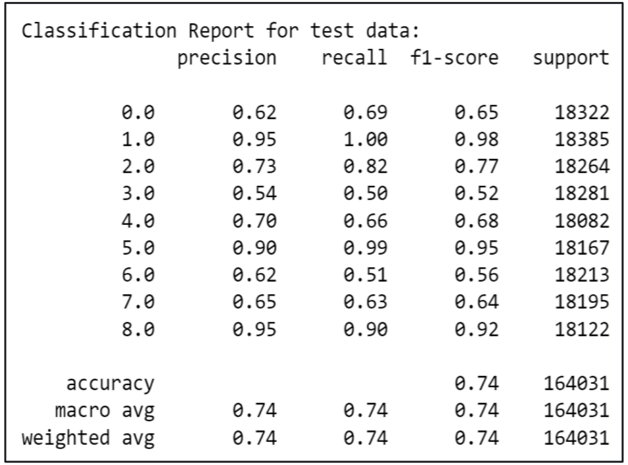
```
Logistic Model:
ROC_AUC_Score= 0.942696;  Bias= 0.057304;  Variance= 0.002301

MultinomialNB Model:
ROC_AUC_Score= 0.874001;  Bias= 0.125999;  Variance= 0.002360

DecisionTree Model:
ROC_AUC_Score= 0.855339;  Bias= 0.144661;  Variance= 0.003890
```



Full Base Model - Logistic Regression

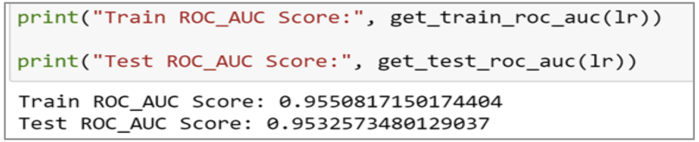


```
Classification Report for test data:
              precision    recall  f1-score   support

         0.0       0.62      0.69      0.65     18322
         1.0       0.95      1.00      0.98     18385
         2.0       0.73      0.82      0.77     18264
         3.0       0.54      0.50      0.52     18281
         4.0       0.70      0.66      0.68     18082
         5.0       0.90      0.99      0.95     18167
         6.0       0.62      0.51      0.56     18213
         7.0       0.65      0.63      0.64     18195
         8.0       0.95      0.90      0.92     18122

    accuracy                           0.74    164031
   macro avg       0.74      0.74      0.74    164031
weighted avg       0.74      0.74      0.74    164031
```

ROC_AUC score:

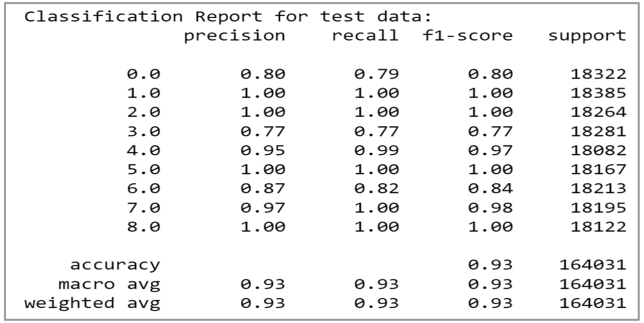


```
print("Train ROC_AUC Score:", get_train_roc_auc(lr))

print("Test ROC_AUC Score:", get_test_roc_auc(lr))
```

```
Train ROC_AUC Score: 0.9550817150174404
Test ROC_AUC Score: 0.9532573480129037
```

Random Forest Classifier

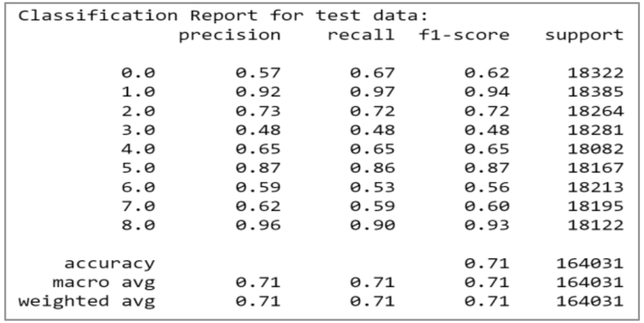


```
Classification Report for test data:
              precision    recall  f1-score   support

         0.0       0.80      0.79      0.80     18322
         1.0       1.00      1.00      1.00     18385
         2.0       1.00      1.00      1.00     18264
         3.0       0.77      0.77      0.77     18281
         4.0       0.95      0.99      0.97     18082
         5.0       1.00      1.00      1.00     18167
         6.0       0.87      0.82      0.84     18213
         7.0       0.97      1.00      0.98     18195
         8.0       1.00      1.00      1.00     18122

    accuracy                           0.93    164031
   macro avg       0.93      0.93      0.93    164031
weighted avg       0.93      0.93      0.93    164031
```

Gradient Boost Classifier

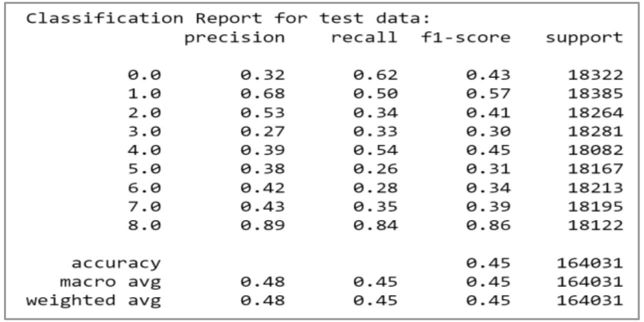


```
Classification Report for test data:
              precision    recall  f1-score   support

         0.0       0.57      0.67      0.62     18322
         1.0       0.92      0.97      0.94     18385
         2.0       0.73      0.72      0.72     18264
         3.0       0.48      0.48      0.48     18281
         4.0       0.65      0.65      0.65     18082
         5.0       0.87      0.86      0.87     18167
         6.0       0.59      0.53      0.56     18213
         7.0       0.62      0.59      0.60     18195
         8.0       0.96      0.90      0.93     18122

    accuracy                           0.71    164031
   macro avg       0.71      0.71      0.71    164031
weighted avg       0.71      0.71      0.71    164031
```

Adaboost Classifier

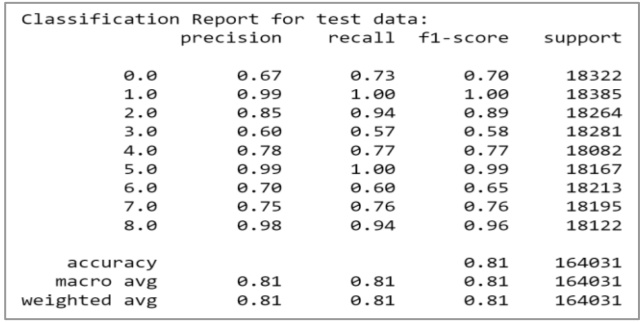


```
Classification Report for test data:
              precision    recall  f1-score   support

         0.0       0.32      0.62      0.43     18322
         1.0       0.68      0.50      0.57     18385
         2.0       0.53      0.34      0.41     18264
         3.0       0.27      0.33      0.30     18281
         4.0       0.39      0.54      0.45     18082
         5.0       0.38      0.26      0.31     18167
         6.0       0.42      0.28      0.34     18213
         7.0       0.43      0.35      0.39     18195
         8.0       0.89      0.84      0.86     18122

    accuracy                           0.45    164031
   macro avg       0.48      0.45      0.45    164031
weighted avg       0.48      0.45      0.45    164031
```

XGBoost Classifier



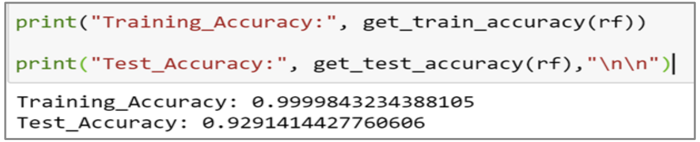
```
Classification Report for test data:
              precision    recall  f1-score   support

         0.0       0.67      0.73      0.70     18322
         1.0       0.99      1.00      1.00     18385
         2.0       0.85      0.94      0.89     18264
         3.0       0.60      0.57      0.58     18281
         4.0       0.78      0.77      0.77     18082
         5.0       0.99      1.00      0.99     18167
         6.0       0.70      0.60      0.65     18213
         7.0       0.75      0.76      0.76     18195
         8.0       0.98      0.94      0.96     18122

    accuracy                           0.81    164031
   macro avg       0.81      0.81      0.81    164031
weighted avg       0.81      0.81      0.81    164031
```

Ensemble Model - Random Forest Classifier

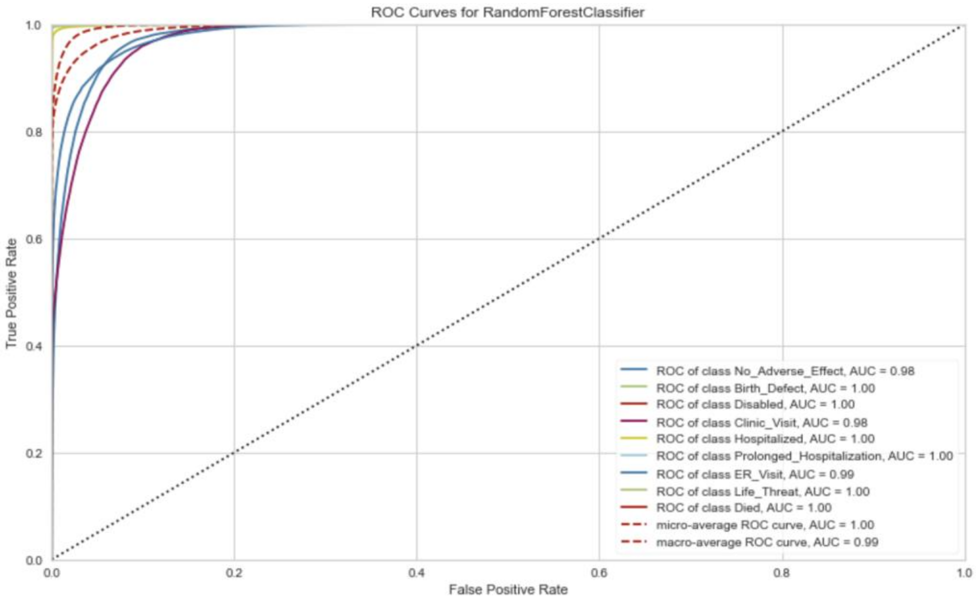
Accuracy:

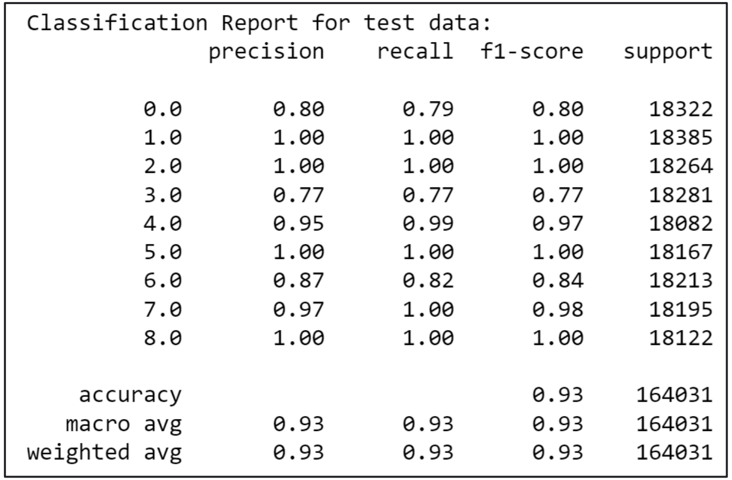


```
print("Training_Accuracy:", get_train_accuracy(rf))

print("Test_Accuracy:", get_test_accuracy(rf),"\n\n")
```

```
Training_Accuracy: 0.9999843234388105
Test_Accuracy: 0.9291414427760606
```



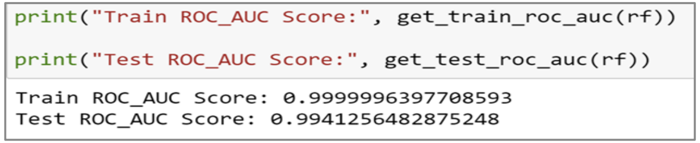


```
Classification Report for test data:
              precision    recall  f1-score   support

         0.0       0.80      0.79      0.80     18322
         1.0       1.00      1.00      1.00     18385
         2.0       1.00      1.00      1.00     18264
         3.0       0.77      0.77      0.77     18281
         4.0       0.95      0.99      0.97     18082
         5.0       1.00      1.00      1.00     18167
         6.0       0.87      0.82      0.84     18213
         7.0       0.97      1.00      0.98     18195
         8.0       1.00      1.00      1.00     18122

    accuracy                           0.93    164031
   macro avg       0.93      0.93      0.93    164031
weighted avg       0.93      0.93      0.93    164031
```

ROC_AUC score:

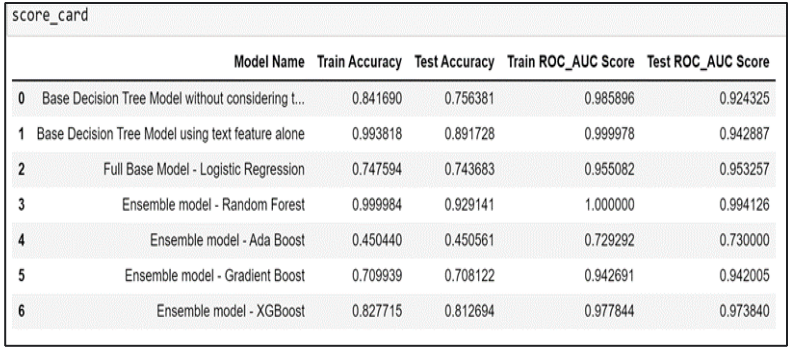


```
print("Train ROC_AUC Score:", get_train_roc_auc(rf))

print("Test ROC_AUC Score:", get_test_roc_auc(rf))
```

```
Train ROC_AUC Score: 0.9999996397708593
Test ROC_AUC Score: 0.9941256482875248
```

Choosing Final Model and Deployment

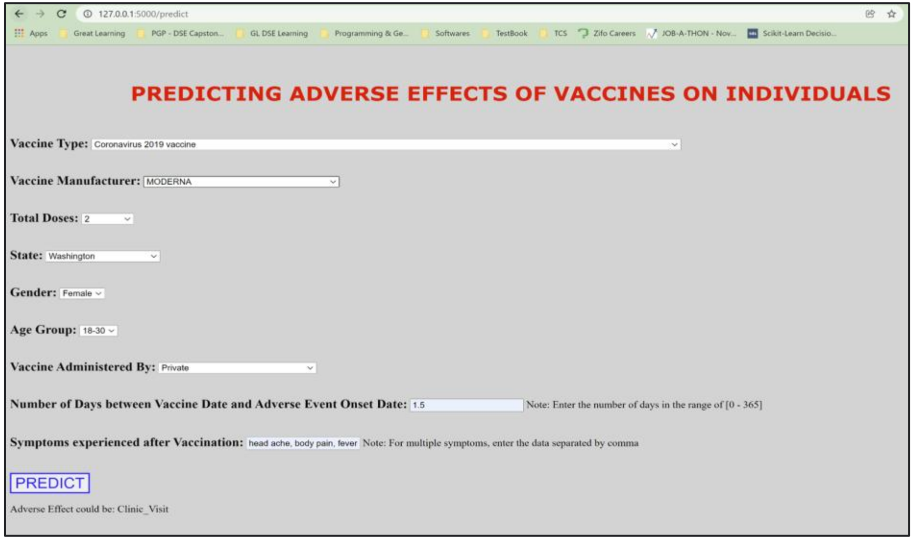


score_card

	Model Name	Train Accuracy	Test Accuracy	Train ROC_AUC Score	Test ROC_AUC Score
0	Base Decision Tree Model without considering t...	0.841690	0.756381	0.985896	0.924325
1	Base Decision Tree Model using text feature alone	0.993818	0.891728	0.999978	0.942887
2	Full Base Model - Logistic Regression	0.747594	0.743683	0.955082	0.953257
3	Ensemble model - Random Forest	0.999984	0.929141	1.000000	0.994126
4	Ensemble model - Ada Boost	0.450440	0.450561	0.729292	0.730000
5	Ensemble model - Gradient Boost	0.709939	0.708122	0.942691	0.942005
6	Ensemble model - XGBoost	0.827715	0.812694	0.977844	0.973840

From the analysis done on various classification model algorithms, we could see that **Random Forest Classifier** with default hyper parameters shows better performance with higher accuracy and ROC_AUC scores than all other models.

Hence, it was chosen as our Final Model and the model was deployed using a web application framework - **FLASK** having a HTML based web UI.



PREDICTING ADVERSE EFFECTS OF VACCINES ON INDIVIDUALS

Vaccine Type: Coronavirus 2019 vaccine

Vaccine Manufacturer: MODERNA

Total Doses: 2

State: Washington

Gender: Female

Age Group: 18-30

Vaccine Administered By: Private

Number of Days between Vaccine Date and Adverse Event Onset Date: 1.5 Note: Enter the number of days in the range of [0 - 365]

Symptoms experienced after Vaccination: head ache, body pain, fever Note: For multiple symptoms, enter the data separated by comma

Adverse Effect could be: Clinic_Visit