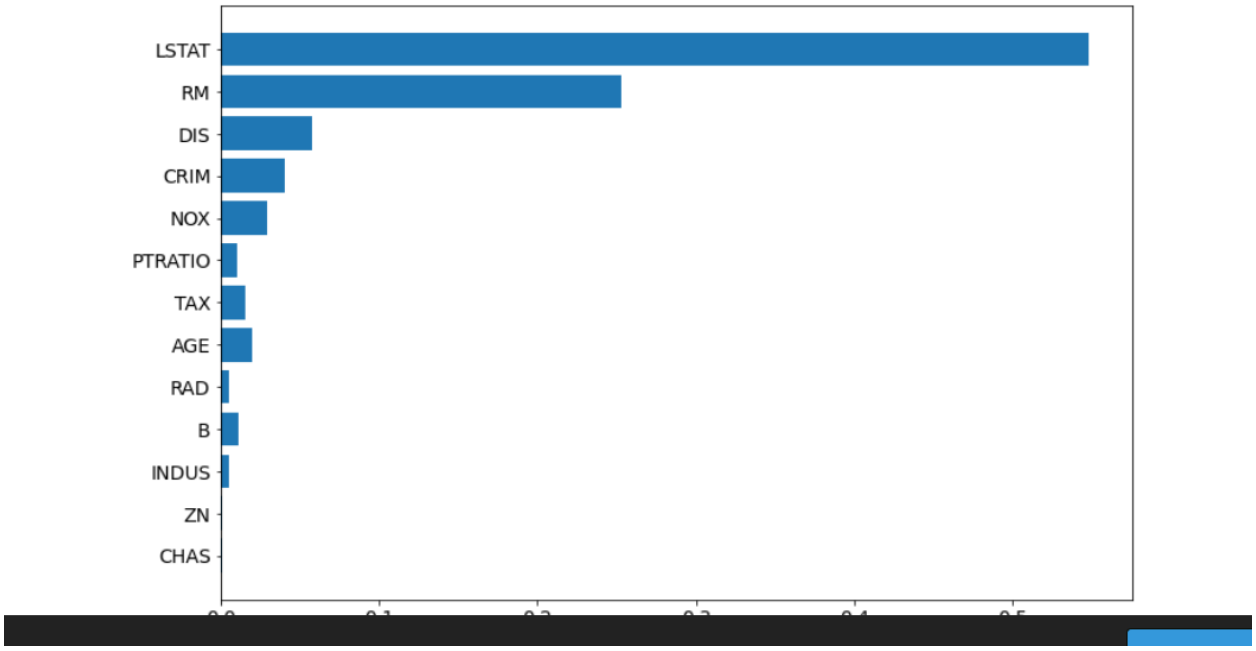
FEATURE IMPORTANCE

to measure the relative importance of each feature by checking how much does using that feature reduce impurity

Scikit-Learn computes this score automatically for each feature after training, then it scales the results so that the sum of all importances is equal to 1. You can access the result using the feature_importances_ variable

```
>>> from sklearn.datasets import load_iris
>>> iris = load_iris()
>>> rnd_clf = RandomForestClassifier(n_estimators=500, n_jobs=-1)
>>> rnd_clf.fit(iris["data"], iris["target"])
>>> for name, score in zip(iris["feature_names"], rnd_clf.feature_importances_):
...     print(name, score)
...
sepal length (cm) 0.112492250999
sepal width (cm) 0.0231192882825
petal length (cm) 0.441030464364
petal width (cm) 0.423357996355
```

```
sorted_idx = rf.feature_importances_.argsort()
plt.barh(boston.feature_names[sorted_idx], rf.feature_importances_[sorted_idx])
plt.xlabel("Random Forest Feature Importance")
```



Permutation Based Feature Importance (with scikit-learn)

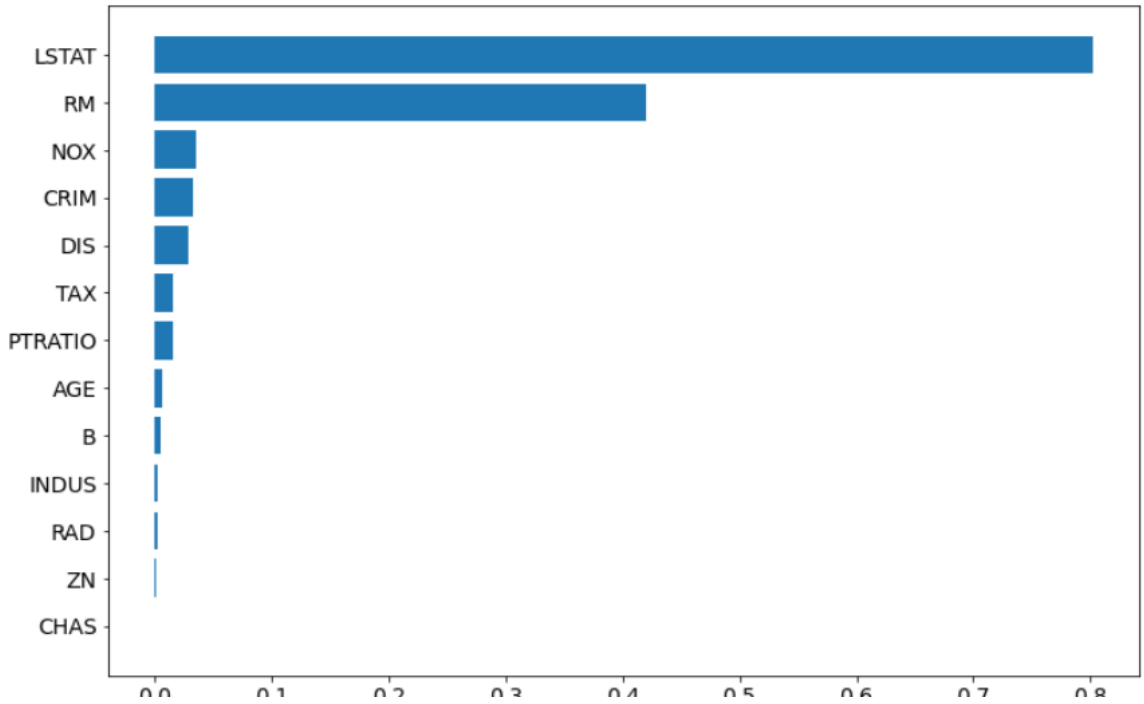
The permutation based importance can be used to overcome drawbacks of default feature importance computed with mean impurity decrease. It is implemented in scikit-learn as permutation_importance method.

This method will randomly shuffle each feature and compute the change in the model's performance.

```
perm_importance = permutation_importance(clf, X_test, y_test)

sorted_idx = perm_importance.importances_mean.argsort()
plt.barh(boston.feature_names[sorted_idx],
perm_importance.importances_mean[sorted_idx])

plt.xlabel("Permutation Importance")
```



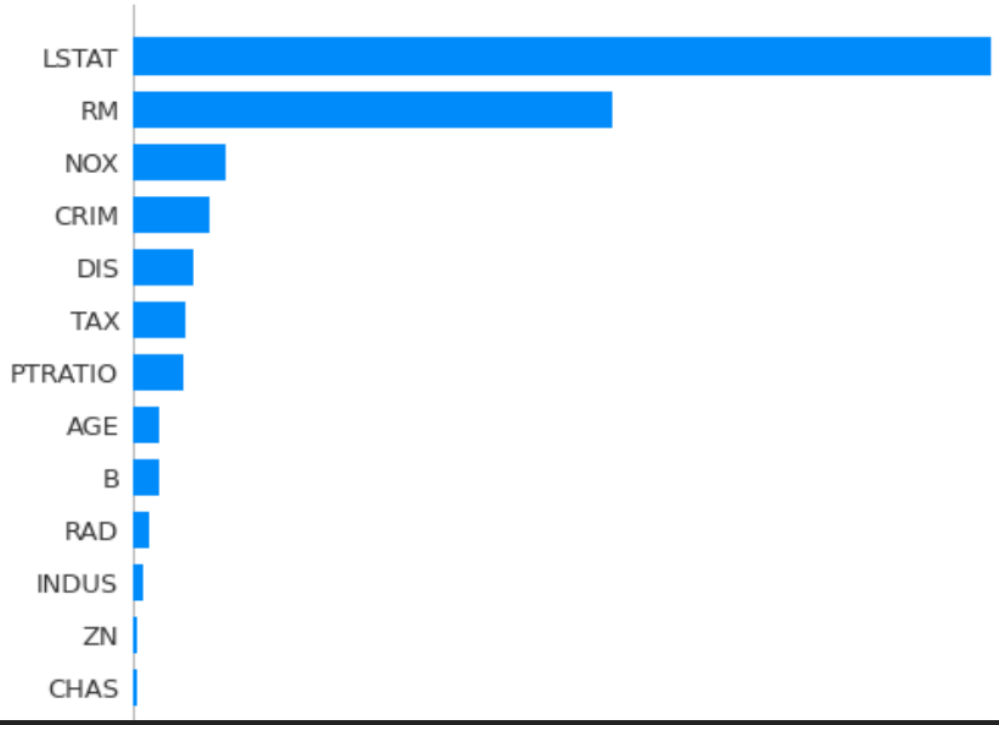
Feature Importance Computed with SHAP Values

It is using the Shapley values from game theory to estimate the how does each feature contribute to the prediction. It can be easily installed (`pip install shap`)

```
explainer = shap.TreeExplainer(rf)
shap_values = explainer.shap_values(X_test)
```

To plot feature importance as the horizontal bar plot we need to use `summary_plot` method:

```
shap.summary_plot(shap_values, X_test, plot_type="bar")
```



Information gain can also be used for feature selection, by evaluating the gain of each variable in the context of the target variable. In this slightly different usage, the calculation is referred to as mutual information between the two random variables.