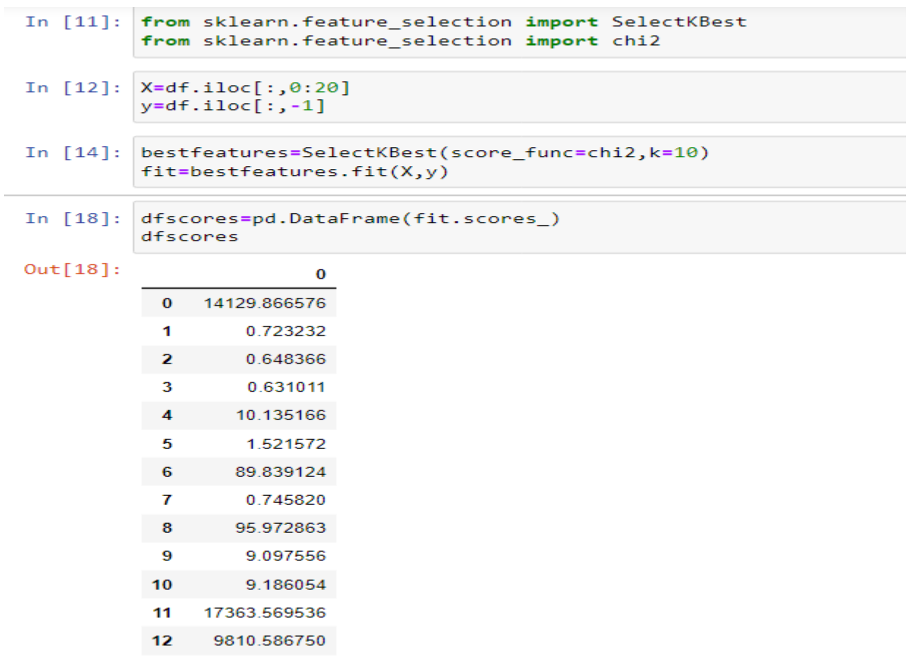
# Write a short guidance note explaining feature selection techniques in machine learning to a hypothetical student struggling with the concept.

Feature selection is the important step in the machine learning. On the basic of selected features your model performance is decided. Selected feature must have high relationship with output or target variable. There are methods to select the best features,

1. Filter Method – Here we are using univariate statistics using following calculations.

A. Chi-square test: The Chi-square test is used for categorical features in a dataset. We calculate Chi-square between each feature and the target and select the desired number of features with the best Chi-square scores. To correctly apply the chi-squared to test the relation between various features in the dataset and the target variable, the following conditions must be met, the variables must be *categorical*, sampled *independently*, and values should have an *expected frequency greater than 5*.

Code Snippet:



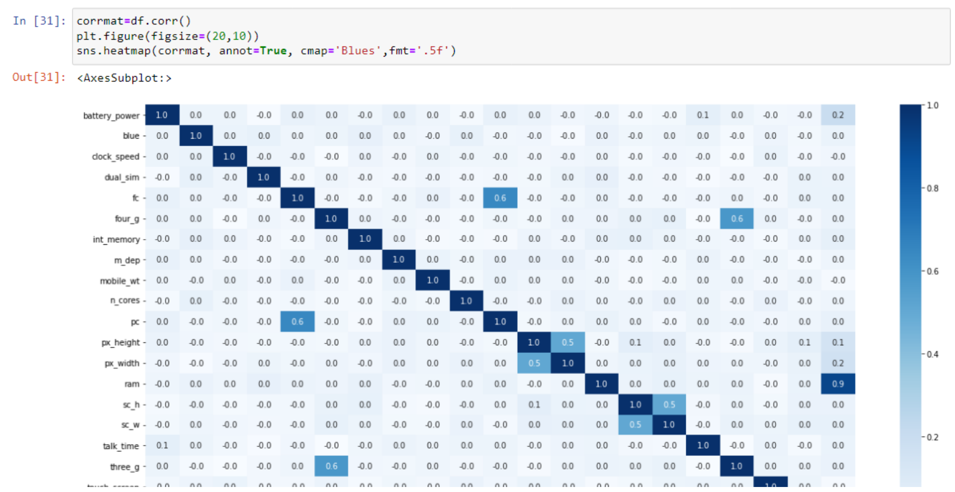
```
In [11]: from sklearn.feature_selection import SelectKBest
         from sklearn.feature_selection import chi2

In [12]: X=df.iloc[:,0:20]
         y=df.iloc[:,-1]

In [14]: bestfeatures=SelectKBest(score_func=chi2,k=10)
         fit=bestfeatures.fit(X,y)

In [18]: dfscores=pd.DataFrame(fit.scores_)
         dfscores

Out[18]:
```

|  | 0 |
|---|---|
| 0 | 14129.866576 |
| 1 | 0.723232 |
| 2 | 0.648366 |
| 3 | 0.631011 |
| 4 | 10.135166 |
| 5 | 1.521572 |
| 6 | 89.839124 |
| 7 | 0.745820 |
| 8 | 95.972863 |
| 9 | 9.097556 |
| 10 | 9.186054 |
| 11 | 17363.569536 |
| 12 | 9810.586750 |

## 2. Correlation Matrix:

Correlation is a measure of the linear relationship between 2 or more variables. Through correlation, we can predict one variable from the other. The logic behind using correlation for feature selection is that good variables correlate highly with the target. Furthermore, variables should be correlated with the target but uncorrelated among themselves.

Code Snippet:



```
In [31]: cormat=df.corr()
         plt.figure(figsize=(20,10))
         sns.heatmap(cormat, annot=True, cmap='Blues',fmt='.5f')

Out[31]: <AxesSubplot:>
```

## 3. Random Forest Importance: Random Forests is a kind of Bagging Algorithm that aggregates a specified number of decision trees. The tree-based strategies used by random forests naturally rank by how well they improve the purity of the node, or in other words, a decrease in the impurity (**Gini impurity**) over all trees. Nodes with the greatest decrease in impurity happen at the start of the trees, while notes with the least decrease in impurity occur at the end of the trees.

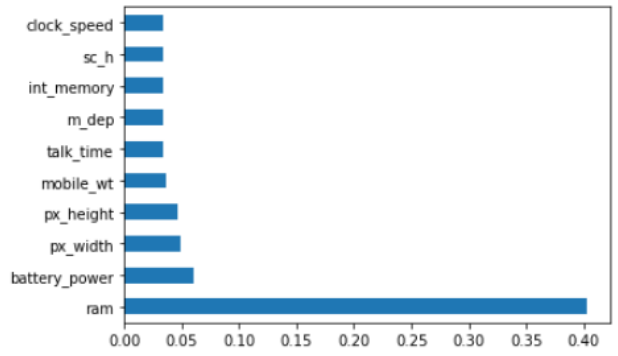
## Code Snippet:

```
In [23]: from sklearn.ensemble import ExtraTreesClassifier
```

```
In [24]: model=ExtraTreesClassifier()
         model.fit(X,y)
         print(model.feature_importances_)
```

```
[0.06107769 0.0192468  0.0333744  0.01899694 0.03211109 0.01698986
 0.03366286 0.03368082 0.03597489 0.03243646 0.03202564 0.04665498
 0.04857605 0.40256834 0.03349916 0.03281232 0.03446693 0.01411307
 0.0185125  0.01921918]
```

```
In [26]: feat_importances=pd.Series(model.feature_importances_,index=X.columns)
         feat_importances.nlargest(10).plot(kind='barh')
         plt.show()
```



Apart from these methods there are many methods like wrapper, forward selection, backward elimination and variance threshold method. Using any one if fine for feature selection.