Dear Intern

Interim project report is an inherent component of your internship. We are enclosing a reference table of content for the interim project report.

The key objective of this report is for you to capture how far you have got in completing the internship work against milestones expected to be achieved within a specific duration and seek the mentor’s feedback. Depending on the internship project and your progress (IT/Non-IT, Technical/Business Domain), you may choose to include or exclude or rename sections or leave some sections blank from the table of content mentioned below. You can also add additional sections. You can refer the project presentation to view the milestones related to your internship project. Please populate milestone# (1 / 2 / 3) and the milestone description in the interim project report based on the milestone for which you are submitting the interim project report.

You can refer the project presentation to view the milestones related to your internship project.

|  |  |
| --- | --- |
| Internship Project Title | Rank features of a smartphone-build a python application to classify and rank dataset |
| Name of the Company | TCS iON |
| Name of the Industry Mentor | Debashis Roy |
| Name of the Institute | ICT Academy of Kerala |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Start Date | End Date | | Total Effort (hrs.) | | Project Environment | Tools used |
| 16/11/22 | 25/11/22 | | 21.5 | | Data Science and Analytics | python |
| Milestone # | 2 | Milestone: | | Merge the classified data and rank the final dataset. | | |

**TABLE OF CONTENT**

* Acknowledgements
* Objective
* Introduction / Description of Internship
* Internship Activities
* Approach / Methodology
* Assumptions
* Exceptions / Exclusions
* Charts, Table, Diagrams
* Algorithms
* Challenges & Opportunities
* Risk Vs Reward
* Reflections on the Internship
* Recommendations
* Outcome / Conclusion
* Enhancement Scope
* Link to code and executable file

**Acknowledgements**

First I would like to thank ICT Academy of Kerala, for giving me the opportunity to do an internship from TATA Consultancy Services.

I also would like to thank to the organization for giving an opportunity to do an internship of one month.

It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of the industry mentor Mr. Debashis Roy.

**Objective**

The objective of this project is to build a python application that ranks the features of a smartphone based on the requests received from various users.

**Introduction**

Smart phone is a mobile phone which offers advanced technologies with functionality similar as a personal computer. While offering a standardized platform for application developers a smart phone performs as complete operating system software. Secondly, there are also very advanced features in smart phones such as internet, instant messenger and e-mail and also built-in keyboard are very typical. Because of these reasons we can say a smart phone a miniature computer with the similarities of a simple phone.

Analyst house Gartner gives the definition of a smart phone as: “A large-screen, data-centric, handheld device designed to offer complete phone functions whilst simultaneously functioning as a personal digital assistant (PDA).” (Analyst House Gartner: 2009)

With the growing speed of technological advancement, Smart phones are now an essential part of our daily life routine. When we go for our convenience we also look for those devices which contain multiple features such as office work, mobility, networking and entertainment. As the world is getting advance our needs become sophisticated. Where we need quality, effectiveness and performance we also ask for these all in one single pocket device so we can take that to anywhere with us.

The main aim of the project is to build a python application to classify and rank dataset i.e. to rank features of a smartphone.  The dataset is obtained from the digital discussion room of TCSiON learning platform. This dataset contains information on battery power, clock speed, dual sim, four g, int memory, touch screen, Wi-Fi, price range, ram, talk time etc.

**Internship Activities**

**Milestone #2:** Merge the classified data and rank the final dataset.

**Day 6 to Day 15:** merged the classified data. Done feature engineering, identified the most important features, conduct feature scaling and modelling, split the data and applied various classification models on it (Logistic model, Decision tree, KNN, Random forest , Gradient Boost, SVM). Call the test data into python environment and predict the output using the highest accuracy model and saved it as a csv file.

**Methodology**

**Merging:** Data merging is the process of combining two or more similar records into a singleone. Merging is done to add variables to a dataset, append or add cases or observations to a dataset, or remove duplicates and other incorrect information.

**Feature Engineering:** Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling.

**Feature Selection:** Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data. It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve.

**Feature Scaling:**  Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing to handle highly varying magnitudes or values or units.

**Modelling:** Data modeling is the process of producing a descriptive diagram of relationships between various types of information that are to be stored in a database. One of the goals of data modeling is to create the most efficient method of storing information while still providing for complete access and reporting.

**Logistic Regression:** Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

 Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

 Works only when the predicted variable is binary, assumes all predictors are independent of each other and assumes data is free of missing values.

**K-Nearest Neighbours**:Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point.

This algorithm is simple to implement, robust to noisy training data, and effective if training data is large. Need to determine the value of K and the computation cost is high as it needs to compute the distance of each instance to all the training samples.

**Decision Tree:** Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

[Decision Tree](https://analyticsindiamag.com/hands-on-tutorial-how-to-use-decision-tree-regression-to-solve-machinehacks-new-data-science-hackathon/) is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data.Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

**Random Forest:** it is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases. Slow real time prediction, difficult to implement, and complex algorithm.

**Support Vector Machine:** it is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient. The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

**Gradient Boosting Classifier:** Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.

**Outcomes**

The classified hardware and software data is merged together. Next done the feature engineering, selection and scaling. Applied SelectKBest method to find out the best twelve features and the best features and it scores are:

Features Score

13 ram 931267.519053

11 px\_height 17363.569536

0 battery\_power 14129.866576

12 px\_width 9810.586750

8 mobile\_wt 95.972863

6 int\_memory 89.839124

15 sc\_w 16.480319

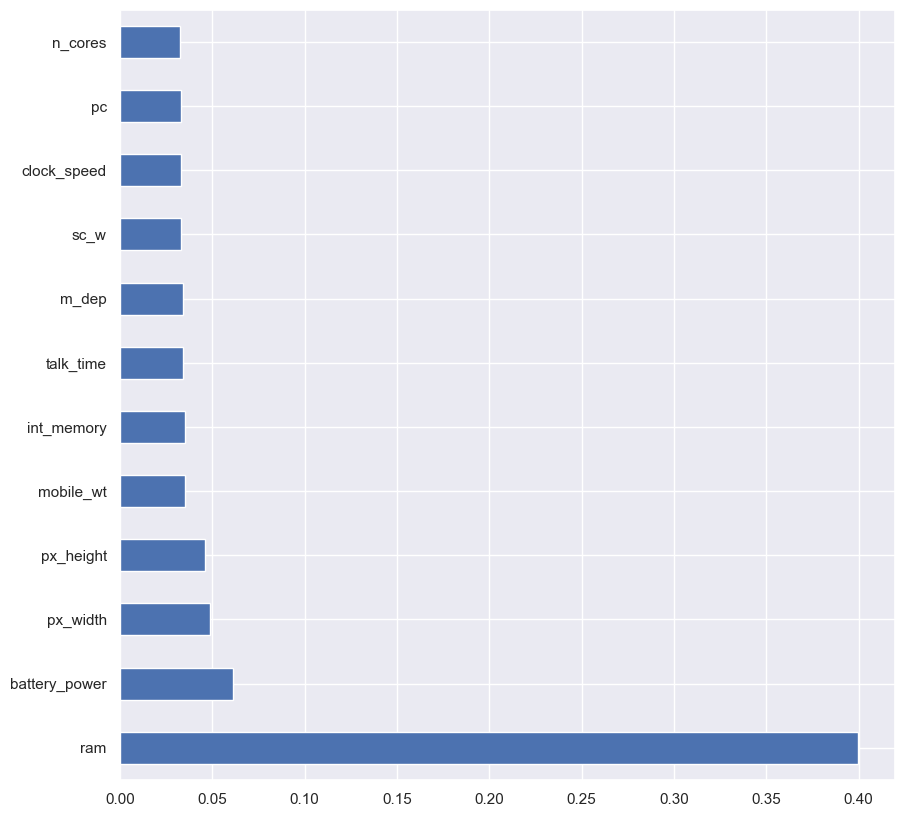
16 talk\_time 13.236400

4 fc 10.135166

14 sc\_h 9.614878

10 pc 9.186054

9 n\_cores 9.097556



Then by using StandardScalar standardise the train data.

**from** sklearn.preprocessing **import** StandardScaler

scaler**=** StandardScaler()

x**=**scaler**.**fit\_transform(x)

Then split the data into train and test set by giving 25% to test and remaining to train set. Then by using different classification models checked for the accuracy.

**Classification Models**

**Logistic Regression:** from sklearn library , imported the logistic model and checked for the accuracy. It gives 96.2% of accuracy.

Accuracy is: 0.962

Results from Logistic Regression

|  | **precision** | **recall** | **f1-score** | **support** |
| --- | --- | --- | --- | --- |
| **0** | 0.983871 | 0.976000 | 0.979920 | 125.000000 |
| **1** | 0.936364 | 0.936364 | 0.936364 | 110.000000 |
| **2** | 0.928000 | 0.958678 | 0.943089 | 121.000000 |
| **3** | 0.992908 | 0.972222 | 0.982456 | 144.000000 |
| **accuracy** | 0.962000 | 0.962000 | 0.962000 | 0.962000 |
| **macro avg** | 0.960286 | 0.960816 | 0.960457 | 500.000000 |
| **weighted avg** | 0.962501 | 0.962000 | 0.962155 | 500.000000 |
| **Decision Tree:** from sklearn library imported decision tree classifier and checked for the accuracy. The accuracy score is 84.8%.  Accuracy is: 0.848  Results from Decision Tree   |  | **precision** | **recall** | **f1-score** | **support** | | --- | --- | --- | --- | --- | | **0** | 0.895161 | 0.932773 | 0.913580 | 119.000000 | | **1** | 0.845455 | 0.781513 | 0.812227 | 119.000000 | | **2** | 0.776000 | 0.782258 | 0.779116 | 124.000000 | | **3** | 0.872340 | 0.891304 | 0.881720 | 138.000000 | | **accuracy** | 0.848000 | 0.848000 | 0.848000 | 0.848000 | | **macro avg** | 0.847239 | 0.846962 | 0.846661 | 500.000000 | | **weighted avg** | 0.847481 | 0.848000 | 0.847318 | 500.000000 | |  |  |  |  |

**KNN:** checked for the k value which gives the highest accuracy and the k value is 14 and the accuracy score is 59%.

accuracy score is : 0.59

Results from K-NN

|  | **precision** | **recall** | **f1-score** | **support** |
| --- | --- | --- | --- | --- |
| **0** | 0.725806 | 0.725806 | 0.725806 | 124.000000 |
| **1** | 0.536364 | 0.421429 | 0.472000 | 140.000000 |
| **2** | 0.488000 | 0.458647 | 0.472868 | 133.000000 |
| **3** | 0.602837 | 0.825243 | 0.696721 | 103.000000 |
| **accuracy** | 0.590000 | 0.590000 | 0.590000 | 0.590000 |
| **macro avg** | 0.588252 | 0.607781 | 0.591849 | 500.000000 |
| **weighted avg** | 0.584174 | 0.590000 | 0.581468 | 500.000000 |
| **Random Forest Classifier:** the accuracy score is 87.4%  Accuracy is: 0.874  Results from RandomForest Classifier   |  | **precision** | **recall** | **f1-score** | **support** | | --- | --- | --- | --- | --- | | **0** | 0.983871 | 0.945736 | 0.964427 | 129.000000 | | **1** | 0.827273 | 0.805310 | 0.816143 | 113.000000 | | **2** | 0.744000 | 0.808696 | 0.775000 | 115.000000 | | **3** | 0.929078 | 0.916084 | 0.922535 | 143.000000 | | **accuracy** | 0.874000 | 0.874000 | 0.874000 | 0.874000 | | **macro avg** | 0.871055 | 0.868956 | 0.869526 | 500.000000 | | **weighted avg** | 0.877639 | 0.874000 | 0.875366 | 500.000000 |   **Gradient Boosting Classifier:** the accuracy score is 92.2%.  Accuracy is: 0.922  Results from Gradient boosting classifier  Out[74]:   |  | **precision** | **recall** | **f1-score** | **support** | | --- | --- | --- | --- | --- | | **0** | 0.975806 | 0.975806 | 0.975806 | 124.000000 | | **1** | 0.909091 | 0.900901 | 0.904977 | 111.000000 | | **2** | 0.872000 | 0.865079 | 0.868526 | 126.000000 | | **3** | 0.929078 | 0.942446 | 0.935714 | 139.000000 | | **accuracy** | 0.922000 | 0.922000 | 0.922000 | 0.922000 | | **macro avg** | 0.921494 | 0.921058 | 0.921256 | 500.000000 | | **weighted avg** | 0.921846 | 0.922000 | 0.921902 | 500.000000 | |  |  |  |  |

**Prediction**

Since the highest accuracy score is for logistic model, we predict the price range using logistic model and saved it as a csv file.

test\_predict**=**log\_model**.**predict(test\_scaled)

result**=**pd**.**DataFrame(test\_predict)

result

| **0** |
| --- |
| **0** | 2 |
| **1** | 3 |
| **2** | 2 |
| **3** | 3 |
| **4** | 1 |
| **...** | ... |
| **995** | 2 |
| **996** | 1 |
| **997** | 0 |
| **998** | 2 |
| **999** | 2 |

1000 rows × 1 columns

result**.**to\_csv('result.csv')

As it is saved as csv file it automatically get downloaded as an excel file.

Next we have ranked the dataset by price range.

Get the latest updates from AIM

Top of Form

Bottom of Form

Top of Form

Bottom of Form