Decision tree:
Splitter :
Best and Random
Train time higher for best. Test times are almost same.

Accuracy was almost equal. Sometime random might also get slight increase in accuracy compared to best splitter. As some random sequences might be better than best and it is greedy.
The negative log loss micro and macro auc are almost same for both.
Outcome: Cannot predict the accuracy. But the time taken is more for the splitter with best in most cases.

Depth:
More depth more accuracy. But after certain depth the increase in accuracy is very less.
Hence depth of 25 is good for mnist data. As more depths more nodes on which split is made so more accuracy.

Presort:
Accuracy is similar. No much difference. But the time taken for sorting is huge. Almost 6 times more . Shouldn't make a difference.

Max features:
As number of features increases the accuracy is increasing. The time taken increases for training as the number of features increase. The testing time is almost the same. 100 as more features more nodes to test criteria on

Criteria:
Both gini and entropy have no difference. Only gini is faster as no log.

Random Forest:

Number of estimators:
As the number of estimators increase the accuracy increases. The time taken to train increases very highly. The time taken for training is almost same as the increase in the multiplicative factor increase in the estimators. The test time increases for a huge jump in the number of estimators. For 100 and 250 the difference in test time was 2.5 times seconds. For 500 and 1000 it almost doubles. 250 value is good.

Random state:
We cannot predict a pattern using the random state. Increaing or decreasing the random state had no effect as the accuracy was almost the same. The time taken also was not predictable.

Criterion:
For different number of estimators the accuracy was same for both entropy and gini. The time taken for entropy was high due to the log calculation involved in the formula.
https://datascience.stackexchange.com/questions/10228/gini-impurity-vs-entropy

max depth:
As the depth increases the accuracy was increasing for various number of estimators. The accuracy keeps increasing upto a depth and then increases very slowly after that. Time taken in while testing is almost same except under the case with very less depth. After a certain depth the time and accuracy increase is very less. 25 depth is fine. The depth also depends on the number of estimators. With more estimators the depth required is very less.

Max Features:
As the features increase the time taken to train also increaes with increase in accuracy. The time taken also increaes almost double for double increase in the number of features.
Log2 or sqrt number of features are fine.

Bootstrap:
Had no effect on accuracy . But the train time for the model without bootstraping is higher than the one with boostraping.

Reducing the number of samples:
When trained using the half of the training examples the accuracy was almost using the entire training data with the time being half.

Random search:
{'max_depth': 30, 'bootstrap': False, 'max_features': 30, 'n_estimators': 35}

Both random search and grid search gave almost the same outputs.

Random Forest with PPCA
At 50 dimension with 100 estimators it gave 94% accuracy. Which is because ??