Product Analytics on Intel Xeon Processors

Abstract

Product analytics is a specialized application of business intelligence (BI) and analytical software that help manufacturers evaluate product defects, identify opportunities for product improvements, detect patterns in usage or capacity of products, and link all these factors to customers. Product analytics allows companies to fully understand how users engage with what they build.

Without analytics, the data they collect is often inconsistent and improperly formatted (known as unstructured data). Product analytics makes that data useful again by integrating all data sources into one single organized view.

Here, we have applied product analytics on data about Intel Xeon Processors to find various patterns and meanings from the data. We got various insights from the data and used these insights to make future predictions about the use of Intel Xeon Processors.

Requirements

Softwares required

- Python 3.5
- Any Python IDE (Pycharm, Jupyter Notebook or Spyder)

Libraries required

- **Pandas-** Pandas is probably the best tool around for reading/writing data and processing it. It converts our datasets into dataframes on which we perform a lot of data manipulation operations.

- **Scikit-learn-** Scikit-learn is a powerful Python library for machine learning. It contains function for regression, classification, clustering, model selection and dimensionality reduction. Here we intent it to used for building the regression model in which the target value is expected to be the linear combination of the input variables

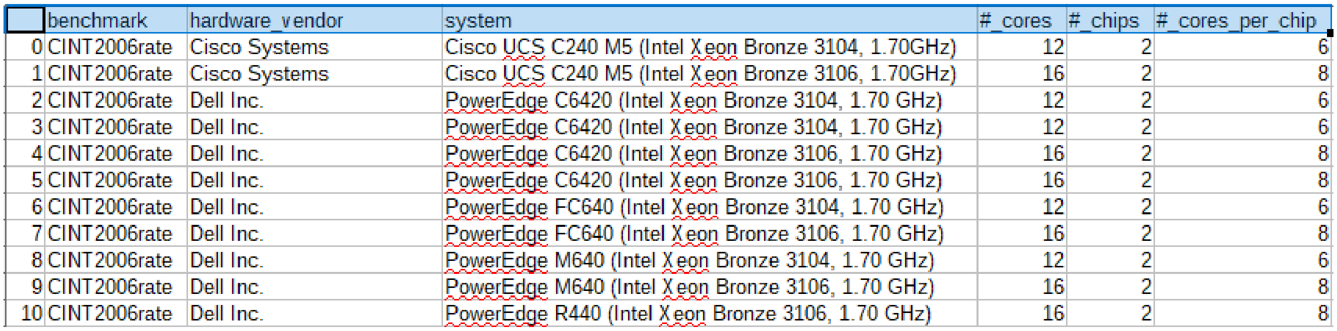
- **Seaborn-** Adds python support for visualization of data with the help of various graphs, charts, plots etc.

Dataset

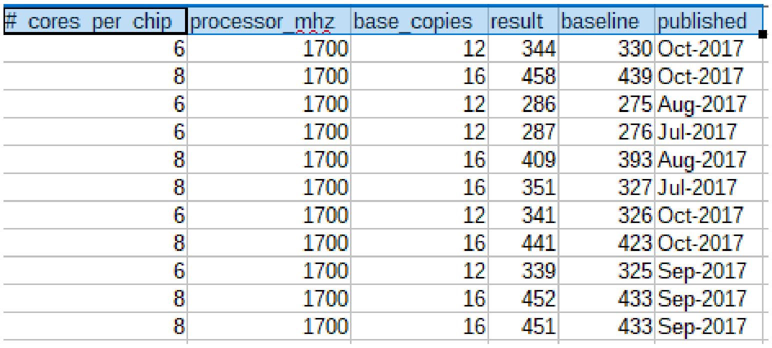
Link- https://www.kaggle.com/miniushkin/intel-xeon-scalable-for-capacity-planning/data

Screenshots of dataset

Bronze:



	benchmark	hardware_vendor	system	#_cores	#_chips	#_cores_per_chip
0	CINT2006rate	Cisco Systems	Cisco UCS C240 M5 (Intel Xeon Bronze 3104, 1.70GHz)	12	2	6
1	CINT2006rate	Cisco Systems	Cisco UCS C240 M5 (Intel Xeon Bronze 3106, 1.70GHz)	16	2	8
2	CINT2006rate	Dell Inc.	PowerEdge C6420 (Intel Xeon Bronze 3104, 1.70 GHz)	12	2	6
3	CINT2006rate	Dell Inc.	PowerEdge C6420 (Intel Xeon Bronze 3104, 1.70 GHz)	12	2	6
4	CINT2006rate	Dell Inc.	PowerEdge C6420 (Intel Xeon Bronze 3106, 1.70 GHz)	16	2	8
5	CINT2006rate	Dell Inc.	PowerEdge C6420 (Intel Xeon Bronze 3106, 1.70 GHz)	16	2	8
6	CINT2006rate	Dell Inc.	PowerEdge FC640 (Intel Xeon Bronze 3104, 1.70 GHz)	12	2	6
7	CINT2006rate	Dell Inc.	PowerEdge FC640 (Intel Xeon Bronze 3106, 1.70 GHz)	16	2	8
8	CINT2006rate	Dell Inc.	PowerEdge M640 (Intel Xeon Bronze 3104, 1.70 GHz)	12	2	6
9	CINT2006rate	Dell Inc.	PowerEdge M640 (Intel Xeon Bronze 3106, 1.70 GHz)	16	2	8
10	CINT2006rate	Dell Inc.	PowerEdge R440 (Intel Xeon Bronze 3106, 1.70 GHz)	16	2	8



#_cores_per_chip	processor_mhz	base_copies	result	baseline	published
6	1700	12	344	330	Oct-2017
8	1700	16	458	439	Oct-2017
6	1700	12	286	275	Aug-2017
6	1700	12	287	276	Jul-2017
8	1700	16	409	393	Aug-2017
8	1700	16	351	327	Jul-2017
6	1700	12	341	326	Oct-2017
8	1700	16	441	423	Oct-2017
6	1700	12	339	325	Sep-2017
8	1700	16	452	433	Sep-2017
8	1700	16	451	433	Sep-2017

Dataset Column Description

- **hardware_vendor-** Name of the of hardware vendor.
- **system-** Name of the processor system with its processor series.
- **#_cores-** Number of cores in the system.
- **#_chips-** Number of chips in the system.
- **#_cores_per_chip-** Number of cores per chip in the system.
- **published-** Month and Year the system was published.

..Similarly for Silver, Gold and Platinum series

Machine Learning Algorithm for Prediction

K-Nearest Neighbor-

K-Nearest Neighbors is one of the most essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection. K-nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

Algorithm

- A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor.

- Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

Here, we have use KNN algorithm to predict which series has the most demand for any particular month in future, we preferred KNN algorithm over other machine learning in this scenario because KNN provided us an option to limit the number of data which will be used in prediction, using only the series which were used in the recent time are used to predict the upcoming demand and the outdated series which might have harmed our prediction are ignored.

Working

- All the unnecessary columns in the dataset are removed and required one’s are selected for further processing. (Example-Benchmark column was removed as it was not required)

- Dataset structure is formatted and in way in which it can be handled by the software (Example- Spaces within the column names were replace by underscores)

- New fields introduced in the dataset using fields columns to easily manage the data.(Example- Days column was introduced using published column to help analyze the data and make future predictions)

- Data analysis is performed to discover various insights about the data and the outputs are provided.(Example- Most used series, System count for each hardware series etc.)

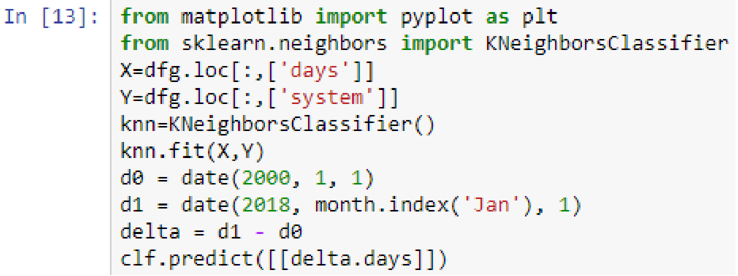
- Some of the insights are presents with the help of visual graphs and plots for easier understanding(Example- Example- Most used series, System count for each hardware series for each type processors etc.)

- Machine Learning Algorithm (KNN) is used to make future prediciton about the Intel Xeon Processors(Example- which series has the most demand for any particular month in future)

Output

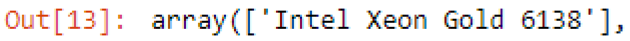
Predict series most in demand.

Input (Month and Year)- Jan, 2018



```
In [13]: from matplotlib import pyplot as plt
         from sklearn.neighbors import KNeighborsClassifier
         X=dfg.loc[:,['days']]
         Y=dfg.loc[:,['system']]
         knn=KNeighborsClassifier()
         knn.fit(X,Y)
         d0 = date(2000, 1, 1)
         d1 = date(2018, month.index('Jan'), 1)
         delta = d1 - d0
         clf.predict([[delta.days]])
```

Output (Series)- Intel Xeon Gold 6138



```
Out[13]:  array(['Intel Xeon Gold 6138'],
```