

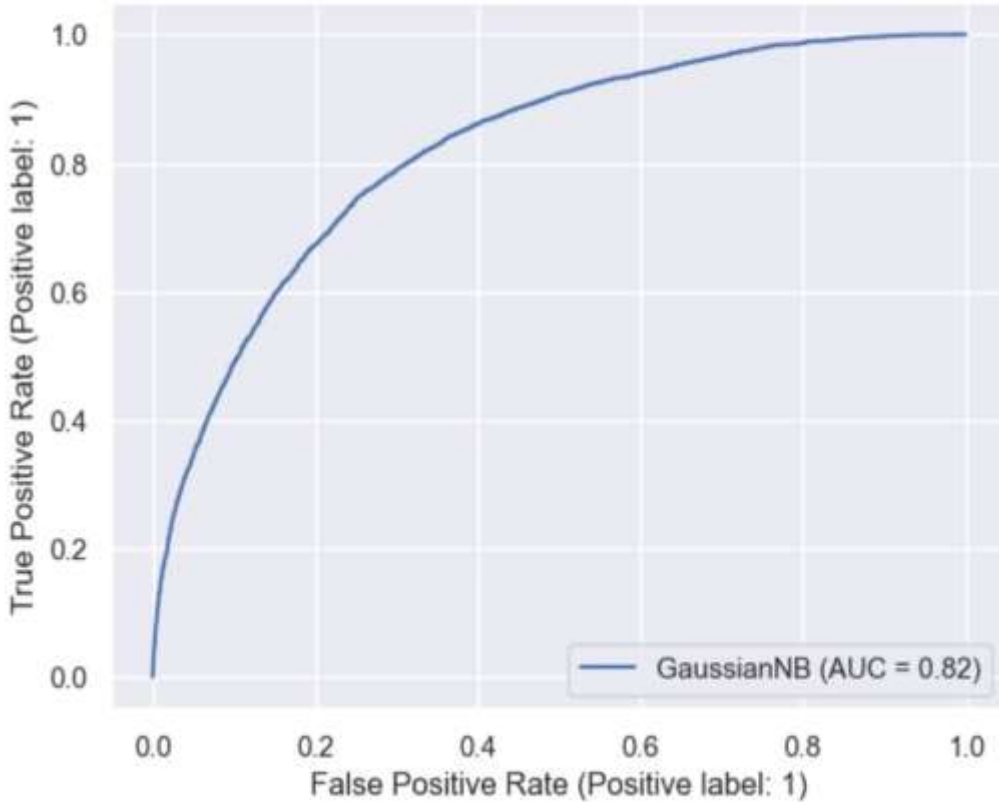
Project Development Phase : Model Performance Test

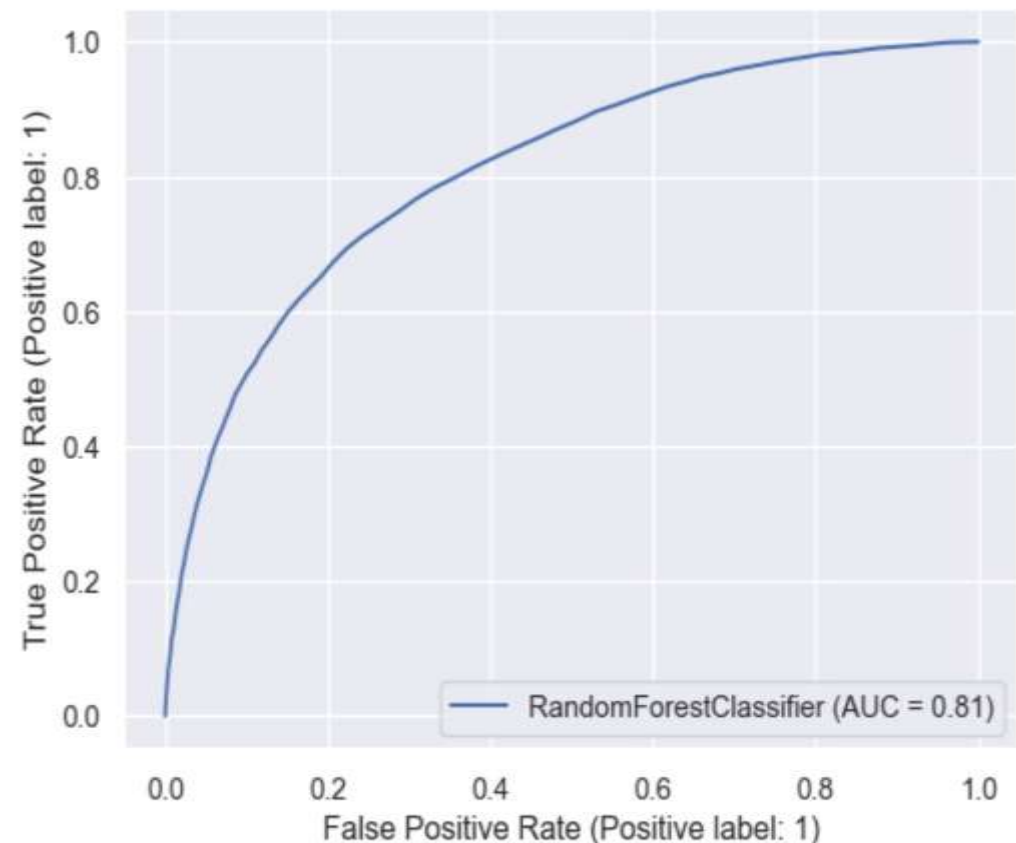
Date	07 November 2022
Team ID	PNT2022TMID40045
Project Name	Exploratory Analysis of RainFall Data in India for Agriculture.
Maximum Marks	4 Marks

Model Performance Testing:

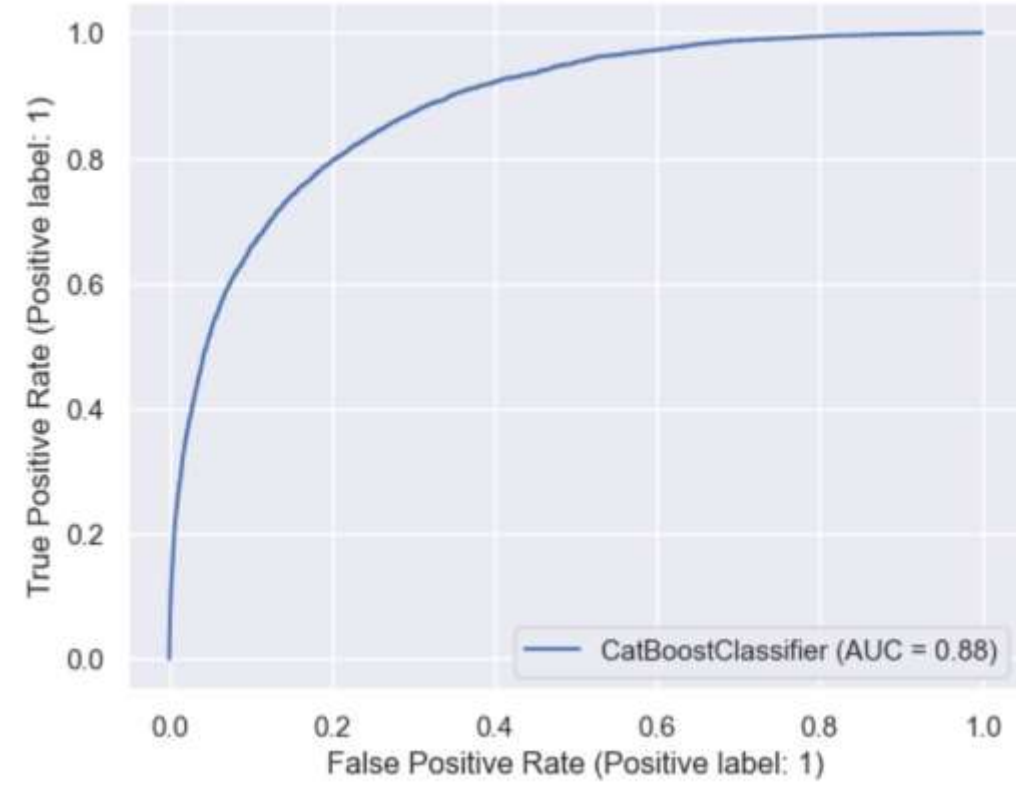
Sl. No.	Parameter	Values [Classification Model]	Screenshot
---------	-----------	-------------------------------	------------

1.		<div>Logistic Regression</div> <div>Confusion Matrix – <code>[[17634 5083]</code> <code>[1467 4908]]</code></div> <div>Accuracy Score- 0.7748521930427609</div> <div>& Classification Report –</div> <table><tr><th></th><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.92</td><td>0.78</td><td>0.84</td><td>22717</td><td></td></tr><tr><td>1</td><td>0.49</td><td>0.77</td><td>0.60</td><td>6375</td><td></td></tr><tr><td></td><td>accuracy</td><td></td><td></td><td>0.77</td><td>29092</td></tr><tr><td></td><td>macro avg</td><td>0.71</td><td>0.77</td><td>0.72</td><td>29092</td></tr><tr><td></td><td>avg</td><td>0.83</td><td>0.77</td><td>0.79</td><td>29092</td></tr><tr><td></td><td>weighted avg</td><td></td><td></td><td></td><td></td></tr></table> <div><code>metrics.plot_roc_curve(lr, X_test3, y_test3)</code> <code>metrics.roc_auc_score(y_test3, y_pred3, average = None)</code> 0.7730646082617578</div>			precision	recall	f1-score	support	0	0.92	0.78	0.84	22717		1	0.49	0.77	0.60	6375			accuracy			0.77	29092		macro avg	0.71	0.77	0.72	29092		avg	0.83	0.77	0.79	29092		weighted avg					<pre>y_pred3 = lr.predict(X_test3) print(confusion_matrix(y_test3, y_pred3)) print() print(accuracy_score(y_test3, y_pred3)) print() print(classification_report(y_test3, y_pred3))</pre> <div><code>[[17634 5083]</code> <code>[1467 4908]]</code> 0.7748521930427609</div> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.92</td><td>0.78</td><td>0.84</td><td>22717</td></tr><tr><td>1</td><td>0.49</td><td>0.77</td><td>0.60</td><td>6375</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.77</td><td>29092</td></tr><tr><td>macro avg</td><td>0.71</td><td>0.77</td><td>0.72</td><td>29092</td></tr><tr><td>weighted avg</td><td>0.83</td><td>0.77</td><td>0.79</td><td>29092</td></tr></table> <div><code>metrics.plot_roc_curve(lr, X_test3, y_test3)</code> <code>metrics.roc_auc_score(y_test3, y_pred3, average = None)</code> 0.7730646082617578</div>		precision	recall	f1-score	support	0	0.92	0.78	0.84	22717	1	0.49	0.77	0.60	6375	accuracy			0.77	29092	macro avg	0.71	0.77	0.72	29092	weighted avg	0.83	0.77	0.79	29092
		precision	recall	f1-score	support																																																																						
0	0.92	0.78	0.84	22717																																																																							
1	0.49	0.77	0.60	6375																																																																							
	accuracy			0.77	29092																																																																						
	macro avg	0.71	0.77	0.72	29092																																																																						
	avg	0.83	0.77	0.79	29092																																																																						
	weighted avg																																																																										
	precision	recall	f1-score	support																																																																							
0	0.92	0.78	0.84	22717																																																																							
1	0.49	0.77	0.60	6375																																																																							
accuracy			0.77	29092																																																																							
macro avg	0.71	0.77	0.72	29092																																																																							
weighted avg	0.83	0.77	0.79	29092																																																																							

2.		<div><p><i>Gaussian Naïve Bayes</i></p><p>Confusion Matrix – <code>[[17179 5538]</code> <code>[1700 4675]]</code></p><p>Accuracy Score- 0.7512030798845043</p><p>& Classification Report –</p><table><tr><th></th><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td></td><td>0.91</td><td>0.76</td><td>0.83</td><td>22717</td></tr><tr><td>1</td><td></td><td>0.46</td><td>0.73</td><td>0.56</td><td>6375</td></tr><tr><td></td><td>accuracy</td><td></td><td></td><td>0.75</td><td>29092</td></tr><tr><td></td><td>macro avg</td><td>0.68</td><td>0.74</td><td>0.69</td><td>29092</td></tr><tr><td></td><td>weighted avg</td><td>0.81</td><td>0.75</td><td>0.77</td><td>29092</td></tr></table></div>			precision	recall	f1-score	support	0		0.91	0.76	0.83	22717	1		0.46	0.73	0.56	6375		accuracy			0.75	29092		macro avg	0.68	0.74	0.69	29092		weighted avg	0.81	0.75	0.77	29092	<div><pre>y_pred1 = gnb.predict(X_test1) print(confusion_matrix(y_test1, y_pred1)) print() print(accuracy_score(y_test1, y_pred1)) print() print(classification_report(y_test1, y_pred1))</pre><p><code>[[17179 5538]</code> <code>[1700 4675]]</code></p><p>0.7512030798845043</p><table><tr><th></th><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td></td><td>0.91</td><td>0.76</td><td>0.83</td><td>22717</td></tr><tr><td>1</td><td></td><td>0.46</td><td>0.73</td><td>0.56</td><td>6375</td></tr><tr><td></td><td>accuracy</td><td></td><td></td><td>0.75</td><td>29092</td></tr><tr><td></td><td>macro avg</td><td>0.68</td><td>0.74</td><td>0.69</td><td>29092</td></tr><tr><td></td><td>weighted avg</td><td>0.81</td><td>0.75</td><td>0.77</td><td>29092</td></tr></table></div> <div><pre>metrics.plot_roc_curve(gnb, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre><p>0.7447755718918284</p></div>			precision	recall	f1-score	support	0		0.91	0.76	0.83	22717	1		0.46	0.73	0.56	6375		accuracy			0.75	29092		macro avg	0.68	0.74	0.69	29092		weighted avg	0.81	0.75	0.77	29092
		precision	recall	f1-score	support																																																																						
0		0.91	0.76	0.83	22717																																																																						
1		0.46	0.73	0.56	6375																																																																						
	accuracy			0.75	29092																																																																						
	macro avg	0.68	0.74	0.69	29092																																																																						
	weighted avg	0.81	0.75	0.77	29092																																																																						
		precision	recall	f1-score	support																																																																						
0		0.91	0.76	0.83	22717																																																																						
1		0.46	0.73	0.56	6375																																																																						
	accuracy			0.75	29092																																																																						
	macro avg	0.68	0.74	0.69	29092																																																																						
	weighted avg	0.81	0.75	0.77	29092																																																																						

3.	Metrics	<div>Random Forest Classifier</div> <div>Confusion Matrix – <code>[[20552 2165]</code> <code>[2470 3905]]</code></div> <div>Accuracy Score- 0.8406778495806407</div> <div>& Classification Report –</div> <table><thead><tr><th></th><th>recall</th><th>f1-score</th><th>support</th><th>precision</th></tr></thead><tbody><tr><td>0</td><td>0.89</td><td>0.90</td><td>0.90</td><td>22717</td></tr><tr><td>1</td><td>0.64</td><td>0.61</td><td>0.63</td><td>6375</td></tr></tbody></table> <div><table><thead><tr><th></th><th>accuracy</th><th></th><th>0.84</th><th>29092</th></tr></thead><tbody><tr><td>macro avg</td><td>0.77</td><td>0.76</td><td>0.76</td><td>29092</td></tr><tr><td>weighted avg</td><td>0.84</td><td>0.84</td><td>0.84</td><td>29092</td></tr></tbody></table></div>		recall	f1-score	support	precision	0	0.89	0.90	0.90	22717	1	0.64	0.61	0.63	6375		accuracy		0.84	29092	macro avg	0.77	0.76	0.76	29092	weighted avg	0.84	0.84	0.84	29092	<div><pre>y_pred3 = rfClassifier.predict(X_test3) print(confusion_matrix(y_test3, y_pred3)) print() print(accuracy_score(y_test3, y_pred3)) print() print(classification_report(y_test3, y_pred3))</pre></div> <div><code>[[20552 2165]</code> <code>[2470 3905]]</code></div> <div>0.8406778495806407</div> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.89</td><td>0.90</td><td>0.90</td><td>22717</td></tr><tr><td>1</td><td>0.64</td><td>0.61</td><td>0.63</td><td>6375</td></tr></tbody></table> <div><table><thead><tr><th></th><th>accuracy</th><th></th><th>0.84</th><th>29092</th></tr></thead><tbody><tr><td>macro avg</td><td>0.77</td><td>0.76</td><td>0.76</td><td>29092</td></tr><tr><td>weighted avg</td><td>0.84</td><td>0.84</td><td>0.84</td><td>29092</td></tr></tbody></table></div>		precision	recall	f1-score	support	0	0.89	0.90	0.90	22717	1	0.64	0.61	0.63	6375		accuracy		0.84	29092	macro avg	0.77	0.76	0.76	29092	weighted avg	0.84	0.84	0.84	29092	<div><pre>metrics.plot_roc_curve(rfClassifier, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre></div> <div>0.6840454561540247</div> 
	recall	f1-score	support	precision																																																												
0	0.89	0.90	0.90	22717																																																												
1	0.64	0.61	0.63	6375																																																												
	accuracy		0.84	29092																																																												
macro avg	0.77	0.76	0.76	29092																																																												
weighted avg	0.84	0.84	0.84	29092																																																												
	precision	recall	f1-score	support																																																												
0	0.89	0.90	0.90	22717																																																												
1	0.64	0.61	0.63	6375																																																												
	accuracy		0.84	29092																																																												
macro avg	0.77	0.76	0.76	29092																																																												
weighted avg	0.84	0.84	0.84	29092																																																												

4.		<div><div>XGBoost Classifier</div><div>Confusion Matrix – [[21308 1409] [2925 3450]]</div><div>Accuracy Score- 0.851024336587378</div><div>& Classification Report –<table><tr><th></th><th>recall</th><th>f1-score</th><th>support</th><th>precision</th><th></th></tr><tr><td>0</td><td>0.88</td><td>0.94</td><td>0.91</td><td>22717</td><td></td></tr><tr><td>1</td><td>0.71</td><td>0.54</td><td>0.61</td><td>6375</td><td></td></tr><tr><td>accuracy</td><td></td><td></td><td></td><td>0.85</td><td>29092</td></tr><tr><td>macro avg</td><td>0.79</td><td>0.74</td><td>0.76</td><td></td><td>29092</td></tr><tr><td>weighted avg</td><td>0.84</td><td>0.85</td><td>0.84</td><td></td><td>29092</td></tr></table></div></div>		recall	f1-score	support	precision		0	0.88	0.94	0.91	22717		1	0.71	0.54	0.61	6375		accuracy				0.85	29092	macro avg	0.79	0.74	0.76		29092	weighted avg	0.84	0.85	0.84		29092	<div><pre>y_pred1 = xgbC.predict(X_test1) print(confusion_matrix(y_test1, y_pred1)) print() print(accuracy_score(y_test1, y_pred1)) print() print(classification_report(y_test1, y_pred1))</pre><div>[[21308 1409] [2925 3450]] 0.851024336587378</div><table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.88</td><td>0.94</td><td>0.91</td><td>22717</td></tr><tr><td>1</td><td>0.71</td><td>0.54</td><td>0.61</td><td>6375</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.85</td><td>29092</td></tr><tr><td>macro avg</td><td>0.79</td><td>0.74</td><td>0.76</td><td>29092</td></tr><tr><td>weighted avg</td><td>0.84</td><td>0.85</td><td>0.84</td><td>29092</td></tr></table></div> <div><pre>metrics.plot_roc_curve(xgbC, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre><div>0.7395762178622384</div></div>		precision	recall	f1-score	support	0	0.88	0.94	0.91	22717	1	0.71	0.54	0.61	6375	accuracy			0.85	29092	macro avg	0.79	0.74	0.76	29092	weighted avg	0.84	0.85	0.84	29092
	recall	f1-score	support	precision																																																																	
0	0.88	0.94	0.91	22717																																																																	
1	0.71	0.54	0.61	6375																																																																	
accuracy				0.85	29092																																																																
macro avg	0.79	0.74	0.76		29092																																																																
weighted avg	0.84	0.85	0.84		29092																																																																
	precision	recall	f1-score	support																																																																	
0	0.88	0.94	0.91	22717																																																																	
1	0.71	0.54	0.61	6375																																																																	
accuracy			0.85	29092																																																																	
macro avg	0.79	0.74	0.76	29092																																																																	
weighted avg	0.84	0.85	0.84	29092																																																																	

5.	Metrics	<div>Cat Boost Classifier</div> <div>Confusion Matrix – [[21416 1301] [2891 3484]]</div> <div>Accuracy Score- 0.855905403547367</div> <div>& Classification Report – <table><tr><th></th><th>recall</th><th>f1-score</th><th>support</th><th>precision</th><th></th></tr><tr><td>0</td><td></td><td>0.88</td><td>0.94</td><td>0.91</td><td>22717</td></tr><tr><td>1</td><td></td><td>0.73</td><td>0.55</td><td>0.62</td><td>6375</td></tr><tr><td></td><td>accuracy</td><td></td><td></td><td>0.86</td><td>29092</td></tr><tr><td></td><td>macro avg</td><td>0.80</td><td>0.74</td><td>0.77</td><td>29092</td></tr><tr><td></td><td>weighted avg</td><td>0.85</td><td>0.86</td><td>0.85</td><td>29092</td></tr></table></div>		recall	f1-score	support	precision		0		0.88	0.94	0.91	22717	1		0.73	0.55	0.62	6375		accuracy			0.86	29092		macro avg	0.80	0.74	0.77	29092		weighted avg	0.85	0.86	0.85	29092	<div><pre>y_pred1 = catBoostC.predict(X_test1) print(confusion_matrix(y_test1, y_pred1)) print() print(accuracy_score(y_test1, y_pred1)) print() print(classification_report(y_test1, y_pred1))</pre></div> <div>[[21416 1301] [2891 3484]]</div> <div>0.855905403547367</div> <div><table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>0.88</td><td>0.94</td><td>0.91</td><td>22717</td></tr><tr><td>1</td><td>0.73</td><td>0.55</td><td>0.62</td><td>6375</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.86</td><td>29092</td></tr><tr><td>macro avg</td><td>0.80</td><td>0.74</td><td>0.77</td><td>29092</td></tr><tr><td>weighted avg</td><td>0.85</td><td>0.86</td><td>0.85</td><td>29092</td></tr></table></div> <div><pre>metrics.plot_roc_curve(catBoostC, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre></div> <div>0.744619958966551</div> <div></div>		precision	recall	f1-score	support	0	0.88	0.94	0.91	22717	1	0.73	0.55	0.62	6375	accuracy			0.86	29092	macro avg	0.80	0.74	0.77	29092	weighted avg	0.85	0.86	0.85	29092
		recall	f1-score	support	precision																																																																
0		0.88	0.94	0.91	22717																																																																
1		0.73	0.55	0.62	6375																																																																
	accuracy			0.86	29092																																																																
	macro avg	0.80	0.74	0.77	29092																																																																
	weighted avg	0.85	0.86	0.85	29092																																																																
	precision	recall	f1-score	support																																																																	
0	0.88	0.94	0.91	22717																																																																	
1	0.73	0.55	0.62	6375																																																																	
accuracy			0.86	29092																																																																	
macro avg	0.80	0.74	0.77	29092																																																																	
weighted avg	0.85	0.86	0.85	29092																																																																	
6.	Hyperparameters Tuning	<div>iterations = {1000, 2000, 3000,}</div> <div>eval_metric = 'AUC'</div>	<div><pre>catBoostC = CatBoostClassifier(iterations = 3000, eval_metric = 'AUC')</pre></div>																																																																		