CS 513
Knowledge Discovery and Data Mining

Prudential Life Insurance Risk Assessment

Team Members

Parth Parab

CWID: 10444835

Varsha R

CWID: 10444838

Aryan Anmol

CWID: 10450866

Supriti Garnaik

CWID: 10444348

Problem Statement

https://www.kaggle.com/c/prudential-life-insurance-assessment/data

The process of life insurance purchase is complicated and involves risk analysis at multiple levels and various medical exams.

Prudential wants to make the process quick less labor intensive and automated.

By using predictive modeling we are classifying the risk profiles of the customer based on their information.

Data Set Explanation

The Data set (train.csv) has 59K Rows and 127 Columns.

The data set from Prudential has the information about customer’s personal details and their insurance preference.

The task is to classify the risk profile of the customer.

Data fields

Variable	Description
Id	A unique identifier associated with an application.
Product_Info_1-7	A set of normalized variables relating to the product applied for
Ins_Age	Normalized age of applicant
Ht	Normalized height of applicant
Wt	Normalized weight of applicant
BMI	Normalized BMI of applicant
Employment_Info_1-6	A set of normalized variables relating to the employment history of the applicant.
InsuredInfo_1-6	A set of normalized variables providing information about the applicant.
Insurance_History_1-9	A set of normalized variables relating to the insurance history of the applicant.
Family_Hist_1-5	A set of normalized variables relating to the family history of the applicant.
Medical_History_1-41	A set of normalized variables relating to the medical history of the applicant.
Medical_Keyword_1-48	A set of dummy variables relating to the presence of/absence of a medical keyword being associated with the application.
Response	This is the target variable, an ordinal variable relating to the final decision associated with an application

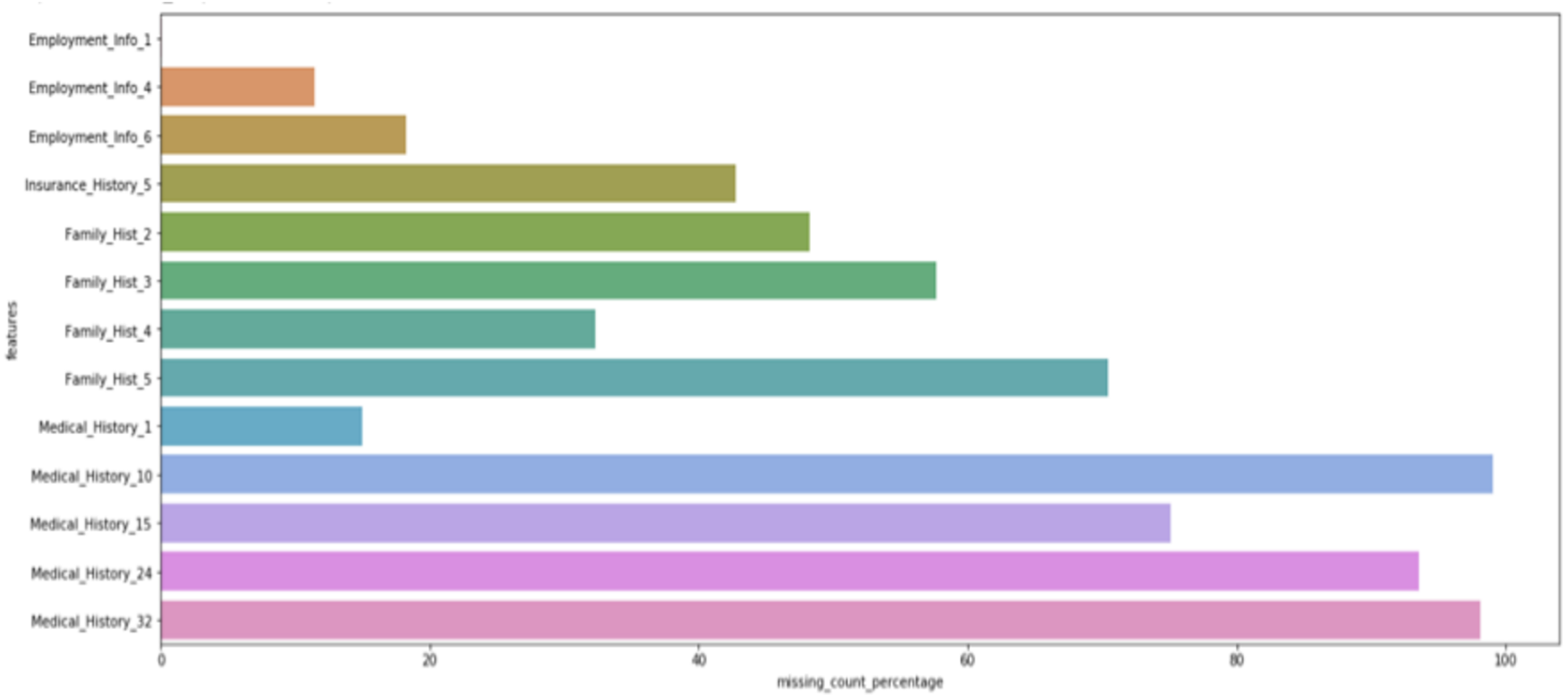
Preprocessing

We replaced the missing entities from the data set with the column means of the data set.

We also removed specific columns based on relevance of the data

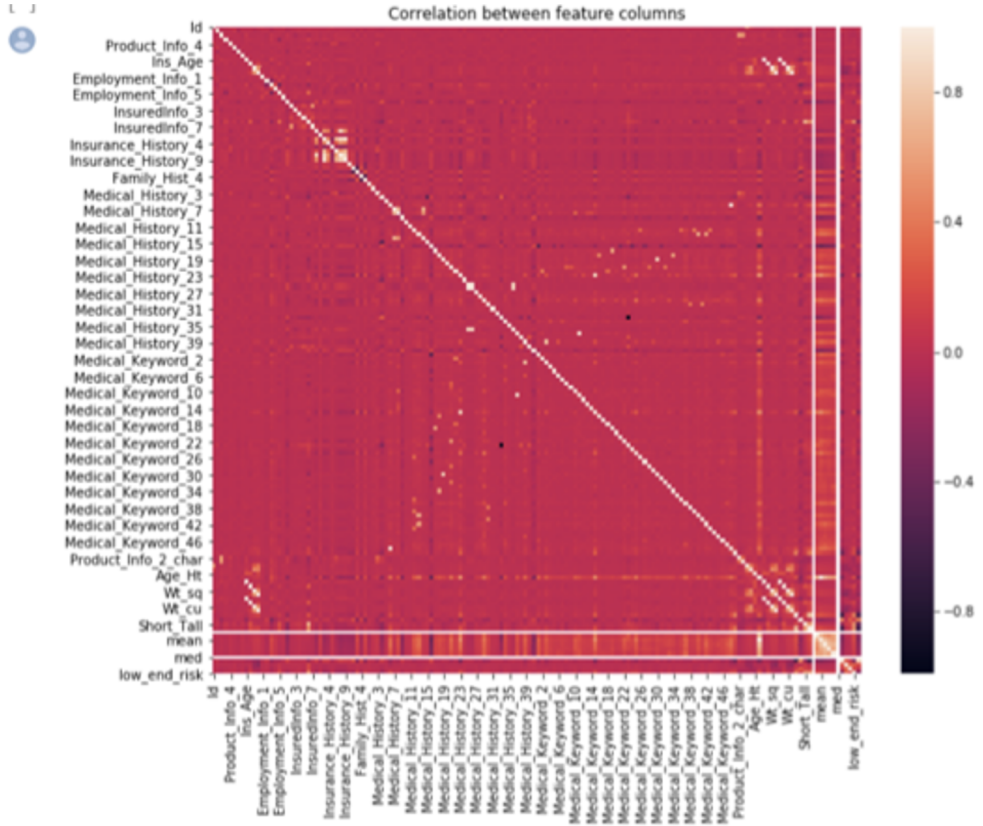
We Picked up top 75 Columns based on feature importance and missing features in training set

Percentage of missing feature values in the train set

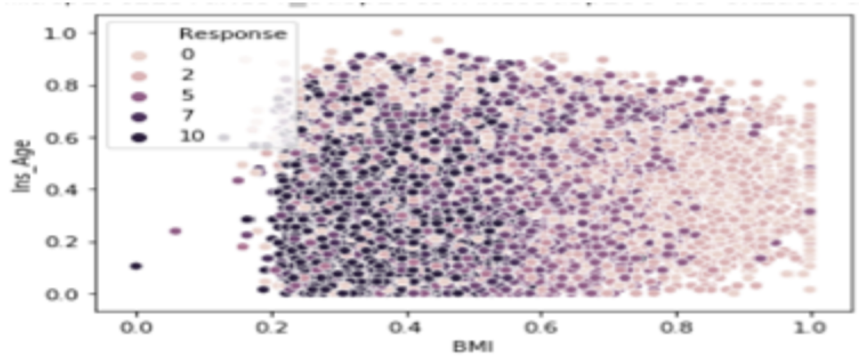


Correlation between the feature Columns

Heat map representation to show case how different features columns are correlated to each to determine the relevance.

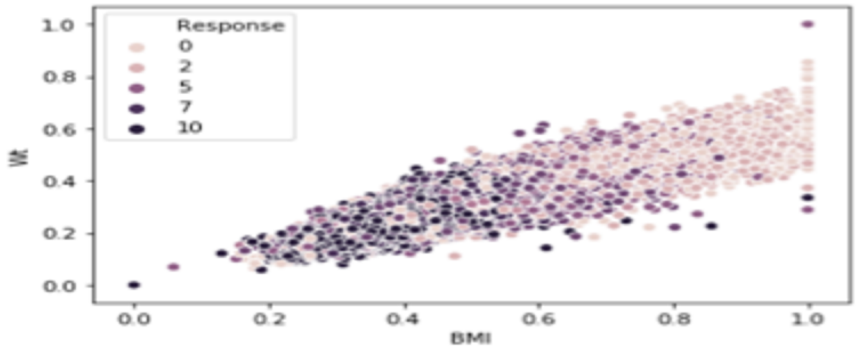


Scatter plot of key feature columns in relation to response

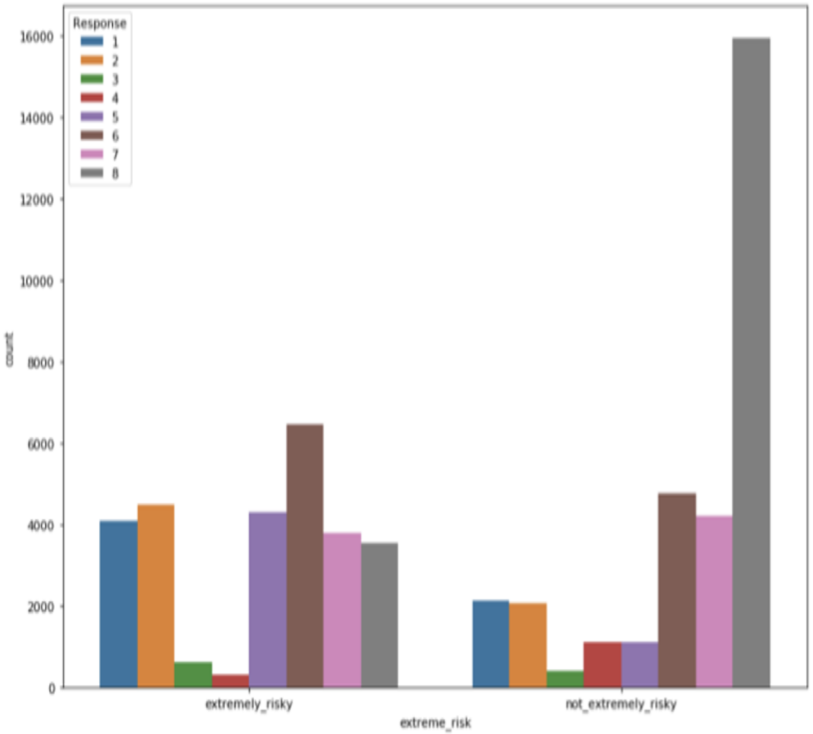



```
[ ]  sns.scatterplot(data=df_all,x='BMI',y='Wt',hue='Response',alpha=1)
```

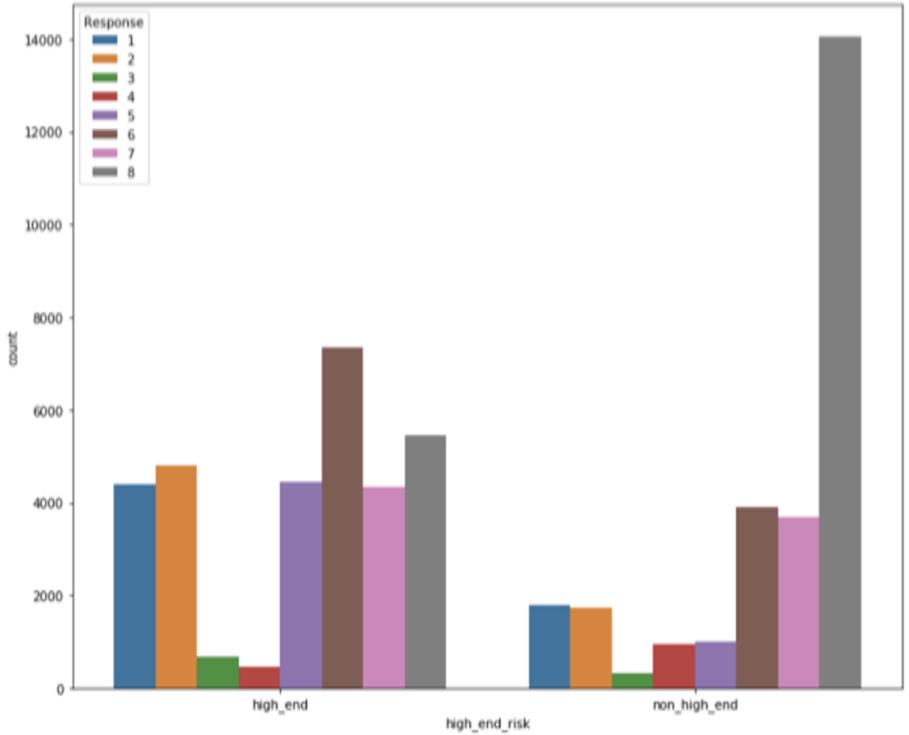
```
<matplotlib.axes._subplots.AxesSubplot at 0x132172510>
```



Response Profile Comparison

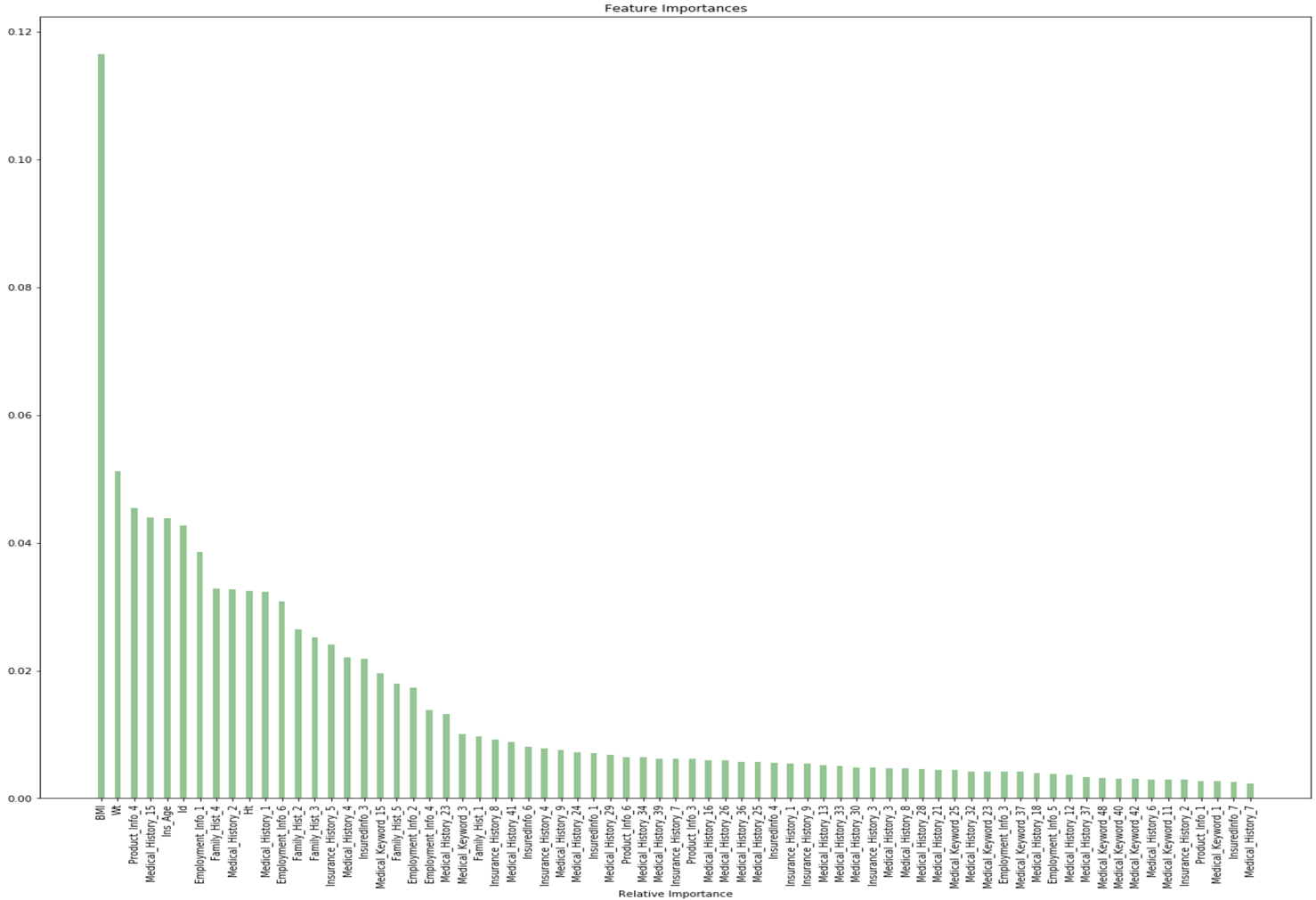


Based on weight



Based on age/weight/height

Analysis feature label Chart using important feature selection


Feature Importances
0.12
0.10
0.08
0.06
0.04
0.02
0.00
BMI
Wt
Product_Info_4
Medical_History_15
Ins_Age
Id
Employment_Info_1
Family_Hist_4
Medical_History_2
Ht
Medical_History_1
Employment_Info_6
Family_Hist_2
Family_Hist_3
Insurance_History_5
Medical_History_4
InsuredInfo_3
Medical_Keyword_15
Family_Hist_5
Employment_Info_2
Employment_Info_4
Medical_History_23
Medical_Keyword_3
Family_Hist_1
Insurance_History_8
Medical_History_41
InsuredInfo_6
Insurance_History_4
Medical_History_9
Medical_History_24
InsuredInfo_1
Medical_History_29
Product_Info_6
Medical_History_34
Medical_History_39
Insurance_History_7
Product_Info_3
Medical_History_16
Medical_History_26
Medical_History_36
Medical_History_25
InsuredInfo_4
Insurance_History_1
Insurance_History_9
Medical_History_13
Medical_History_33
Medical_History_30
Insurance_History_3
Medical_History_3
Medical_History_8
Medical_History_28
Medical_History_21
Medical_Keyword_25
Medical_History_32
Medical_Keyword_23
Employment_Info_3
Medical_Keyword_37
Medical_History_18
Employment_Info_5
Medical_History_12
Medical_History_37
Medical_Keyword_48
Medical_Keyword_40
Medical_Keyword_42
Medical_History_6
Medical_Keyword_11
Insurance_History_2
Product_Info_1
Medical_Keyword_1
InsuredInfo_7
Medical_History_7
Relative Importance

Classification Algorithms

Different classification algorithms used and their results

BGC Classifier

Decision Tree Classifier

K-Nearest Neighbour

Multinomial Naive Bayes

Random Forest

Random Forest with Grid Search

XG Boosting

XG Boosting with Grid Search

BGC Classifier

	precision	recall	f1-score	support
1	0.30	0.21	0.25	1879
2	0.33	0.23	0.27	1959
3	0.53	0.41	0.46	301
4	0.63	0.58	0.60	431
5	0.57	0.51	0.54	1662
6	0.48	0.52	0.50	3379
7	0.42	0.41	0.42	2333
8	0.69	0.84	0.76	5871
accuracy			0.55	17815
macro avg	0.49	0.46	0.47	17815
weighted avg	0.52	0.55	0.53	17815

Decision Tree classifier

	precision	recall	f1-score	support
1	0.24	0.22	0.23	1879
2	0.22	0.22	0.22	1959
3	0.34	0.33	0.33	301
4	0.52	0.48	0.50	431
5	0.41	0.40	0.41	1662
6	0.40	0.42	0.41	3379
7	0.31	0.33	0.32	2333
8	0.68	0.67	0.67	5871
accuracy			0.44	17815
macro avg	0.39	0.38	0.39	17815
weighted avg	0.45	0.44	0.44	17815

K-Nearest Neighbor classifier

	precision	recall	f1-score	support
1	0.13	0.14	0.13	1879
2	0.11	0.10	0.10	1959
3	0.22	0.09	0.13	301
4	0.27	0.12	0.17	431
5	0.11	0.07	0.09	1662
6	0.19	0.18	0.18	3379
7	0.13	0.11	0.12	2333
8	0.37	0.45	0.40	5871
accuracy			0.23	17815
macro avg	0.19	0.16	0.17	17815
weighted avg	0.22	0.23	0.22	17815

Multinomial Naïve Bayes

	precision	recall	f1-score	support
1	0.24	0.09	0.13	1879
2	0.12	0.35	0.18	1959
3	0.18	0.53	0.27	301
4	0.24	0.31	0.27	431
5	0.18	0.03	0.05	1662
6	0.21	0.10	0.13	3379
7	0.26	0.09	0.14	2333
8	0.41	0.51	0.45	5871
accuracy			0.26	17815
macro avg	0.23	0.25	0.20	17815
weighted avg	0.27	0.26	0.24	17815

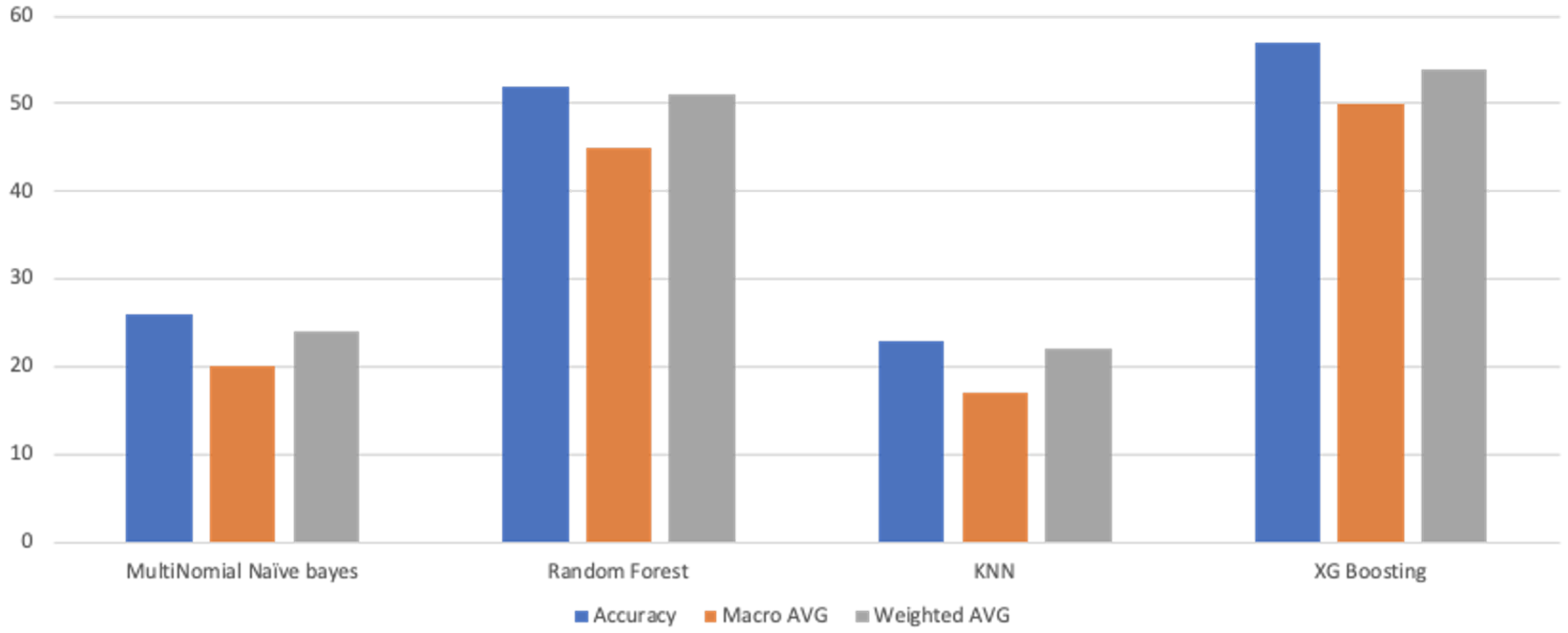
Random Forest without Grid Search

	precision	recall	f1-score	support
1	0.28	0.22	0.25	1879
2	0.30	0.22	0.25	1959
3	0.52	0.38	0.44	301
4	0.63	0.57	0.60	431
5	0.53	0.45	0.48	1662
6	0.43	0.49	0.46	3379
7	0.42	0.32	0.36	2333
8	0.69	0.84	0.76	5871
accuracy			0.52	17815
macro avg	0.47	0.44	0.45	17815
weighted avg	0.50	0.52	0.51	17815

XG Boosting without Grid Search

	precision	recall	f1-score	support
1	0.30	0.21	0.25	1879
2	0.33	0.23	0.27	1959
3	0.53	0.41	0.46	301
4	0.63	0.58	0.60	431
5	0.57	0.51	0.54	1662
6	0.48	0.52	0.50	3379
7	0.42	0.41	0.42	2333
8	0.69	0.84	0.76	5871
accuracy			0.55	17815
macro avg	0.49	0.46	0.47	17815
weighted avg	0.52	0.55	0.53	17815

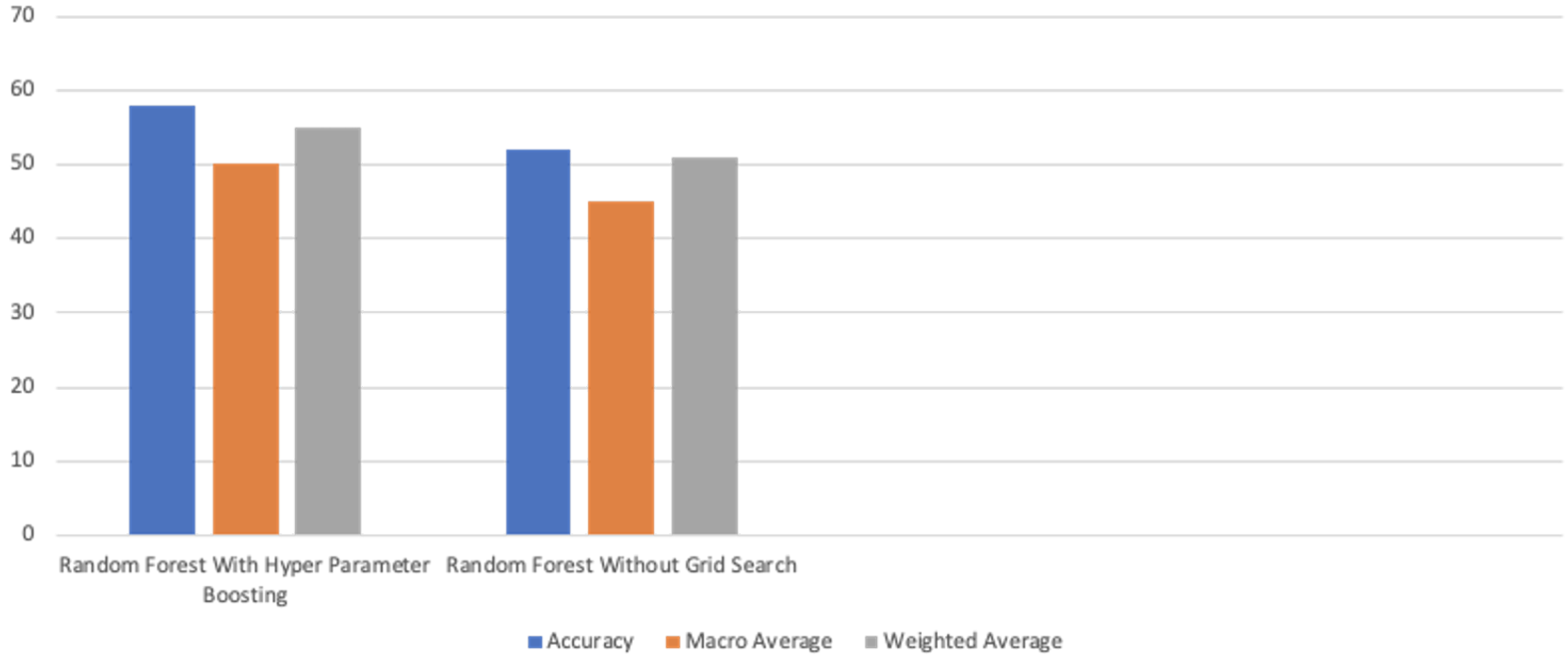
Accuracy Comparison



Random Forest with Hyper Parameter Boosting

	precision	recall	f1-score	support
1	0.46	0.18	0.26	1879
2	0.47	0.23	0.31	1959
3	0.60	0.40	0.48	301
4	0.62	0.68	0.65	431
5	0.64	0.53	0.58	1662
6	0.49	0.60	0.54	3379
7	0.48	0.39	0.43	2333
8	0.67	0.92	0.77	5871
accuracy			0.58	17815
macro avg	0.55	0.49	0.50	17815
weighted avg	0.56	0.58	0.55	17815

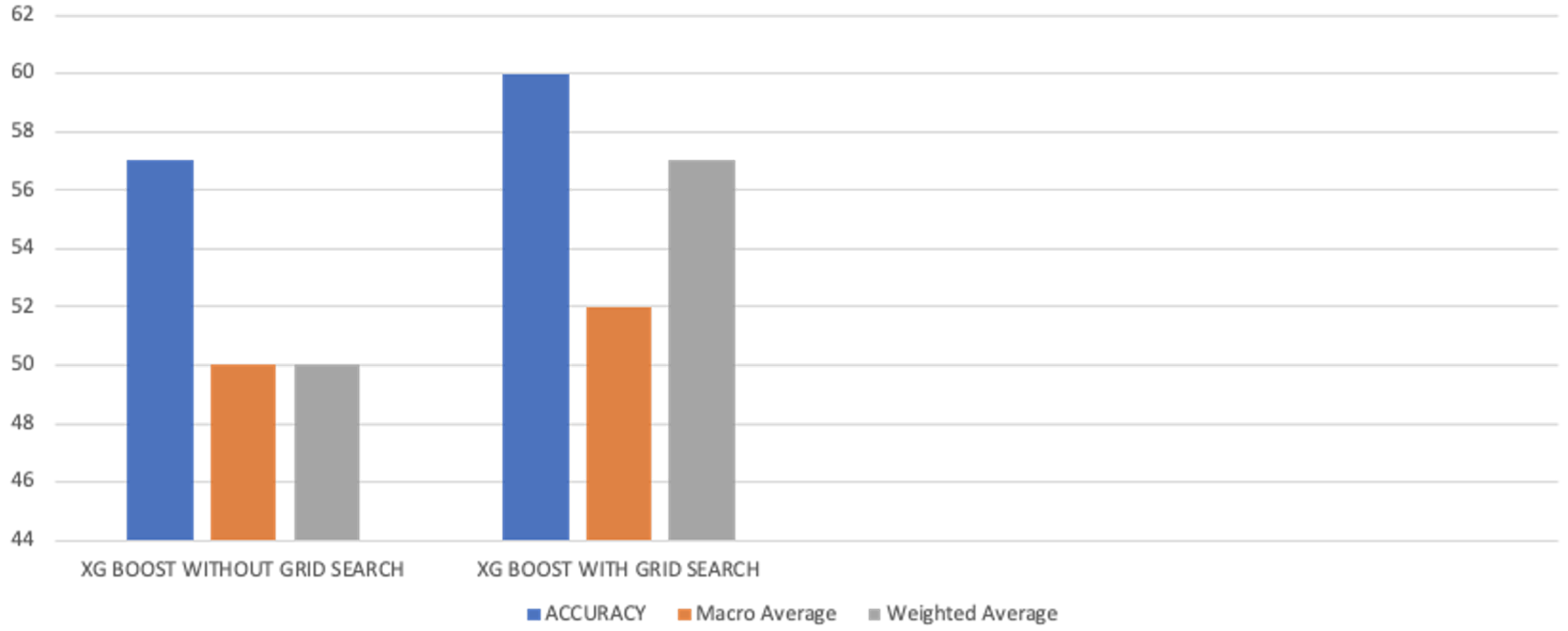
Random Forest Accuracy Comparison



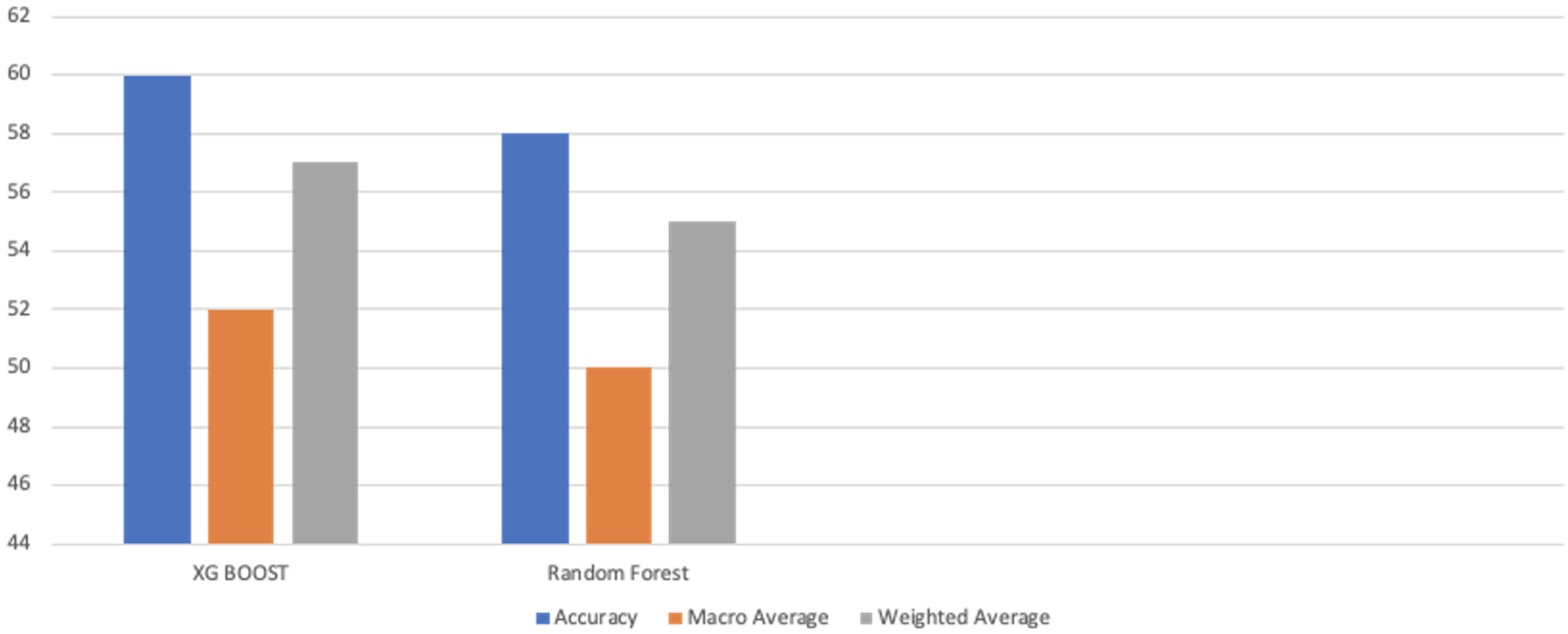
XG Boost with Grid Search

	precision	recall	f1-score	support
1	0.48	0.25	0.33	1879
2	0.45	0.25	0.32	1959
3	0.62	0.41	0.49	301
4	0.64	0.68	0.66	431
5	0.66	0.54	0.59	1662
6	0.54	0.57	0.55	3379
7	0.48	0.46	0.47	2333
8	0.68	0.92	0.78	5871
accuracy			0.60	17815
macro avg	0.57	0.51	0.52	17815
weighted avg	0.58	0.60	0.57	17815

XG Boost Accuracy Comparison



XG Boosting Vs Random Forest
Using Grid Search in both the cases



In Conclusion

XG Boost with grid search performed well across all algorithms including xgboost without grid search