

```
range(50, 400, 50)
```

```
n_splits=5
```

```
Best: -1.100342 using {'n_estimators': 50}
```

```
-1.100342 (0.002228) with: {'n_estimators': 50}
```

```
-1.112120 (0.002707) with: {'n_estimators': 100}
```

```
-1.123898 (0.003198) with: {'n_estimators': 150}
```

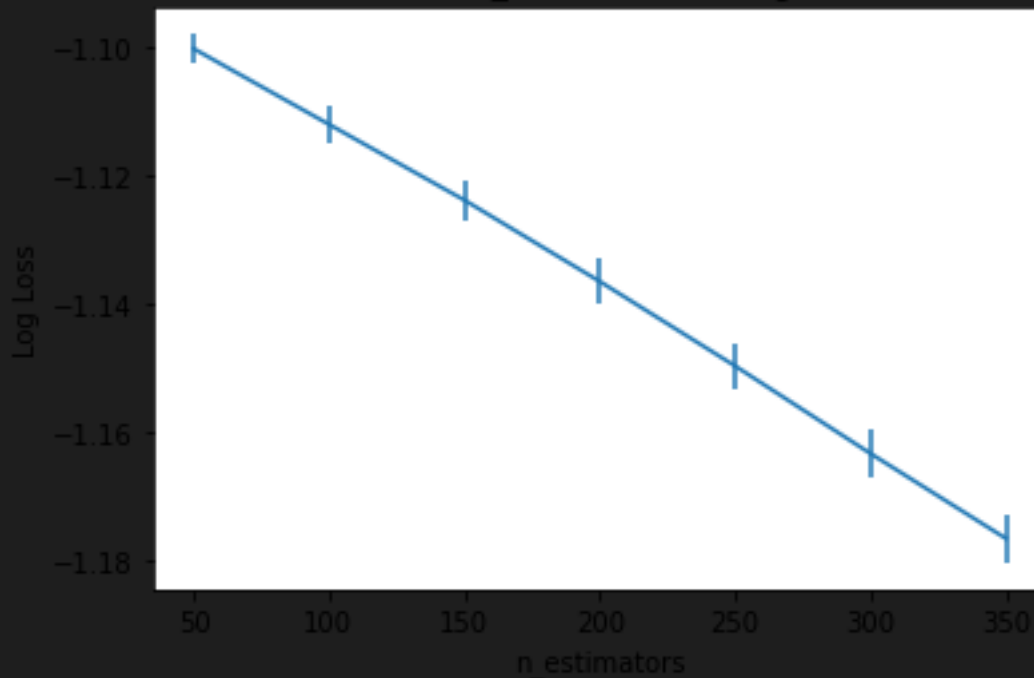
```
-1.136555 (0.003456) with: {'n_estimators': 200}
```

```
-1.149706 (0.003570) with: {'n_estimators': 250}
```

```
-1.163323 (0.003785) with: {'n_estimators': 300}
```

```
-1.176534 (0.003839) with: {'n_estimators': 350}
```

XGBoost n_estimators vs Log Loss



```
range(10, 50, 10)
```

```
n_splits=5
```

```
Best: -1.096355 using {'n_estimators': 30}
```

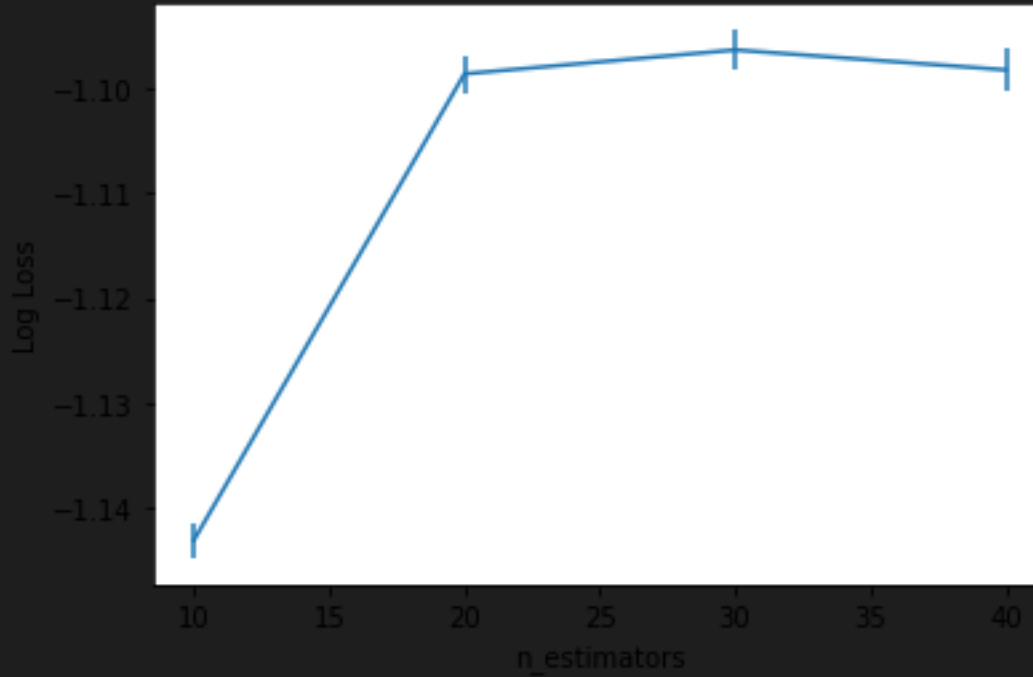
```
-1.143042 (0.001708) with: {'n_estimators': 10}
```

```
-1.098631 (0.001797) with: {'n_estimators': 20}
```

```
-1.096355 (0.001867) with: {'n_estimators': 30}
```

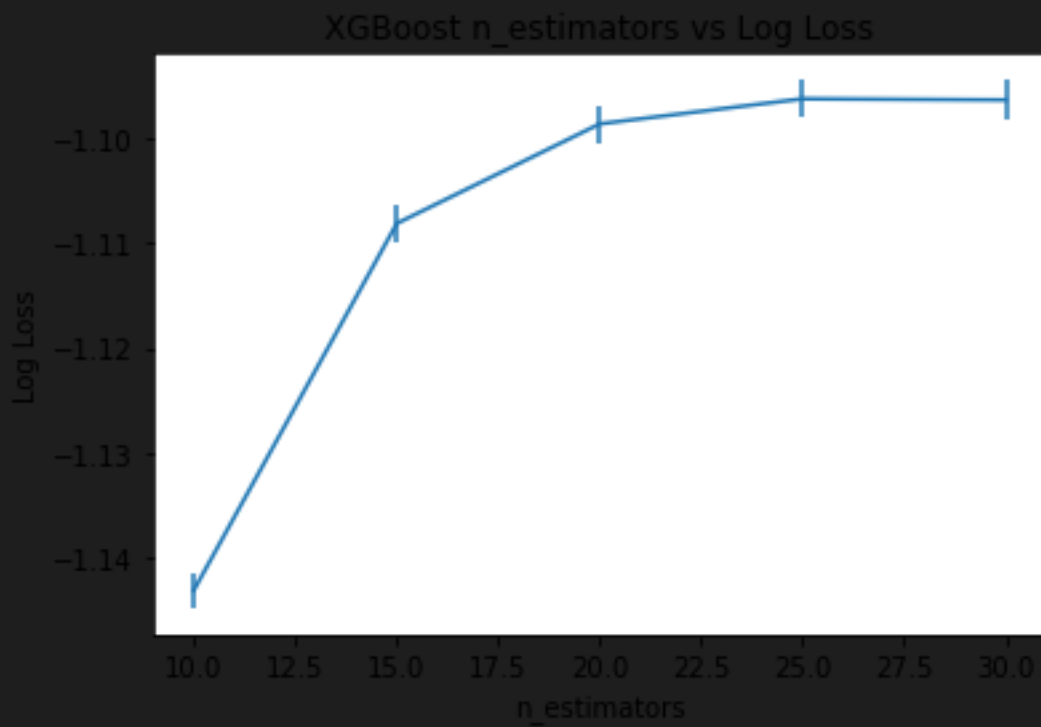
```
-1.098251 (0.002034) with: {'n_estimators': 40}
```

XGBoost n_estimators vs Log Loss



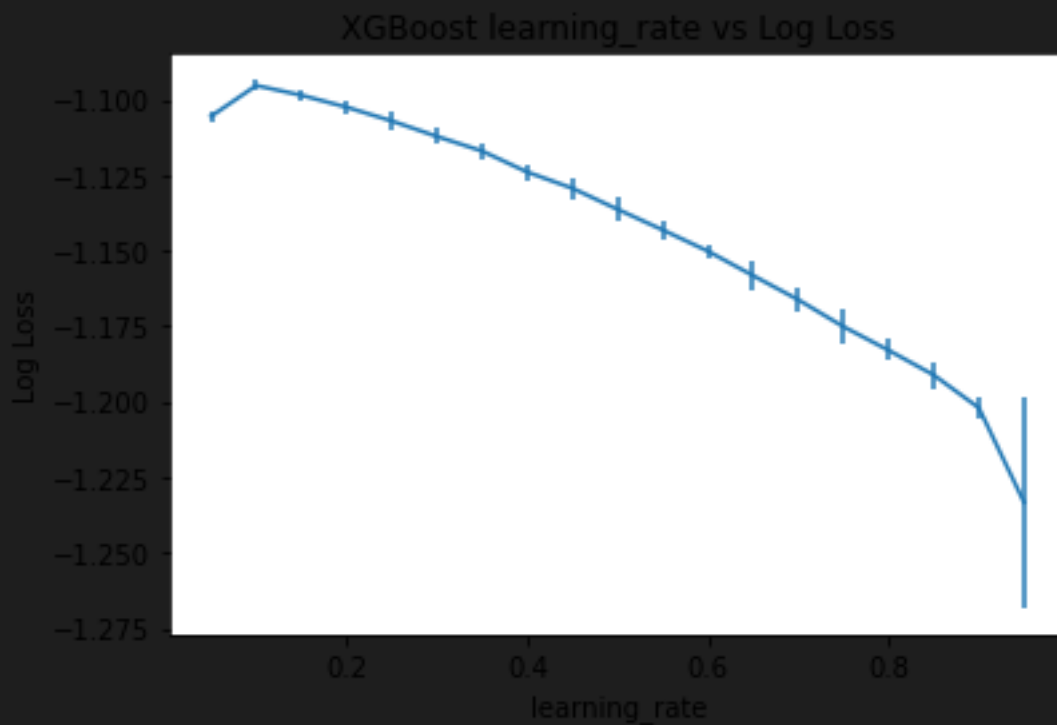
```
range(10, 35, 5)  
n_splits=5
```

```
Best: -1.096243 using {'n_estimators': 25}  
-1.143042 (0.001708) with: {'n_estimators': 10}  
-1.108116 (0.001704) with: {'n_estimators': 15}  
-1.098631 (0.001797) with: {'n_estimators': 20}  
-1.096243 (0.001734) with: {'n_estimators': 25}  
-1.096355 (0.001867) with: {'n_estimators': 30}
```



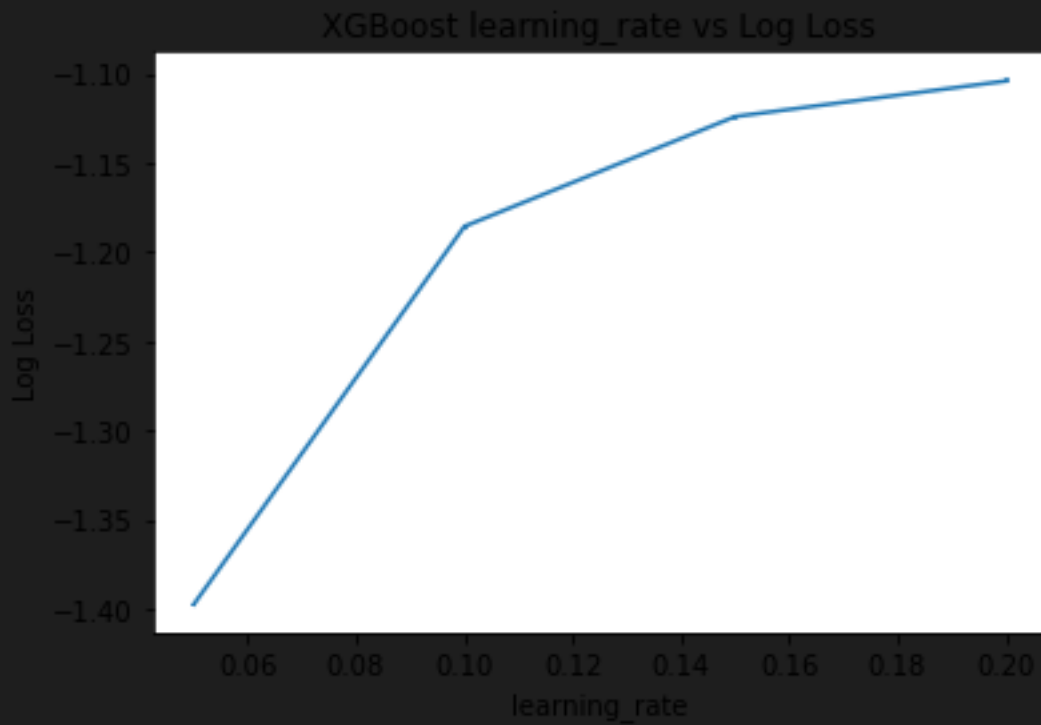
```
np.arange(0.05, 1.0, 0.05)  
n_splits=5
```

```
Best: -1.095263 using {'learning_rate': 0.1}  
-1.105450 (0.001559) with: {'learning_rate': 0.05}  
-1.095263 (0.001713) with: {'learning_rate': 0.1}  
-1.098482 (0.002098) with: {'learning_rate': 0.15}  
-1.102429 (0.002032) with: {'learning_rate': 0.2}  
-1.106996 (0.002807) with: {'learning_rate': 0.25}  
-1.112120 (0.002707) with: {'learning_rate': 0.3}  
-1.116954 (0.002883) with: {'learning_rate': 0.35}  
-1.123909 (0.002748) with: {'learning_rate': 0.4}  
-1.129275 (0.003600) with: {'learning_rate': 0.45}  
-1.136240 (0.003832) with: {'learning_rate': 0.5}  
-1.143072 (0.003091) with: {'learning_rate': 0.55}  
-1.150093 (0.002223) with: {'learning_rate': 0.6}  
-1.158179 (0.004971) with: {'learning_rate': 0.65}  
-1.166181 (0.004109) with: {'learning_rate': 0.7}  
-1.175081 (0.005406) with: {'learning_rate': 0.75}  
-1.182909 (0.003591) with: {'learning_rate': 0.8}  
-1.191088 (0.004351) with: {'learning_rate': 0.85}  
-1.201975 (0.003522) with: {'learning_rate': 0.9}  
-1.233195 (0.035114) with: {'learning_rate': 0.95}
```



```
n_estimators=25  
np.arange(0.05, 0.2, 0.05)  
n_splits=5
```

```
Best: -1.104023 using {'learning_rate': 0.2}  
-1.397490 (0.001072) with: {'learning_rate': 0.05}  
-1.185552 (0.001383) with: {'learning_rate': 0.1}  
-1.123941 (0.001420) with: {'learning_rate': 0.15}  
-1.104023 (0.001504) with: {'learning_rate': 0.2}
```



```
n_estimators=25
```

```
learning_rate = np.arange(0.2, 0.5, 0.05)
```

```
n_splits=5
```

```
Best: -1.096243 using {'learning_rate': 0.3}
```

```
-1.104023 (0.001504) with: {'learning_rate': 0.2}
```

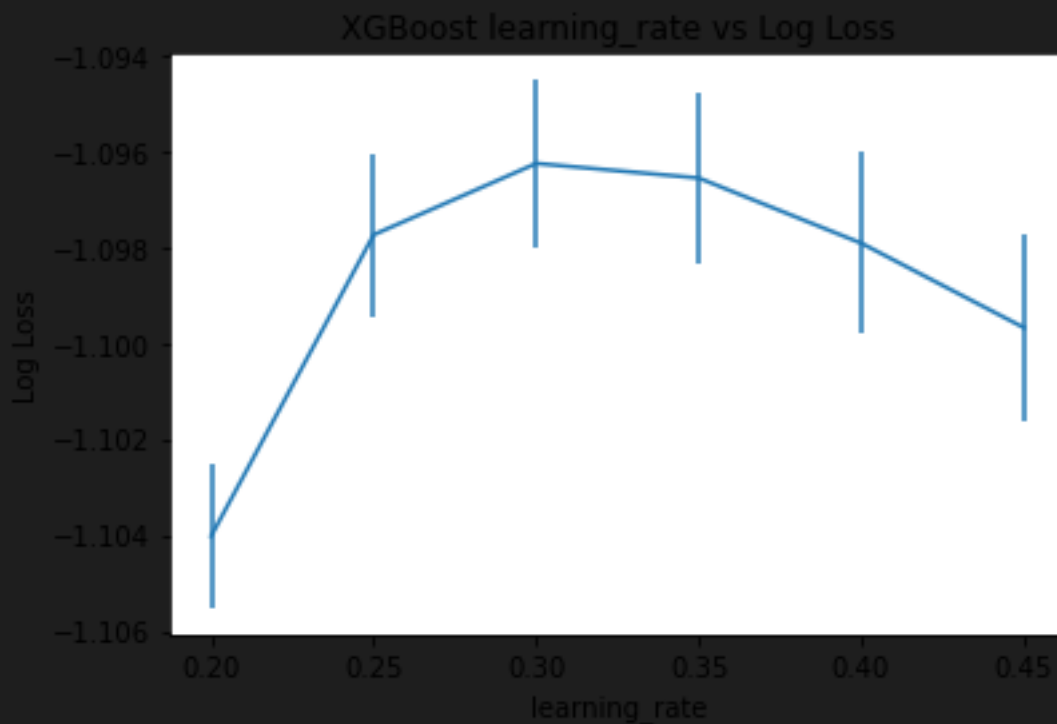
```
-1.097732 (0.001693) with: {'learning_rate': 0.25}
```

```
-1.096243 (0.001734) with: {'learning_rate': 0.3}
```

```
-1.096548 (0.001803) with: {'learning_rate': 0.35}
```

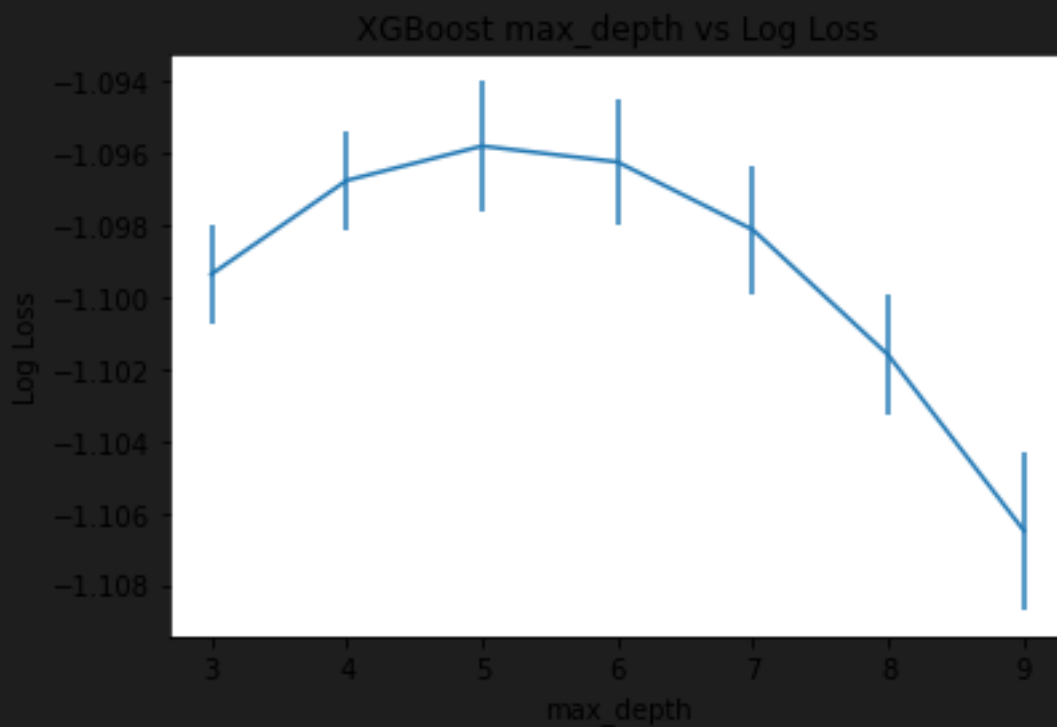
```
-1.097909 (0.001890) with: {'learning_rate': 0.399}
```

```
-1.099671 (0.001967) with: {'learning_rate': 0.449}
```



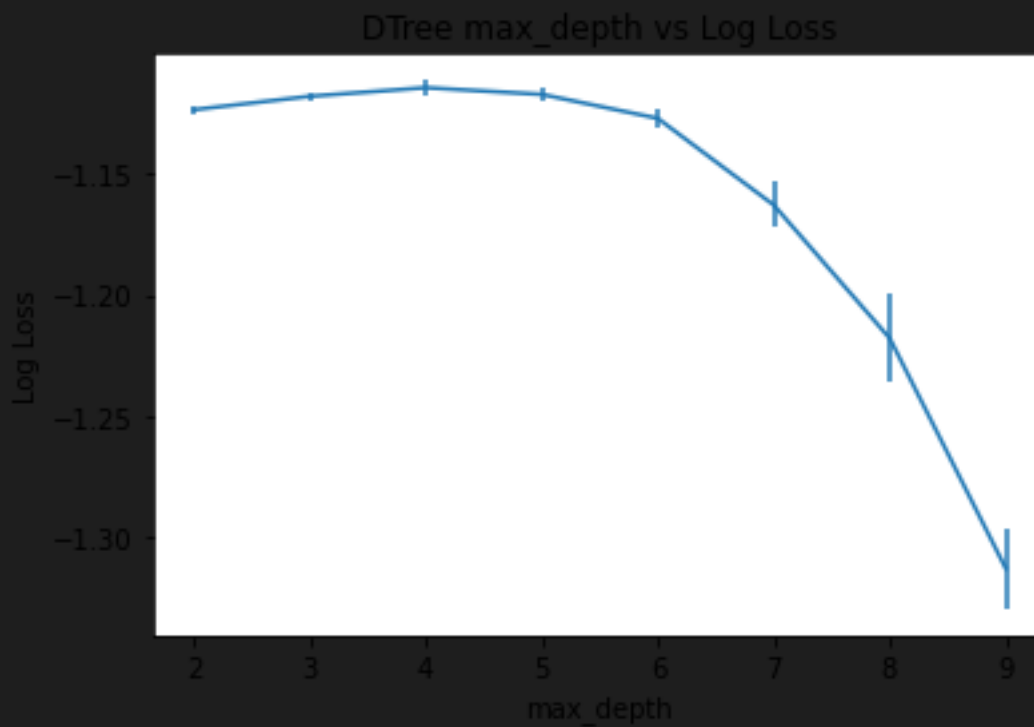
```
n_estimators=25, learning_rate=0.3  
range(3, 10, 1)  
n_splits=5
```

```
Best: -1.095799 using {'max_depth': 5}  
-1.099366 (0.001394) with: {'max_depth': 3}  
-1.096756 (0.001397) with: {'max_depth': 4}  
-1.095799 (0.001794) with: {'max_depth': 5}  
-1.096243 (0.001734) with: {'max_depth': 6}  
-1.098127 (0.001800) with: {'max_depth': 7}  
-1.101614 (0.001675) with: {'max_depth': 8}  
-1.106487 (0.002202) with: {'max_depth': 9}
```



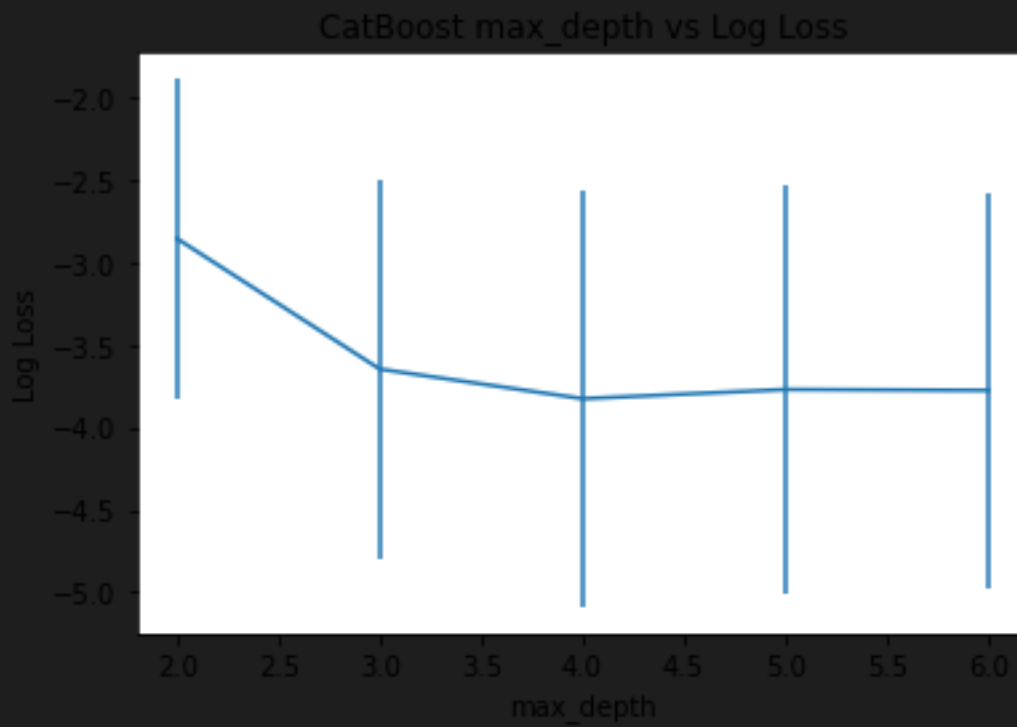
DTree

```
Best: -1.114450 using {'max_depth': 4}
-1.123525 (0.001435) with: {'max_depth': 2}
-1.118003 (0.001822) with: {'max_depth': 3}
-1.114450 (0.002972) with: {'max_depth': 4}
-1.117220 (0.002846) with: {'max_depth': 5}
-1.127086 (0.003823) with: {'max_depth': 6}
-1.162801 (0.009436) with: {'max_depth': 7}
-1.217732 (0.018014) with: {'max_depth': 8}
-1.312511 (0.016755) with: {'max_depth': 9}
```

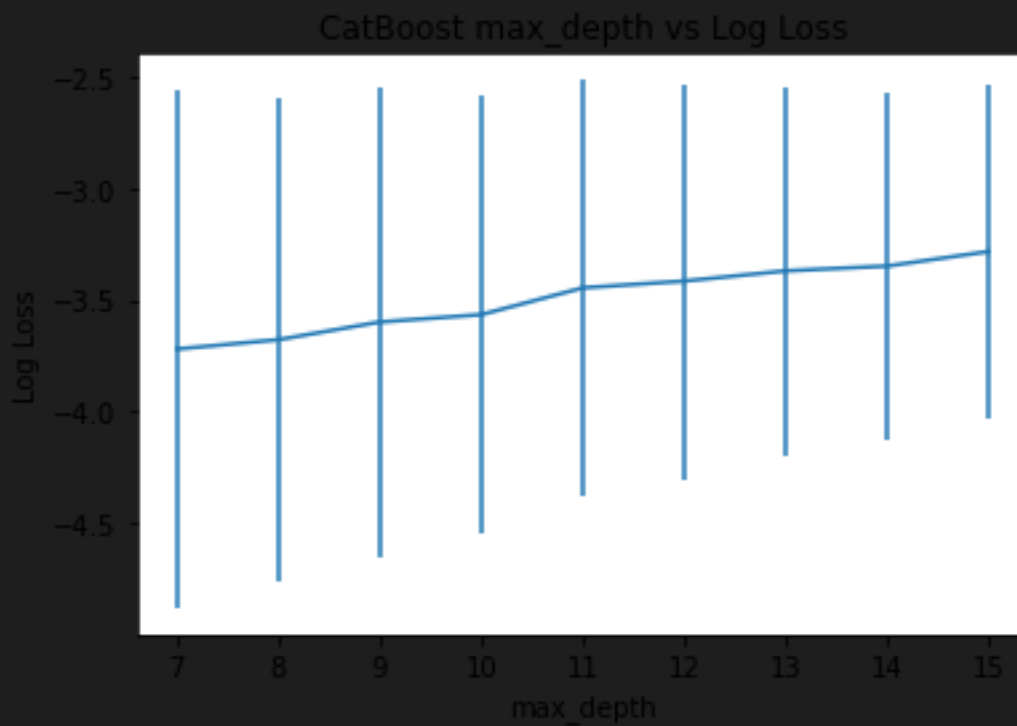


CatBoost

```
Best: -2.853930 using {'max_depth': 2}  
-2.853930 (0.965470) with: {'max_depth': 2}  
-3.646661 (1.152948) with: {'max_depth': 3}  
-3.826226 (1.264978) with: {'max_depth': 4}  
-3.770364 (1.246268) with: {'max_depth': 5}  
-3.775947 (1.199639) with: {'max_depth': 6}
```



```
Best: -3.282312 using {'max_depth': 15}
-3.718812 (1.159693) with: {'max_depth': 7}
-3.677008 (1.082534) with: {'max_depth': 8}
-3.597895 (1.050442) with: {'max_depth': 9}
-3.564781 (0.980964) with: {'max_depth': 10}
-3.444886 (0.928562) with: {'max_depth': 11}
-3.414600 (0.884335) with: {'max_depth': 12}
-3.368979 (0.826077) with: {'max_depth': 13}
-3.347443 (0.773817) with: {'max_depth': 14}
-3.282312 (0.749220) with: {'max_depth': 15}
```

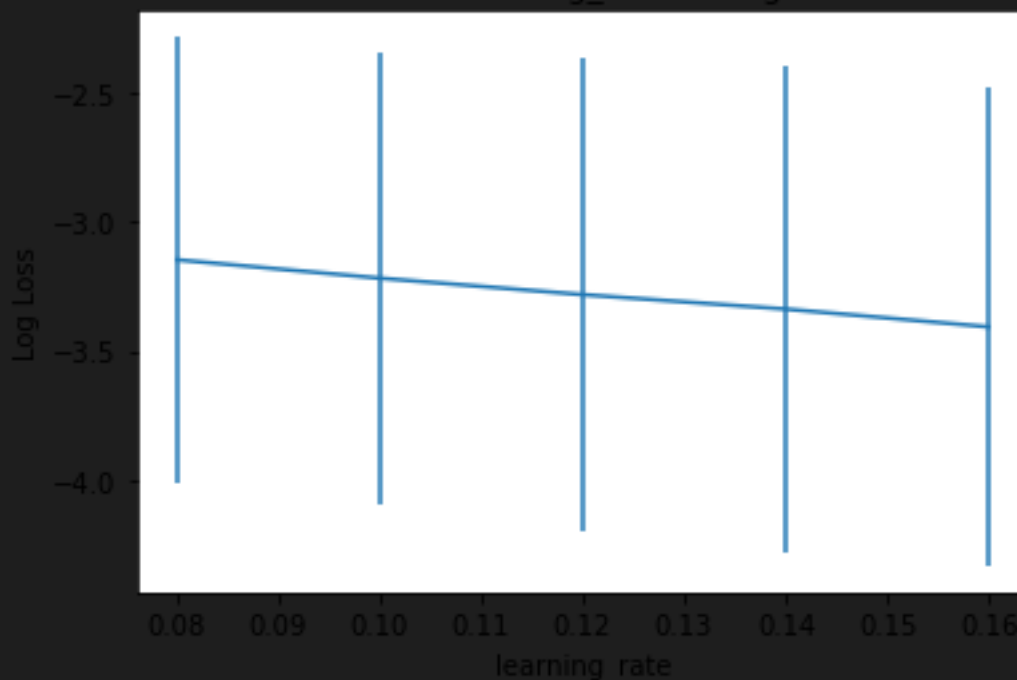


We'll use depth 11 to avoid overfitting.

```
Best: -3.211709 using {'learning_rate': 0.1}
-3.211709 (0.869604) with: {'learning_rate': 0.1}
-3.371588 (0.908517) with: {'learning_rate': 0.15}
-3.486291 (0.937487) with: {'learning_rate': 0.2}
-3.647119 (0.955429) with: {'learning_rate': 0.25}
-3.795891 (0.948137) with: {'learning_rate': 0.3}
-3.895610 (0.936535) with: {'learning_rate': 0.35}
-3.994661 (0.916300) with: {'learning_rate': 0.4}
-4.150152 (1.016399) with: {'learning_rate': 0.45}
-4.223616 (1.006911) with: {'learning_rate': 0.5}
-4.390883 (1.062381) with: {'learning_rate': 0.55}
```

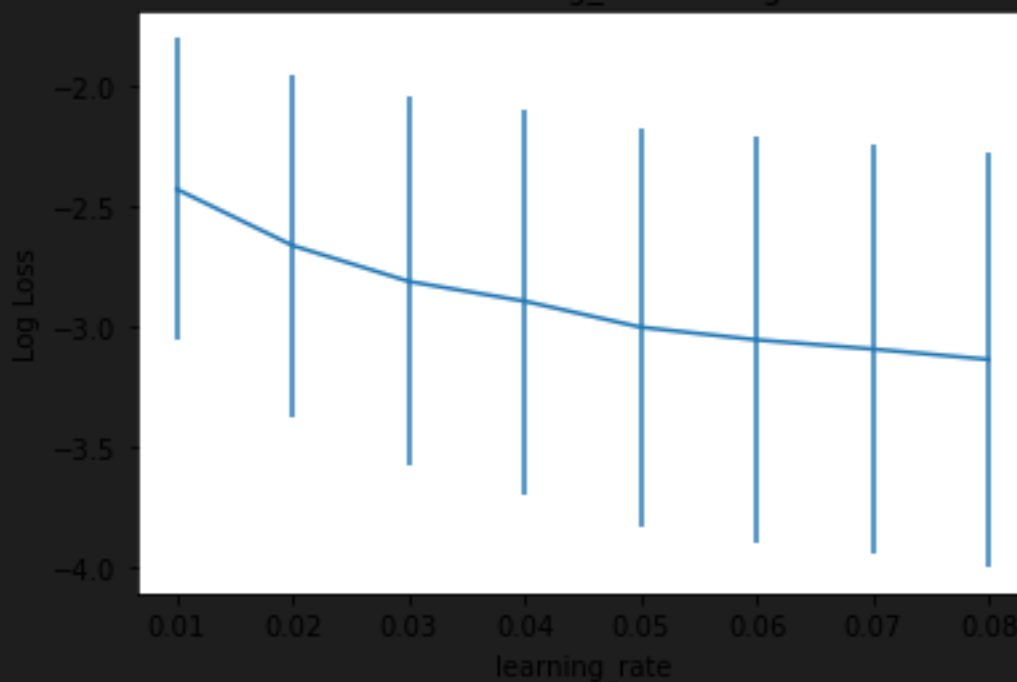
```
Best: -3.145579 using {'learning_rate': 0.08}
-3.145579 (0.862124) with: {'learning_rate': 0.08}
-3.216895 (0.879153) with: {'learning_rate': 0.1}
-3.280788 (0.915726) with: {'learning_rate': 0.12}
-3.336793 (0.939939) with: {'learning_rate': 0.14}
-3.405818 (0.927321) with: {'learning_rate': 0.16}
```

CatBoost learning_rate vs Log Loss

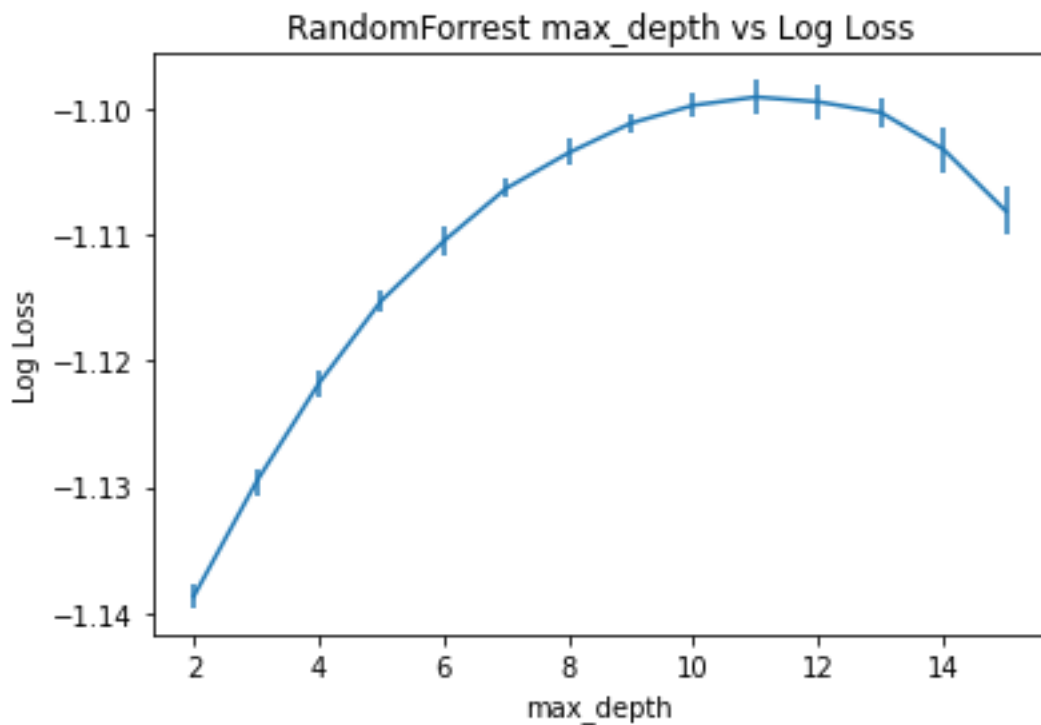


```
Best: -2.427671 using {'learning_rate': 0.01}
-2.427671 (0.627099) with: {'learning_rate': 0.01}
-2.662094 (0.709147) with: {'learning_rate': 0.02}
-2.811026 (0.770587) with: {'learning_rate': 0.03}
-2.893334 (0.801483) with: {'learning_rate': 0.04}
-3.001489 (0.825639) with: {'learning_rate': 0.05}
-3.054062 (0.844487) with: {'learning_rate': 0.06}
-3.092333 (0.845483) with: {'learning_rate': 0.069}
-3.135859 (0.862941) with: {'learning_rate': 0.08}
```

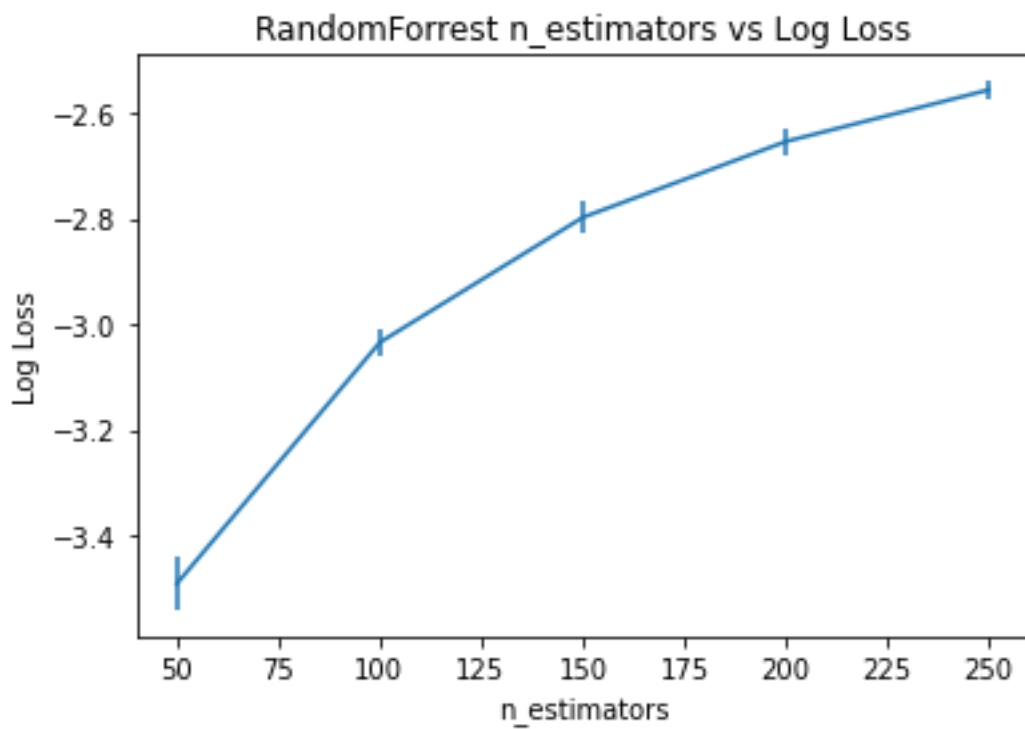
CatBoost learning_rate vs Log Loss



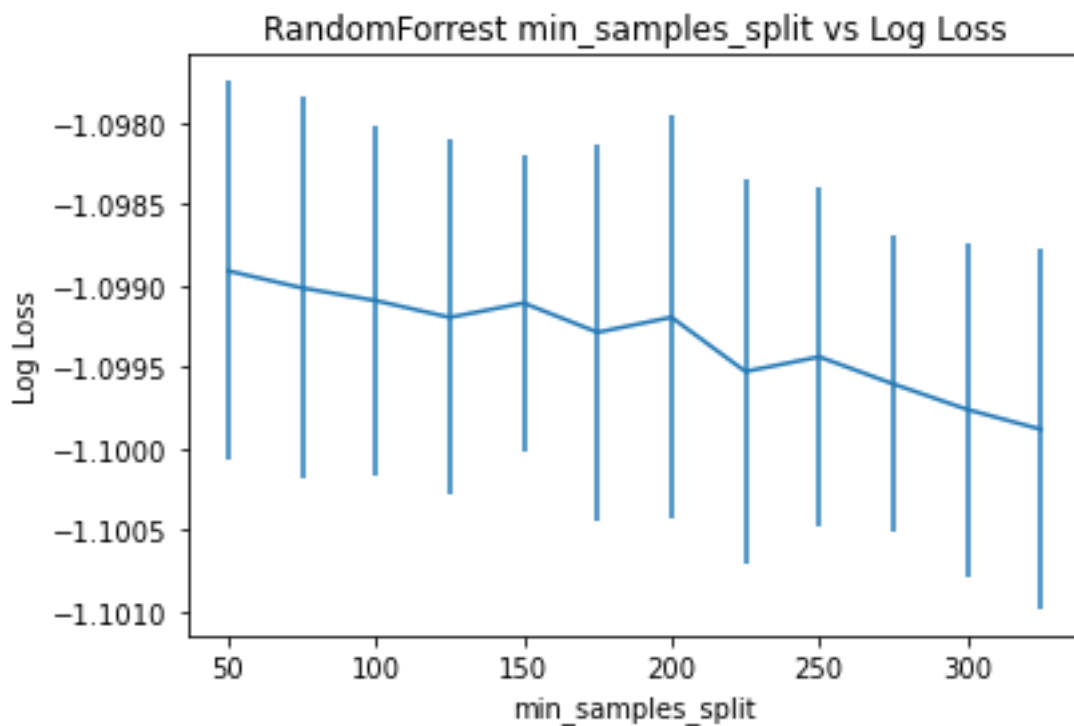
Best: -1.099071 using {'max_depth': 11}
-1.138641 (0.000951) with: {'max_depth': 2}
-1.129592 (0.001081) with: {'max_depth': 3}
-1.121788 (0.001102) with: {'max_depth': 4}
-1.115233 (0.000854) with: {'max_depth': 5}
-1.110498 (0.001245) with: {'max_depth': 6}
-1.106325 (0.000764) with: {'max_depth': 7}
-1.103499 (0.001055) with: {'max_depth': 8}
-1.101144 (0.000709) with: {'max_depth': 9}
-1.099729 (0.000978) with: {'max_depth': 10}
-1.099071 (0.001318) with: {'max_depth': 11}
-1.099455 (0.001456) with: {'max_depth': 12}
-1.100293 (0.001109) with: {'max_depth': 13}
-1.103206 (0.001805) with: {'max_depth': 14}
-1.108103 (0.001904) with: {'max_depth': 15}



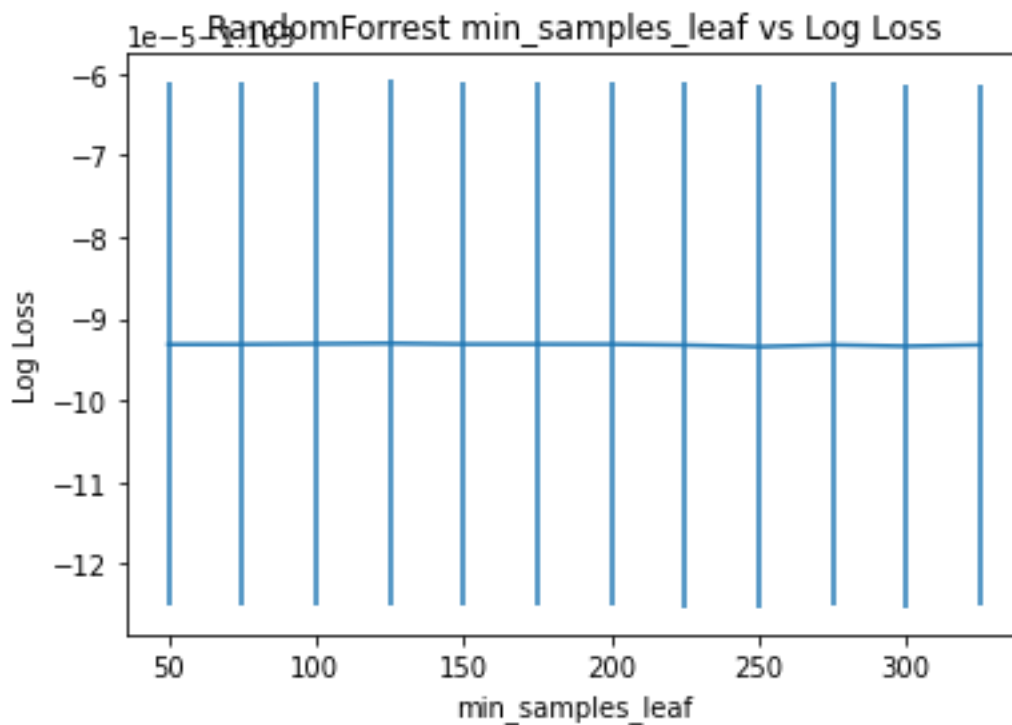
Best: -2.556128 using {'n_estimators': 250}
-3.491031 (0.049859) with: {'n_estimators': 50}
-3.034237 (0.024136) with: {'n_estimators': 100}
-2.797610 (0.031510) with: {'n_estimators': 150}
-2.653941 (0.024113) with: {'n_estimators': 200}
-2.556128 (0.015931) with: {'n_estimators': 250}



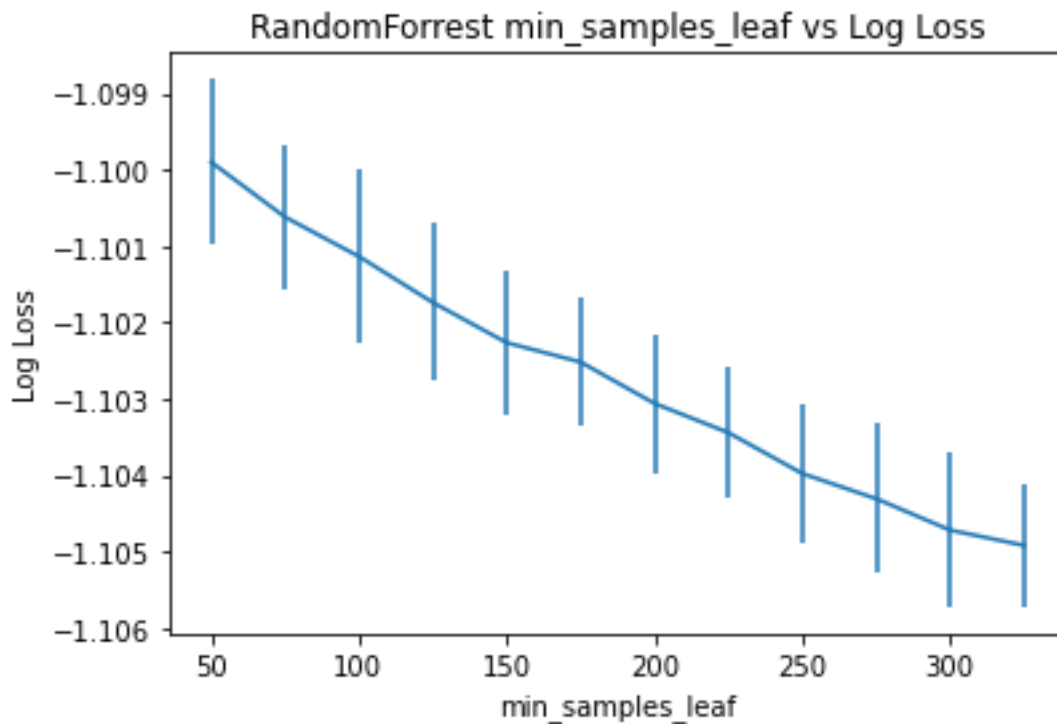
Best: -1.098908 using {'min_samples_split': 50}
-1.098908 (0.001163) with: {'min_samples_split': 50}
-1.099013 (0.001171) with: {'min_samples_split': 75}
-1.099091 (0.001072) with: {'min_samples_split': 100}
-1.099194 (0.001091) with: {'min_samples_split': 125}
-1.099106 (0.000904) with: {'min_samples_split': 150}
-1.099285 (0.001157) with: {'min_samples_split': 175}
-1.099193 (0.001238) with: {'min_samples_split': 200}
-1.099526 (0.001185) with: {'min_samples_split': 225}
-1.099436 (0.001038) with: {'min_samples_split': 250}
-1.099602 (0.000910) with: {'min_samples_split': 275}
-1.099759 (0.001021) with: {'min_samples_split': 300}
-1.099881 (0.001104) with: {'min_samples_split': 325}



Best: -1.163093 using {'min_samples_leaf': 125}
-1.163093 (0.000032) with: {'min_samples_leaf': 50}
-1.163093 (0.000032) with: {'min_samples_leaf': 75}
-1.163093 (0.000032) with: {'min_samples_leaf': 100}
-1.163093 (0.000032) with: {'min_samples_leaf': 125}
-1.163093 (0.000032) with: {'min_samples_leaf': 150}
-1.163093 (0.000032) with: {'min_samples_leaf': 175}
-1.163093 (0.000032) with: {'min_samples_leaf': 200}
-1.163093 (0.000032) with: {'min_samples_leaf': 225}
-1.163093 (0.000032) with: {'min_samples_leaf': 250}
-1.163093 (0.000032) with: {'min_samples_leaf': 275}
-1.163093 (0.000032) with: {'min_samples_leaf': 300}
-1.163093 (0.000032) with: {'min_samples_leaf': 325}

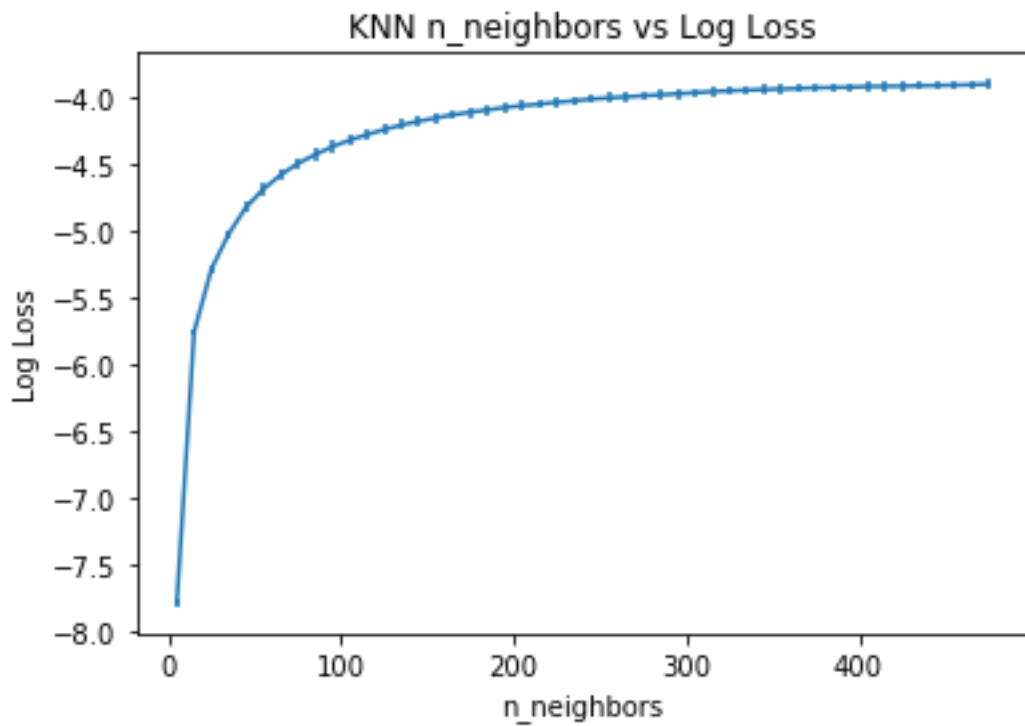


Best: -1.099900 using {'min_samples_leaf': 50}
-1.099900 (0.001083) with: {'min_samples_leaf': 50}
-1.100619 (0.000935) with: {'min_samples_leaf': 75}
-1.101135 (0.001127) with: {'min_samples_leaf': 100}
-1.101738 (0.001030) with: {'min_samples_leaf': 125}
-1.102265 (0.000940) with: {'min_samples_leaf': 150}
-1.102518 (0.000840) with: {'min_samples_leaf': 175}
-1.103060 (0.000911) with: {'min_samples_leaf': 200}
-1.103442 (0.000866) with: {'min_samples_leaf': 225}
-1.103978 (0.000902) with: {'min_samples_leaf': 250}
-1.104307 (0.000980) with: {'min_samples_leaf': 275}
-1.104718 (0.001006) with: {'min_samples_leaf': 300}
-1.104919 (0.000809) with: {'min_samples_leaf': 325}



Best: -3.897427 using {'n_neighbors': 475}

- 7.785743 (0.031444) with: {'n_neighbors': 5}
- 5.754482 (0.022161) with: {'n_neighbors': 15}
- 5.283463 (0.027134) with: {'n_neighbors': 25}
- 5.020585 (0.031626) with: {'n_neighbors': 35}
- 4.821358 (0.039362) with: {'n_neighbors': 45}
- 4.685367 (0.041506) with: {'n_neighbors': 55}
- 4.579255 (0.043993) with: {'n_neighbors': 65}
- 4.492156 (0.044155) with: {'n_neighbors': 75}
- 4.426814 (0.046933) with: {'n_neighbors': 85}
- 4.365463 (0.049260) with: {'n_neighbors': 95}
- 4.317553 (0.046713) with: {'n_neighbors': 105}
- 4.276223 (0.040196) with: {'n_neighbors': 115}
- 4.238013 (0.037277) with: {'n_neighbors': 125}
- 4.204536 (0.040463) with: {'n_neighbors': 135}
- 4.175703 (0.035770) with: {'n_neighbors': 145}
- 4.151520 (0.035362) with: {'n_neighbors': 155}
- 4.126799 (0.036823) with: {'n_neighbors': 165}
- 4.107953 (0.037129) with: {'n_neighbors': 175}
- 4.090584 (0.035743) with: {'n_neighbors': 185}
- 4.072532 (0.037231) with: {'n_neighbors': 195}
- 4.058774 (0.035167) with: {'n_neighbors': 205}
- 4.047271 (0.036166) with: {'n_neighbors': 215}
- 4.033368 (0.034470) with: {'n_neighbors': 225}
- 4.022534 (0.036146) with: {'n_neighbors': 235}
- 4.009408 (0.034484) with: {'n_neighbors': 245}
- 4.000399 (0.036677) with: {'n_neighbors': 255}
- 3.993245 (0.035919) with: {'n_neighbors': 265}
- 3.984560 (0.034705) with: {'n_neighbors': 275}
- 3.978107 (0.037725) with: {'n_neighbors': 285}
- 3.969203 (0.037749) with: {'n_neighbors': 295}
- 3.962016 (0.036975) with: {'n_neighbors': 305}
- 3.953374 (0.034352) with: {'n_neighbors': 315}
- 3.947084 (0.033191) with: {'n_neighbors': 325}
- 3.942710 (0.034452) with: {'n_neighbors': 335}
- 3.937407 (0.034266) with: {'n_neighbors': 345}
- 3.934452 (0.034175) with: {'n_neighbors': 355}
- 3.929452 (0.034523) with: {'n_neighbors': 365}
- 3.924906 (0.035704) with: {'n_neighbors': 375}
- 3.923038 (0.034934) with: {'n_neighbors': 385}
- 3.920542 (0.035657) with: {'n_neighbors': 395}
- 3.915538 (0.034561) with: {'n_neighbors': 405}
- 3.913476 (0.033560) with: {'n_neighbors': 415}
- 3.912383 (0.034904) with: {'n_neighbors': 425}
- 3.909104 (0.035448) with: {'n_neighbors': 435}
- 3.907076 (0.034036) with: {'n_neighbors': 445}
- 3.904847 (0.033818) with: {'n_neighbors': 455}
- 3.902207 (0.035302) with: {'n_neighbors': 465}
- 3.897427 (0.035822) with: {'n_neighbors': 475}



Best k is 475 and \sqrt{n} of n samples in the training set is 462. Which backs the common rule of thumb of choosing \sqrt{n} for k neighbours.