

## Project Development Phase Model Performance Test

Date	27 June 2025
Team ID	LTVIP2025TMID38326
Project Name	Revolutionizing Liver Care: Predicting Liver Cirrhosis Using Advanced Machine Learning Techniques
Maximum Marks	

### Model Performance Testing:

Our project Model Performance Testing:

S.No.	Parameter	Values	Screenshot																																																						
1.	Trained and compared Logistic Regression, Support Vector Classifier, XGBoost, and K-Nearest Neighbors. KNN showed best performance.	<ul style="list-style-type: none"><li>• Data Size: 950 records, 42 features</li><li>• Target Variable: Binary classification – Patient has liver cirrhosis (Yes/No)</li><li>• Train-Test Split: 80-20%</li><li>• Best Model: K-Nearest Neighbors.</li><li>• Evaluation Metrics Used: Accuracy, Precision, Recall, F1-score, Confusion Matrix</li><li>• Fine-tuning: Performed using RandomizedSearchCV for k parameter in KNN</li></ul>	<pre># Define your model list with trained models model_list = {     'logistic regression': log,     'logistic regression CV': lcv,     'naive bayes': nb,     'XGBoost': model,     'Ridge classifier': rg,     'Random Forest': rf,     'Support Vector Classifier': svc,     'KNN': knn # Assuming knn is your trained KNN model }  print(model_eval_info_df)</pre> <table><thead><tr><th></th><th>Name</th><th>Accuracy</th><th>F1 Score</th><th>Precision</th><th>Recall</th></tr></thead><tbody><tr><td>0</td><td>logistic regression</td><td>78.42</td><td>84.53</td><td>91.80</td><td>78.32</td></tr><tr><td>1</td><td>logistic regression CV</td><td>82.63</td><td>87.36</td><td>93.44</td><td>82.01</td></tr><tr><td>2</td><td>naive bayes</td><td>35.79</td><td>0.00</td><td>0.00</td><td>0.00</td></tr><tr><td>3</td><td>XGBoost</td><td>35.79</td><td>6.15</td><td>3.28</td><td>50.00</td></tr><tr><td>4</td><td>Ridge classifier</td><td>84.21</td><td>88.37</td><td>93.44</td><td>83.82</td></tr><tr><td>5</td><td>Random Forest</td><td>35.79</td><td>0.00</td><td>0.00</td><td>0.00</td></tr><tr><td>6</td><td>Support Vector Classifier</td><td>35.79</td><td>0.00</td><td>0.00</td><td>0.00</td></tr><tr><td>7</td><td>KNN</td><td>86.32</td><td>89.84</td><td>94.26</td><td>85.82</td></tr></tbody></table>		Name	Accuracy	F1 Score	Precision	Recall	0	logistic regression	78.42	84.53	91.80	78.32	1	logistic regression CV	82.63	87.36	93.44	82.01	2	naive bayes	35.79	0.00	0.00	0.00	3	XGBoost	35.79	6.15	3.28	50.00	4	Ridge classifier	84.21	88.37	93.44	83.82	5	Random Forest	35.79	0.00	0.00	0.00	6	Support Vector Classifier	35.79	0.00	0.00	0.00	7	KNN	86.32	89.84	94.26	85.82
	Name	Accuracy	F1 Score	Precision	Recall																																																				
0	logistic regression	78.42	84.53	91.80	78.32																																																				
1	logistic regression CV	82.63	87.36	93.44	82.01																																																				
2	naive bayes	35.79	0.00	0.00	0.00																																																				
3	XGBoost	35.79	6.15	3.28	50.00																																																				
4	Ridge classifier	84.21	88.37	93.44	83.82																																																				
5	Random Forest	35.79	0.00	0.00	0.00																																																				
6	Support Vector Classifier	35.79	0.00	0.00	0.00																																																				
7	KNN	86.32	89.84	94.26	85.82																																																				
2.	Accuracy	Training Accuracy - 91.30% (KNN)	<pre>Train score with tuned model: 0.9130434782608695 Test score with tuned model: 0.8736842105263158 Optimal hyperparameters for KNN: {'n_neighbors': np.int64(8)} Accuracy on test set: 0.87 Confusion Matrix (KNN): [[ 55  17]  [ 7 115]] Classification Report (KNN):</pre> <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>0.88</td><td>0.75</td><td>0.81</td><td>48</td></tr><tr><td>1</td><td>0.87</td><td>0.94</td><td>0.91</td><td>122</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.87</td><td>190</td></tr><tr><td>macro avg</td><td>0.88</td><td>0.85</td><td>0.86</td><td>190</td></tr><tr><td>weighted avg</td><td>0.87</td><td>0.87</td><td>0.87</td><td>190</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.88	0.75	0.81	48	1	0.87	0.94	0.91	122	accuracy			0.87	190	macro avg	0.88	0.85	0.86	190	weighted avg	0.87	0.87	0.87	190																								
	precision	recall	f1-score	support																																																					
0	0.88	0.75	0.81	48																																																					
1	0.87	0.94	0.91	122																																																					
accuracy			0.87	190																																																					
macro avg	0.88	0.85	0.86	190																																																					
weighted avg	0.87	0.87	0.87	190																																																					
3.	Validation Accuracy - 87.36% (after tuning KNN using RandomizedSearchCV)	Validation Accuracy -87.36% (KNN)	<pre>Train score with tuned model: 0.9130434782608695 Test score with tuned model: 0.8736842105263158 Optimal hyperparameters for KNN: {'n_neighbors': np.int64(8)} Accuracy on test set: 0.87 Confusion Matrix (KNN): [[ 55  17]  [ 7 115]] Classification Report (KNN):</pre> <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>0.88</td><td>0.75</td><td>0.81</td><td>48</td></tr><tr><td>1</td><td>0.87</td><td>0.94</td><td>0.91</td><td>122</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.87</td><td>190</td></tr><tr><td>macro avg</td><td>0.88</td><td>0.85</td><td>0.86</td><td>190</td></tr><tr><td>weighted avg</td><td>0.87</td><td>0.87</td><td>0.87</td><td>190</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.88	0.75	0.81	48	1	0.87	0.94	0.91	122	accuracy			0.87	190	macro avg	0.88	0.85	0.86	190	weighted avg	0.87	0.87	0.87	190																								
	precision	recall	f1-score	support																																																					
0	0.88	0.75	0.81	48																																																					
1	0.87	0.94	0.91	122																																																					
accuracy			0.87	190																																																					
macro avg	0.88	0.85	0.86	190																																																					
weighted avg	0.87	0.87	0.87	190																																																					