POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Characterizing skyline points through Indicators

TESI DI LAUREA MAGISTRALE IN
COMPUTER SCIENCE AND ENGINEERING

Author: **Nitish Kashinath Viraktamath**

Student ID:	10713501
Advisor:	Davide Martinenghi
Co-advisor:	-
Academic Year:	2021-22

Abstract

This paper aims to show that there are certain points in the skyline that cannot be found using a top-1 linear function but are interesting to consider from a human standpoint.

The skyline represents the set of points that are not dominated by any other point in the dataset. A data point say A is said to dominate another data point B if A is not worse than B on all the attributes and A is strictly better than B in at least one of the attributes. The set of points not dominated forms the skyline of a dataset. The main objective here is to find those skyline points that provide a good choice from the user's perspective and to show that such points exist in the skyline.

In order to prove that such points exist, the skyline of the dataset is calculated using the Block Nested Loop Algorithm. New indicators – Grid Strength, Increase in Domination Volume, Concavity Index and Maxrank are being introduced. These indicators are calculated for the skyline points present in several datasets. We make conclusions on the nature of the skyline points based on the values of the above indicators. In addition to the various experiments, we deduce the possible relationship between the different indicators and find a correlation between them that would help further explain the nature of the skyline points. Datasets of varying dimensions and size are analyzed.

Through our experiments and the results obtained we try to find these interesting points and pave way for a deeper investigation of these skyline points in the dataset.

Key-words: Grid Strength, Domination Volume, Maxrank, Concavity Index

Abstract in lingua italiana

Questo articolo mira a mostrare che ci sono alcuni punti dello skyline che non possono essere trovati utilizzando una funzione lineare top-1, ma sono interessanti da considerare dal punto di vista umano.

Lo skyline rappresenta l'insieme di punti che non sono dominati da nessun altro punto nel set di dati. Si dice che un punto dati dica che A domini un altro punto dati B se A non è peggiore di B su tutti gli attributi e A è rigorosamente migliore di B in almeno uno degli attributi. L'insieme dei punti non dominati forma lo skyline di un dataset. L'obiettivo principale qui è trovare quei punti dello skyline che forniscono una buona scelta dal punto di vista dell'utente e mostrare che tali punti esistono nello skyline.

Per dimostrare che tali punti esistono, lo skyline del set di dati viene calcolato utilizzando l'algoritmo Block Nested Loop. Nuovi indicatori: vengono introdotti la forza della griglia, l'aumento del volume di dominazione, l'indice di concavità e il rango massimo. Questi indicatori sono calcolati per i punti dello skyline presenti in diversi dataset. Traiamo conclusioni sulla natura dei punti dello skyline in base ai valori degli indicatori di cui sopra. Oltre ai vari esperimenti, deduciamo la possibile relazione tra i diversi indicatori e troviamo una correlazione tra loro che aiuterebbe a spiegare ulteriormente la natura dei punti dello skyline. Vengono analizzati set di dati di dimensioni e dimensioni variabili.

Attraverso i nostri esperimenti e i risultati ottenuti cerchiamo di trovare questi punti interessanti e di aprire la strada a un'indagine più approfondita di questi punti dello skyline nel set di dati.

Parole chiave: Forza della griglia, Volume di dominazione, Maxrank, Indice di concavità

Contents

1. Chapter one

Introduction

Big Data today has made it increasingly important and difficult to extract useful information from large data. Information is needed to be retrieved according to user preferences and in an appropriate size. Top-k queries retrieve a set of results according to a specific scoring function. The best 'k' tuples/records are the output of a top-k query. The k best tuples with the highest scores are retrieved. The Skyline queries employ the notion of dominance between records/tuples which makes it more efficient than a top-k query. A tuple is said to dominate another tuple if it is equal or better than the other tuple in all the features/attributes and is strictly better in at least one of the features. Therefore, the skyline consists of the set of non-dominated tuples/records.

There are numerous examples where it is important to find the best results from a large set of data. For instance, a user has to select among various hotels based on criteria like distance from the center, price and rating. The best hotels have to be retrieved from a huge number of hotels in the database. As the number of criteria increase it becomes increasingly difficult to provide a result that matches the user's criteria. More importantly, we do not know how the user weighs/values each of the criteria. For some users price may be a strong indicator while others might consider distance from the center to be an important factor in deciding the best hotel. The problem becomes complex as users do not exactly know what criteria they actually prefer which results in varying weights for each of the criteria. A strict assumption of the weights is not possible from a practical perspective. If the weights are fluctuated by a small value, then there might be drastic changes in the results that we might get. We must try to provide all the results according to varying weights.

We sometimes come across hotels that are not the topmost result but they make a good choice according to our criteria. Our focus is mainly on such hotels/points in the dataset. The skyline may contain points that provide a good compromise but are never top-1 for any given linear function so we must look for functions of a higher degree. We need to find the characteristics of such points in the skyline.

This thesis is divided into four chapters. In the first chapter we discuss the problem at hand, the objective, the background related to the problem and the work related to Skyline and Maxrank. In the second chapter the new indicators – Grid Strength, Domination Volume and Maxrank are being introduced along with a few examples. The third chapter contains all the datasets that were experimented upon including the description, the values of the indicators and summary plots for each of the dataset analyzed. The fourth and final chapter consists of the conclusions made, bibliography and references used for this thesis.

1.1 Problem Statement

In the era of Big Data today, it is important to extract and filter out useful information from a large set of data. When we want to book a hotel or book a flight ticket or when we want to select something given a certain set of criteria it becomes difficult to choose when there are potentially a large set of options with us. Skyline queries make the process of finding the best hotel/flight or information easier by narrowing down the best options given a set of criteria. With the result from the skyline query, it makes our job easier to choose which flight/hotel to book.

However, there are certain skyline points that are not top-1 for any linear function but provide a relatively good choice from a human standpoint. These interesting points could be overlooked by the user just because it is not the top result based on the given linear function. Consider the below graph having 4 skyline points with the target or the best-case point being the co-ordinates (0,0) -:

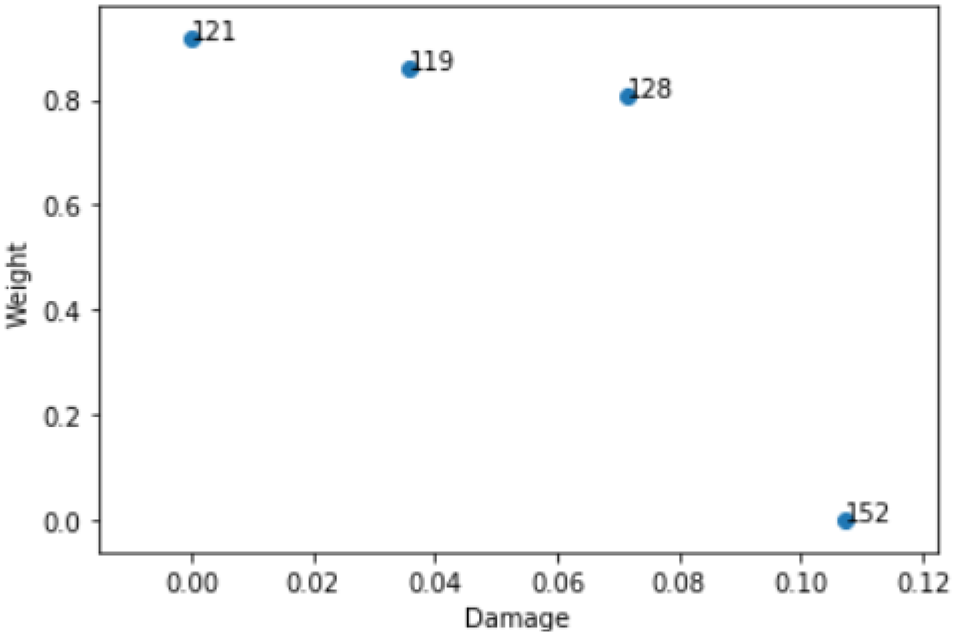


Figure 0.1: Skyline plot – Skyrim Dataset

If we start drawing lines starting from the origin to the skyline points then we can imagine that the points 121 and 152 can be reached with a linear line which corresponds to a linear function. This is not applicable for the points with ids 119 and 128 which may need a curve or a corresponding higher degree function, higher than 1. We need to find the characteristics of such skyline points that are not top-1 for a linear function and we need to know what makes such points interesting and whether they are top-1 for some other function.

1.2 Objective

To tackle the above problem, new indicators which determine certain characteristics of the skyline points are being introduced. The indicators are as follows -:

1.Grid Strength

2.Increase in Domination Volume

3.Maxrank

Using the three indicators, their values for each of the skyline points are being noted and with the help of these values and data visualization using bar plots we determine which of the skyline points might be interesting to consider. For instance, a point with a high value for the Grid strength, a high value for Domination volume and a considerable value for the Maxrank indicator would be a suitable candidate point that we are looking for. We look into the basic skyline plot in order to determine candidate points visually by seeing if a linear line can be drawn to reach them and then we confirm our predictions by showing the experimental results.

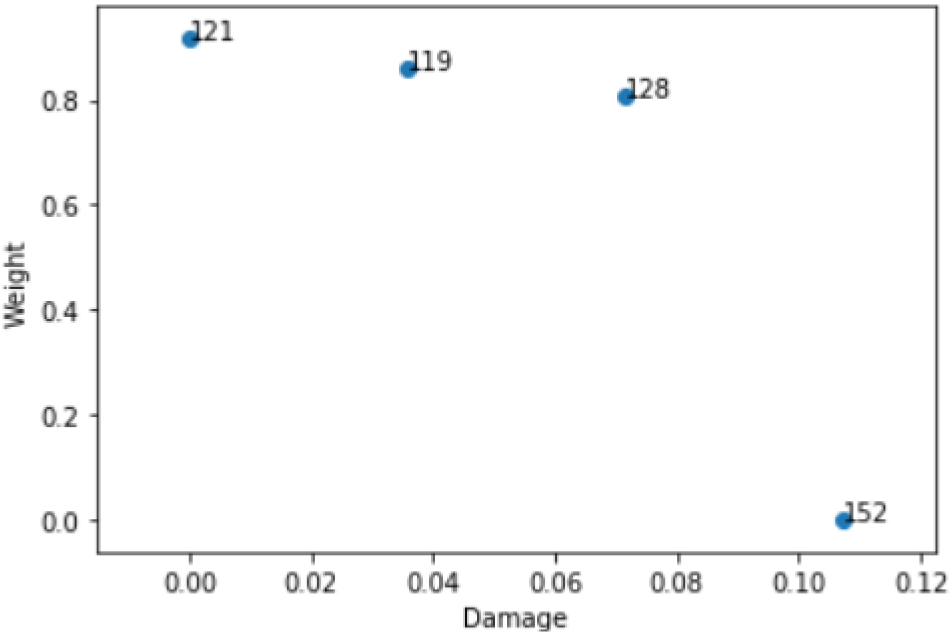


Figure 0.2: Skyline plot- Skyrim Dataset

Our main goal here is to calculate the indicators for all the skyline points and see if there are certain points based on the values of the indicators that are interesting and also to possibly establish a pattern/correlation that would help us easily pinpoint the points in the skyline. As we can observe from the plot above, we calculate the indicator values for each of the points – 121,119,128 and 152. Depending on the values of the indicators we make conclusions about the nature of the skyline points, possibly the points 119 and 128 are a good choice. 152 point seems like the worst skyline point whereas 121 point seems like the best skyline point.

1.3 Background

1.3.1 Top-k Queries and Skyline Queries

Top-k Queries: The top-k queries aim to return the 'k' best tuples from a database based on a particular scoring function. The score is computed for each tuple in the database and the tuples are sorted according to the respective scores. Suppose that we have the data of 4 football players with two attributes – Number of goals and Number of Assists provided in a year. The best player would be the player with the highest number of goals and assists. The data of the four players is given below -:

Player	Goals	Assists
Lionel Messi	20	6
De Bruyne	14	12
Cesc Fabregas	12	10
Zidane	15	4

Table 0.3.1: Players with Goals and Assists

Now the scoring function according to what we want would be:
S(tuple) = Goals + Assists
S (Lionel Messi) = 20 + 6 = 26
S (De Bruyne) = 14 + 12 = 28
S (Cesc Fabregas) = 12 + 10 = 22
S(Zidane) = 15 + 4 = 19
According to the above calculations, if we want to retrieve the top-2 players then the result would be Lionel Messi and De Bruyne with the scores of 26 and 28 respectively. Top-k queries does return the k-best tuples with the highest score but it carries a disadvantage of being too inefficient when the data is too large. In the above case we just had 4 tuples but in the case of thousands of tuples calculating the score of each tuple and sorting them accordingly would be time consuming and cumbersome. The functions involved in top-k queries can also vary. For instance, in this case we have the scoring function = Goals +

Assists. In some other case perhaps, we want to give more importance to Goals than to Assists, we can modify the scoring function accordingly.
Example: Score = 0.8*Goals + 0.6*Assists. Choosing the right weights for the scoring function becomes a difficult task as we want to adhere to a specific criterion. In the previous case, both Goals and Assists were given equal importance but here the number of Goals was considered to be more valuable than the number of assists. Changes in the weight value of the criteria leads to a variation in the result set. We might see a slight change in the result and might get players other than Messi and De Bruyne.

Skyline Queries: The top-k queries are inefficient and time consuming for large datasets. The skyline introduces the concept of dominance between tuples. A tuple A dominates a tuple B if-:

1. **A is better than B in at least one attribute**
2. **A is not worse than B in any of the attributes**

In formal terms, let A denote the set of attributes and let t and t′ be two tuples in the dataset. Then we have the tuple t dominate tuple t′ if and only if -:

t >= t′ for every attribute/feature a that belongs to the set A and

t > t′ for at least one attribute/feature say b that belongs to the set A

For example, if we have a dataset of 5 hotels with the distance attribute and a rating attribute (rating out of 5). We are looking for hotels with the least distance and the highest rating.

Hotel	Distance	Rating
A	50	4
B	40	5
C	70	2
D	60	3
E	65	5

Table 0.3.2: Hotels with Distance and Rating

In the above example Hotel, A would dominate Hotel C and Hotel D because distance(A) < distance(C) and distance(A) < distance(D). Also Rating(A) > Rating(C) and Rating(A) > Rating(D). Hotel B dominates Hotel A, C, D and E. Hotel E dominates Hotel C as distance(E) < distance(C) and Rating(E) > Rating(C).

Therefore, we have-:

1. A dominates C and D
2. B dominates A, C, D and E
3. E dominates C

The tuple that belongs to the skyline is the one that is not dominated by any other tuple in the database. Here Hotel B would belong to the skyline. Therefore, the skyline consists of the set of non-dominated points in the dataset. When the dataset is too large or when there are too many dimensions to consider, the skyline of a dataset can become quite large and it would not be meaningful to a user. Skyline cardinality reduction techniques have also been introduced which are aimed at reducing the size of the skyline. The **k-skyband** refers to a set of points that have been dominated by fewer than k others. There are various algorithms used to calculate the skyline. In our thesis we have used the BNL algorithm which uses a window of potential skyline points. We compare each point in the dataset with the points in the window, if the point in the window is dominated by say a point p, then this particular point in the window is removed and is being replaced by the point p. We do this calculation for every point in the dataset until we get a set of non-dominated points.

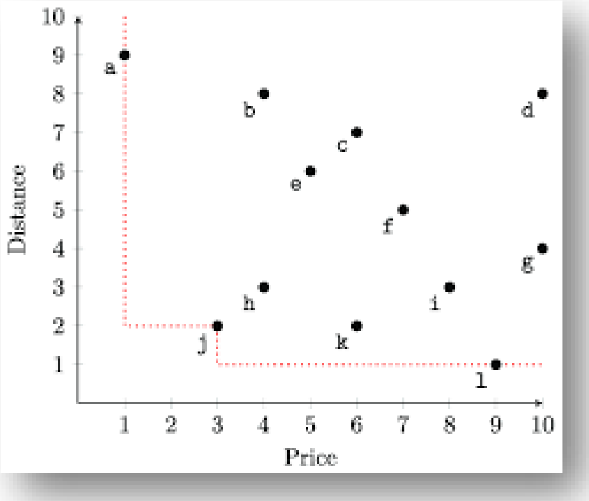


Figure 1.3: Hotel dataset

Shown above is a plot between distance and price for a sample dataset of hotels. The hotels are from a to l. The skyline can be seen as the hotels a,j and l represented by the red dotted line in the plot. The cardinality of the skyline is equal to 3 which denotes the number of points in the skyline.

Skyline and Top-k Queries: There is a relationship between the skyline and top-k queries in particular with 1-NN (nearest neighbor) query. Suppose we have a tuple t which is the only result of a NN-query then it belongs to the skyline and if tuple t is a point on the skyline, then there exists at least one monotone function d that is minimized by only t. There are two key observations with respect to skyline and top-k queries. Suppose we have R which denotes a relation, MD which is a set of monotone distance functions then we have the following observations -:

1. If t is the only result of a 1-NN query for some monotone distance function d that belongs to MD, then t is the tuple/record that belongs to the skyline
2. If t is assumed to be a skyline point, then there is at least one monotone distance function d that belongs to MD which is minimized by t.

Therefore, skyline points are also called potential 1-NN points as they are winners in a 1-NN scenario. This applies not only to monotone distance functions but also to monotone scoring functions.

1.3.2 Concave and Convex Skyline Points

The convex hull is defined as the smallest polygon that can enclose a set of points. A **concave** skyline point is a skyline point that does not lie on the convex hull, therefore it cannot be found using a top-1 linear function. A **convex** skyline point is a skyline point that lies on a convex hull and therefore can be found using a top-1 linear function.

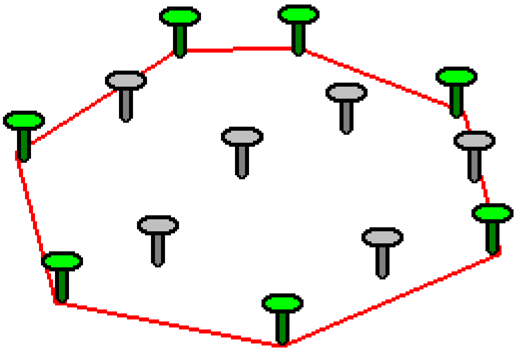


Figure 1.4: Convex hull of nails

The above figure is a simple representation of a convex hull composed of nails. The green nails represent the border of the convex hull. This encloses the other nails which are present inside the convex hull. We can imagine the convex hull to be a rubber band which is stretched until it encloses the nails in the grey color. This band would then adjust itself so as to reduce its distance to the nails next to each other. The convex hull is an important indicator of whether the points can be accessed using a top-1 linear function or would need functions of a higher order. If we have points that lie on the convex hull then it is likely that those points can be found using a top-1 linear function. If the point does not belong to the convex hull, then probably, we need functions of a higher degree say 2, 3...etc. to find such points.

2. Related Work

2.1 Skyline Cardinality Reduction Techniques

The skyline operator/skyline query returns the set of non-dominated objects in a dataset. As the number of dimensions/attributes of a dataset increase, the number of skyline points for that particular criterion increases and it becomes less meaningful to the user. **Skyline cardinality reduction** techniques have been employed to reduce the size of the skyline output. Given below are few of the techniques recently used in this direction -:

1. **Pointwise Ranking**
2. **Subspace reference**
3. **Set-wide maximization**
4. **Other techniques**

1. Pointwise Ranking

In this particular technique the datapoints are being sorted according to their score in the scoring function. The data points with the topmost scores are being retrieved as the result. The **F-Skyline** is related to the concept of Pointwise Ranking. Before delving into the definition of F-Skyline, it is essential to define the concept of **F-Domination**. Let F be the set of all the scoring functions. A tuple t is said to F-Dominate another tuple t′ if and only if t is better or equal to t′ with respect to all the scoring functions in F and t is strictly better than t′ in at least one of the scoring functions in F [11]. We also introduce the concepts of **ND** and **PO** which stands for Non-F-Dominated and Potentially Optimal Tuples. The ND set denotes all the points that are non-F-Dominated and the PO set denotes those points that score the highest with respect to some monotonic scoring function [11]. There is an intersection between the ND and PO sets when F represents the set of all monotone functions. There is a distinction between the ND and PO sets only when we consider a subset of F.

2. Subspace reference

The primary focus here is on the partitional subspaces of an entire dimensional space. The concepts related to subspace reference are -:

1. **Skyline frequency** – The number of subspaces in which a point is a skyline point is being calculated [11]
2. **Top-delta-dominant skyline** – Here we need to find the smallest k such that there are more than k points in the k-dominant skyline. The k-dominant skyline has a requirement that is more relaxed compared to the original skyline definition [11]
3. **Strong skyline point** – Zhang introduced the concept of strong skyline points. If we consider a space called F, then a subspace of F is said to be a delta-subspace if it's skyline contains less than delta points. The union of all the skyline points in all the delta-subspaces makes up the strong skyline points[11].

4. Set-wide Maximization

In order to satisfy a common quantitative goal, we deliberately select a subset of the skyline points. There are two concepts related to the set-wide maximization

- **Number of dominated points** – There are k points selected from the skyline so that the number of points dominated by these points are being maximized [11]
- **Distance based skyline representative point** – The distance between a non-representative skyline point and a skyline point is being minimized[11].

5. Other techniques

- **K-regret Query:** This selects a subset S of the entire dataset D in such a way that the regret ratio is being minimized[11].
- **Skyline ordering**: This technique splits the dataset into several partitions say S1, S2, S3…Sn such that each subset say- S2 = S1/S[11]. Therefore, these subsets are used for the set-wide maximization techniques.

- **Skyband based ranking**: This retrieves the k-skyband from the dataset. The **k-skyband** refers to the set of points that are dominated by fewer than k others. We return the best possible points from the k-skyband[11].
- **Skyrank** : This represents an interesting characteristic of a skyline point. The skyline point is said to be interesting if it manages to dominate the greatest number of points in various subspaces [11].

2.2 Algorithms for computing Skylines

The paper on 'Skylines Front and Back' discuss a few of the used skyline algorithms.There has been a lot of work focused on skyline algorithms. We discuss a few skyline algorithms here -:

1. **Block Nested Loop Algorithm**: The Block Nested Loop Algorithm or BNL works on the NL principle. Suppose we have a window W containing potential skyline points, a tuple t in the database is compared with the tuples in the window W, if the tuple t is dominated then it is discarded. If tuple t dominates any tuple in the window W, then it is inserted into the window and the tuples dominated by t are being removed from the window. Empirical evaluation of BNL shows its high dependency on how the tuples are distributed, the worst case being when they are negatively correlated.[1] The effectiveness of BNL also depends upon how the tuples are processed and the size of the window. There is a variation in the performance of BNL when we consider an unlimited window size where $|W| > |N|$ where N represents the number of tuples and when we consider a window size equal to one.
2. **Sort Filter Skyline:** The Sort Filter Skyline (SFS) is identical to the BNL algorithm with the difference being that the tuples are first sorted according to a monotone scoring function say f. This makes SFS a better algorithm than BNL as – SFS is progressive algorithm as if a tuple t belongs to the window, then it is guaranteed to be in the skyline and can be immediately outputted [6].
   -The number of passes in SFS is optimal [6]
   -If tuple t and tuple s are dominated then they are not being compared to each other which leads to a reduction in comparisons and a more efficient algorithm.

3. **LESS and SaLSa** : The LESS and SaLSa[6] algorithms are a modification of the SFS algorithm. LESS primarily focusses on anticipating the dominance tests in the sorting phase so that the sorting costs can be reduced. The SaLSa algorithm does not read the entire sorted input relation [6].
   Let <t1, t2, t3…tn> be the order in which the tuples are processed with ti (i<n) being the last fetched tuple. Since the function f used to sort the tuples is monotone, we have the relation f(ti) >= f(tj) for every j that is greater than i. Therefore, all the unread tuples correspond to a bounded region BR. Hence if there exists a tuple tj that Pareto-dominates BR then the algorithm can be halted since no more tuples would enter the skyline.
4. **Divide and Conquer Algorithm:** The Divide and Conquer Algorithm [6,18] works on the recursive partitioning of the attribute sets. Suppose that we have partitioned two attributes Ai and Aj and we get the following sets (SHI, HJ) (SLI, LJ) (SHI, LJ) (SLI, HJ) where the sets (SHI, LJ) and (SLI, HJ) need not be compared. This fact along with a suitable merging scheme leads to a worst case sub-quadratic complexity scenario. The Divide and Conquer takes an m-partitioning as compared to two taken previously. Though the Divide and Conquer has a worst case sub-quadratic complexity it cannot be said that it is better than other algorithms in terms of actual running time as there are many factors that need to be taken into consideration.
5. **LS-B Algorithm:** The LS-B Algorithm [6] is a lattice-based approach used for cases in which attributes have a low cardinality. Low cardinality here means that the attributes have a small range of values, for example it can be rating of hotels or ratings of movies which usually have a small range (0-5) or (1-10). In these low cardinality cases LS-B has been introduced. The LS-B Algorithm builds a complete lattice of points and marks each of them as not-present(np), then it scans the entire number of points and marks them as present(p). It identifies all the p-values that are non-dominated and hence are present in the skyline. A second scan will retrieve all the skyline points in the database. The LS Algorithm is a modification of the LS-B algorithm.
6. **Basic Distributed Skyline Algorithm** – The Basic Distributed Skyline Algorithm (BDS)[6] works in a distributed scenario where we have skyline attributes that are distributed over multiple sites wherein each site provides a partial view of the alternatives. The BDS algorithm is derived

from Fagin's Algorithm for multi-objective skyline processing. The BDS Algorithm working is given below -:

- Input: instance r vertically partitioned in d sorted lists L1,…Ld
- Output – skyline (Ra)
- Cyclically perform sorted accesses on the d lists until (at least) one object say Os is retrieved from all the lists.
- For all the objects O that have been fetched from at least one list, perform random accesses to retrieve the missing attribute values
- Perform the necessary dominance tests and return the non-dominated objects.

7. **Horizontal Fragmentation**: [2] When a relation say r is horizontally fragmented over a cluster of P servers r = r1 U r2 U ….rp , the skyline can be computed by exploiting the identity – skyA(r) = skyA(skyA(r1) U …. SkyA(rp))[6]. We first compute the local skylines and then merge the results as was done in the divide and conquer methodology.

2.3 Other types of Skyline Queries

Probabilistic Skyline Queries – Skyline queries retrieve results according to specific user criteria. It is difficult to give a rigid estimate to the user preferences and assign weights to the attributes. We need to retrieve all the tuples which are output with varying weight values. The paper on Probabilistic skyline queries by Christian Bohm, Frank Fiedler and Annahita Oswald discuss skyline queries for uncertain objects in the database. [3] Uncertainty is a natural factor in many applications, which is not restricted to moving objects. One of the classical examples pertaining to uncertain data modelled by probability density functions are sensor networks. The data retrieved from sensors is uncertain and usually have some error in measurement. The sensors that provide accurate data are expensive. [3] Most complex decision processes based on real world data involve uncertainity which can be effectively supported by skyline queries on uncertain data. There are three main definitions introduced in the paper -:

1. **Probabilistic Dominance** – Let f(x) and g(y) be uncertain objects in the database. The probability that f(x) dominates g(y) corresponds to the probability that x generated by the probability density function f(x) dominates y generated by probability density function g(y)

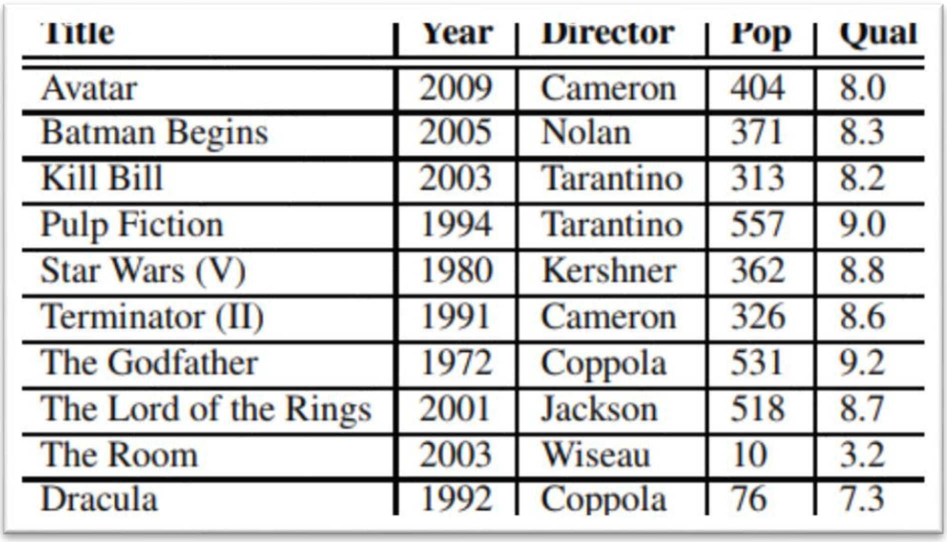
2. **Skyline Probability** – The skyline probability[3] of an uncertain object f(x) that belongs to the database corresponds to the probability with which x taken from the PDF f(x) is not dominated by any of the y each of which has been independently taken from one of the uncertain remaining objects stored in the database g(y) belongs to DB′ where DB′ = DB/f(x)
3. **T-Skyline** – Let T – [0,1] be a threshold, the T-Skyline[3] of a database is of uncertain objects is the set of objects for which the following property holds -: ST = {f(x) belongs to DB | PS(f(x)) >= T }

2.4 Parallel Computation of Skyline Queries

The paper on '**Parallel Computation of Skylines'** by Louis Woods, Gustavo Alonso and Jens Teubner explores processing skyline queries from a hardware perspective. Skyline queries have been extensively studied in software but their paper is one of the first papers to discuss skyline computation with hardware [8]. They propose a way to compute the skyline using **FPGA's** that handle arbitrary dependencies and has no restrictions on the size of the intermediate results [8]. A small set of potentially skyline tuples are being kept in the FPGA and the rest of the tuples are considered as an input data stream that propagates through the FPGA. **Pipeline parallelism** and **nearest neighbor communication** is used for concurrent manipulation of the working set which combines data organization, computational power and synchronization into a parallel processing model which makes use of the FPGA's capabilities [8].

2.5 Aggregate Skyline Operator

The paper on '**From Stars to Galaxies: skyline queries on aggregate data'** by Matteo Magnani and Ira Assent aims to combine the notion of skyline and aggregate queries to form the **aggregate skyline** operator. [4]. This operator can be used to express queries in the form: return the best groups depending on the features of their elements, and thus provides a powerful combination of grouping and skyline functionality. The paper introduces the skyline aggregate operator which performs better than its equivalent in SQL up to two orders of magnitude[4]. An example of dominance between groups of records can be movies taken from IMDB website. We have listed movies with their popularity and their quality and we want to essentially find which groups of movies by directors dominate each other. A **group dominance [4]** would help us determine which director's movie we would prefer to watch given two sets of movies. A table from the paper is given below -:



Title	Year	Director	Pop	Qual
Avatar	2009	Cameron	404	8.0
Batman Begins	2005	Nolan	371	8.3
Kill Bill	2003	Tarantino	313	8.2
Pulp Fiction	1994	Tarantino	557	9.0
Star Wars (V)	1980	Kershner	362	8.8
Terminator (II)	1991	Cameron	326	8.6
The Godfather	1972	Coppola	531	9.2
The Lord of the Rings	2001	Jackson	518	8.7
The Room	2003	Wiseau	10	3.2
Dracula	1992	Coppola	76	7.3

Figure 2.1: Movie ratings - IMDB

The calculation of the groupby followed by the skyline or vice-versa would not result in an optimal output of the dominance between groups so the Aggregate skyline operator is introduced which is the extension of the skyline operator on sets of records.

Aggregate skyline: Let Ug be a set of groups of records. An Aggregate skyline [4] is the set of all groups in Ug not dominated by any other group.

The definitions of strict dominance and weak dominance is described below -:

Strict dominance [4]: If we take two directors – Tarantino and Wiseau, then Tarantino strictly dominates Wiseau as all of Tarantino's movies are better than Wiseau in terms of Population and Quality. The worst movie by Tarantino is better than any movie of Wiseau, therefore we can safely conclude that Tarantino dominates Wiseau.

Weak dominance [4]: If we take the two directors Tarantino and Kershner then we can no longer conclude that one is better than the other. One movie by Kershner called Star Wars(V) is better than one movie by Tarantino. But in general, Tarantino is better than Kershner but we cannot say that given two sets of movies – one by Tarantino and another by Kershner that we prefer

Tarantino over Kershner. This illustrates the concept of weak dominance between groups of tuples/records.

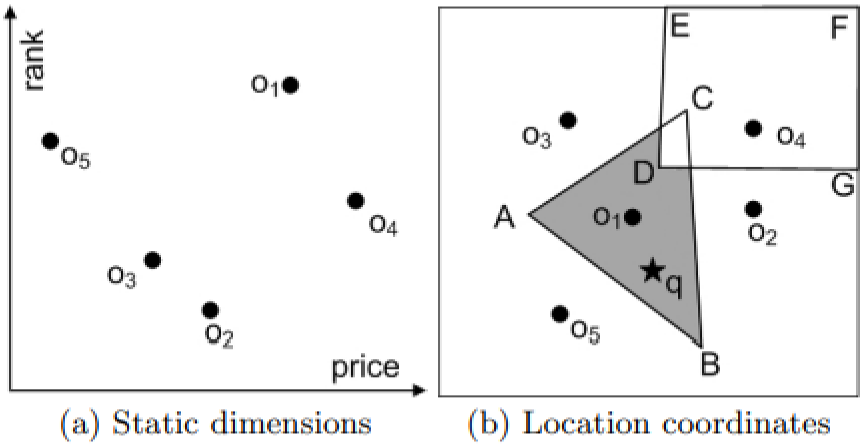
T-Dominance: The concept of T-Dominance [4] is being introduced wherein we compare two sets of records/tuples by randomly taking one record from each set and checking for dominance. If we have two sets of movies by two directors, we would randomly pick one movie from each director and compare them and then calculate the T-value. A higher T-value would indicate that one dominates the other, we need to set the right threshold for T[4].

2.7 Moving Skyline Queries

The paper on '**A Safe Zone Based Approach for Monitoring Moving Skyline Queries'** by Muhammad Amir Cheema and Xeumin Lin proposes the concept of a safe zone to address the problem of a continuous moving skyline query [10]. A safe zone [10] is basically a region within which the results of a query remain unchanged. It is necessary to update the results when we move out of the safe zone. For example, if we are searching for the best restaurant in terms of price and distance while driving in a car, a **Nearest Neighbour[10]** query would return results of all the hotels that is close to the current location without taking into account the price and distance to our location, whereas a skyline query would retrieve the best results that match our criteria. The concept of moving skyline queries [10] is introduced here as the query keeps updating continuously and therefore, we must continuously update our results. A few definitions in relation to the safe zone are given below -:

1. **Complete Dominance:** An object say o completely dominates [10] an object o′ if object o is better than object o′ in all the dimensions and it is strictly better than o′ in at least one of the dimensions.
2. **Skyline Query:** The skyline query returns the set of objects that are non-dominated.
3. **Static equality:** An object o is said to be statically equal to object o′ if o is equal to o′ on every dimension considered [10].
4. **Static Dominance:** An object o is said to statically dominate another object o′ if o is not static equal to o′ on every dimension [10].
5. **Affecting set:** The affecting set of an object o is all the other objects say o′ which are dominated by the object o [10].

6. **Impact region**: The impact region [10] of an object o denoted by IR(o) is all the points say x, for which the NN(x,A(o)) = o, which means that every point x for which o is the closest object in A(o).
7. **Safe Zone:** The safe zone [10] consists of every point that lies inside the impact region of every skyline object and it lies outside the impact region for every non-skyline object.



(a) Static dimensions (b) Location coordinates

Figure 2.2: Location of co-ordinates

The above figure shows the static dimensions and location coordinates for the objects. The Impact region for object o1 is the region triangle ABC and the Impact region for object o4 is the square region DEFG [10].

2.8 Skyline and Ranking Queries Combined

The paper titled – '**Reconciling Skyline and Ranking Queries'** by Davide Martinenghi and Paolo Ciaccia introduces the concept of restricted skylines or **R-Skylines** [1] that aims to combine the notion of skyline and ranking queries in order to accommodate varying weight values in the user preferences [1]. There are three main approaches used to retrieve results from a large set of data with attributes -:

1. **Ranking queries** – This uses the concept of scoring functions to rank tuples/records [1]. The drawback of these queries is that it is difficult to determine the weights in the scoring functions.

2. **Lexicographic approach** – This narrow approach assumes a dominance between attributes thereby considering an attribute to be more important than the others.
3. **Skyline queries** – The skyline is the set of non-dominated tuples/records in the dataset. The drawback with the skyline is that the output size cannot be controlled.

   To combine skyline and ranking queries – R-skylines [1] have been introduced. R-skylines are able to handle varying weight values and imply constraints on the weights. Two R-Skyline operators have been introduced – **ND and PO [1]**. Before delving into the operators, it is essential to explain the concept of **F-dominance**. A tuple t is said to F-dominate another tuple t′ if t is at least equal or better than t′ in all the scoring functions in F and is strictly better than t′ in at least one of the scoring functions in F.

   The two operators' description is given below -:

   1. **ND** – The ND set stands for the set of non-F-dominated tuples [1]
   2. **PO** – The PO stands for potentially optimal tuples/records. It is the set of records that are the top result for some scoring function in F [1].

   The paper explores various algorithms which are efficient in computing the above two opeators.

2.9 Uncertain Top-k Queries

The paper – **'Exact Processing of Uncertain Top-k queries in Multi Criteria Settings'** by Kyriakos Mourtadis and Bo Tang focusses on the problem **of user preferences** being input to the dataset [2]. A traditional top-k query would retrieve the top-k results according to a scoring function for which specific weights are given. The paper states that weights or user preferences cannot be predicted accurately even if it is manually specified by the user. It is necessary to retrieve the top-k results with respect to a range in the weight values. The **Uncertain Top-k queries** [2] are introduced which given a set of weight values will retrieve all the results that may belong to the top-k set. A second version of UTK reports the exact top-k results for each possible weight setting. A practical example of the problem of specifying user preferences is given -:

Suppose we have to choose a car based on price and mileage. The user specifies a preference of say (0.3,0.7) giving more importance to mileage. The top-k query would easily retrieve all the results according to this weight criterion. But sometimes the user would want to check other cars with slight differences in the weights say mileage weight = 0.69, this would result in a different result set. It is difficult to retrieve all the top-k results with variations in the weight values. The uncertain top-k query expands the weight vector into a region and returns the top-k results in relation to the region. Previous research mainly focused on uncertainty or unpredictability of data/records and attributes; the uncertainty of user preferences wasn't considered. The paper describes two versions of the UTK (uncertain top-k query) -:

1. **UTK (1)** – This reports all the top-k results according to a weight vector when it lies in a specific region R. The result-set obtained by UTK (1) is minimal [2].
2. **UTK (2)** – This reports all possible top-k results in the region R with varying values of weight vector [2]. The result-set produced by UTK (2) is a partition of region R with the top-k results of the weight vector for each partition.

Furthermore, for the UTK (1) problem, the **r-skyband** algorithm [866,879] is introduced and the **Joint arrangement algorithm** [866,879] is proposed for the UTK (2) algorithm.

The main steps in the r-skyband algorithm are -:

1. **Filtering step**
2. **Refinement step**
3. **Drill Optimization step**

The main step in relation to the Joint arrangement algorithm (JAA) is the **anchor choosing strategy** [2].

2.10 Skyline Queries in Multi Criteria Settings

[3]. The paper on **FA + TA < FSA** focuses on the issue of choosing tuples/records in a multi-criteria or multi-objective setting. FA stands for **Fagin Algorithm** which was one of the first algorithms to be developed for this purpose. A basic example of Fagin algorithm is shown below -:

A	1
B	2
C	3

C	1
B	2
A	3

B	1
A	2
C	3

Table 2.10.1: Fagin Algorithm

In the above tables the objects along with their scores are provided in the right-hand side. Assuming that all the three sources are equally trustworthy that is w = (1/3,1/3,1/3) we have for example the score of A equal to 0.33*1 + 0.33*2 +0.33*3 = 0.33+0.66+0.99 = 1.98. Now the weights for the above sources can be varied say w= (0.3,0.2,0.5) making source 3 to be a more trustable source. Hence, we have score of A equal to 0.3*1 + 0.2*2 + 0.5*3 = 0.3 + 0.4 + 1.5 = 0.7 + 1.5 = 2.2 where source 3 is given more importance. The FA algorithm is not instance optimal. The **TA algorithm** is introduced which uses a threshold value (tau) to perform sorted + random accesses to retrieve the points from the dataset. The TA Algorithm stands for Threshold Algorithm.It is more efficient than FA. In order to handle the problem of multi criteria selection, the **FSA algorithm** has been introduced. Before we go into the FSA algorithm, we need to introduce the concept of **weak F-dominance**. An object say A weakly F-dominates an object say B if and only if for every scoring function f(A) >= f(B) wherein the strict condition of f(A) > f(B) does not exist in this case. FSA algorithm stops when the last seen k-objects all F-dominate the threshold tau. The FSA algorithm uses F-dominance and not the weak F-dominance for calculating NDK.

2.11 Skyline Queries for varying user preferences

The paper titled **'Flexible Skylines: Dominance for Arbitrary Sets of Monotone Functions'** by Paolo Ciaccia and Davide Martinenghi introduces the concept of **F-Skylines** [3] which aims to combine the concepts of scoring functions and skyline operator in order to handle varying weight values in user preferences. The **F-**

dominance [3] concept is introduced where a tuple t F-dominates another tuple t′ if t is at least equal or better than t′ in all the scoring functions in F and is strictly better than t′ in at least one scoring function in F. The F-Skyline, **ND and PO** operators [3] are being introduced. It becomes increasingly difficult and important to retrieve the best set of results when we have a specific criterion, this reduces into a multi-objective optimization problem. There are three main approaches described in the paper that handles this problem -:

1. **Lexicographic approach:** This method considers some of the attributes to be more important, hence if records/tuples are better on the other attributes then this fact is ignored. This is a narrow approach of retrieving the results.
2. **Top-k scoring functions:** A scoring function is assigned and each record/tuple is assigned a score corresponding to which it is ranked from the best to the worst.
3. **Skyline:** The skyline represents the set of non-dominated tuples/records. The disadvantage with skylines is that it does not provide personalization of the result set and the output size is not controllable.

For example, if a person wants to choose a car based on price and mileage. A lexicographic approach would give a higher priority to the price, hence the lower price car irrespective of the mileage is returned. The mileage is only considered when there are ties in the calculation. If we take scoring functions, we do not know which of the two attributes the user prefers, it can be (0.5,0.5) or (0.6,0.4) or any other distribution. It is hard to predict the exact user preferences in scoring functions. The F-Skylines combines scoring functions and skyline operators to provide a constraint on the weight values (user preferences) used. The F-skylines also reduce the size of the result set obtained. Certain definitions in relation to the F-skyines are given below-:

1. **F-Dominance:** A tuple t is said to F-dominate another tuple t′ if t is equal or better than t′ in all the scoring functions in F and it is strictly better than t′ in at least one of the scoring functions in F [3].
2. **ND** – It represents the set of non-F-dominated [3] tuples in the dataset
3. **PO** – It represents the set of potentially optimal [3] tuples in a dataset. The tuple/record that is the best according to some scoring function in F is said to be potentially optimal with respect to F.
4. **Tuple distinguishing set** – A set of monotone functions F is said to be tuple distinguishing [3] if two tuples have different scores with respect to one of the scoring functions in F.

The paper – **'Marrying Top-k with Skyline Queries: Relaxing the Preference Input while Producing an Output of Considerable Size'** by Kyriakos Mourtadis, Keming Li and Bo Tang introduces two new operators **ORU and ORD** [5] which focusses on combining the concepts of the skyline operator and finding the best results according to the utility function. These new operators have been introduced taking into consideration three requirements: **personalization, controllable output size and flexibility in preference specification** [5] of the results obtained. The requirement of personalization of the results becomes important as the skyline retrieves the results but it does not personalize it and it is the same for every user. The output size of a skyline result cannot be controlled. The ORD and ORU operators aim to satisfy all the three requirements. The ORD focusses on relaxing the notion of dominance whereas the ORU focusses on the ranking by utility functions. The ORD and ORU are also scalable and responsive. Some definitions in relation to ORD and ORU are given below -:

1. **Seed** – The seed [5] represents the user preference input or the vector containing the weights of each attribute in the dataset
2. **Rho-dominance** – A record say a is said to Rho-dominate [5] another record b if and only if a is at least better than b according to every vector v and is strictly better than b in one vector input say vi.
3. **Rho -skyband** – The set of records/tuples that are Rho-dominated by fewer than k others is denoted by the Rho-skyband[5].
4. **ORD** – Given the seed w and the output size denoted by m, the ORD computes the records that are Rho-dominated by fewer than k others for the minimum Rho that produces exactly m records in the output [5].
5. **ORU** – Given the seed w and the output size denoted by m, the ORU computes the records that are the top-k result for at least one preference vector within a distance of Rho from w, for the minimum Rho that produces exactly m records in the output [5].

3. Introduction to Indicators

There are mainly three indicators being introduced in relation to the skyline points. These indicators are calculated only for the skyline points and not for the entire dataset as it is more efficient and less time consuming. These indicators characterize the skyline points. The three indicators introduced are the following -:

1. **Grid strength**
2. **Increase in domination Volume**
3. **Maxrank**

In the next part of this thesis, we will discuss all the three indicators and calculate the indicator values for datasets of various sizes and dimensions.

To give a brief description of all the three indicators, the Grid strength measures the robustness of the skyline point that is how robust it is to the grid of a given size. The increase in domination Volume measures how many points does a given skyline point dominate. This is calculated in terms of volume. Lastly, the Maxrank indicator calculates the maximum rank that a point can take in a dataset with variations in the query space

3.1 Grid Strength

The **Grid strength** calculates the robustness of a particular point with respect to the grid. The grid size can vary. An initial value for the Grid size is taken and this size is being increased at each iteration. Two points lying in the same grid cell would collapse to the target point (that is the left most corner of the cell). The point at which a given data point would collapse indicates the Grid strength of that point. A high value of the Grid strength indicates that the point is more robust to the Grid. The iteration stops when we are left with a single point in the grid mesh.

Consider the below skyline plot which is the skyline plot for the Youtube dataset: The attributes used for skyline calculation are views and duration.

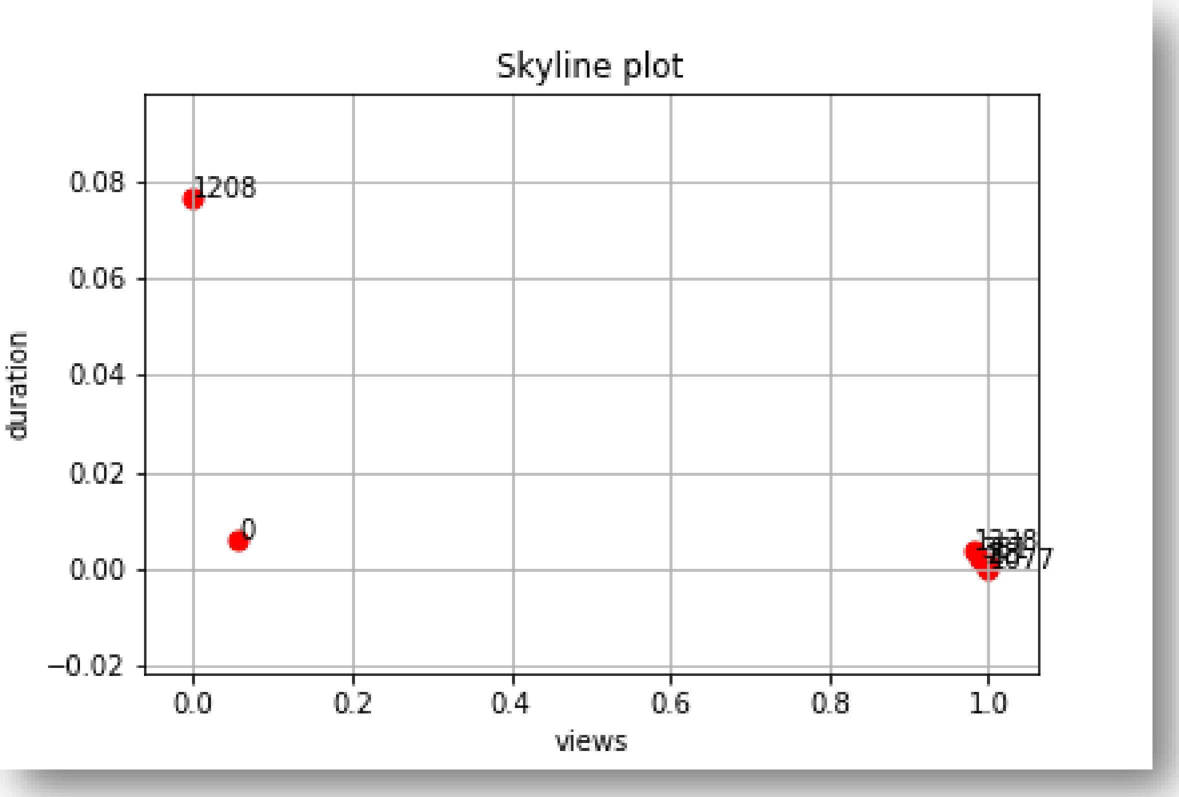


Figure 3.1: Skyline plot-Youtube Dataset

The skyline consists of 7 points. Most of the points are located at the bottom right of the graph with only two points 1208 and 0 being far apart from each other.

Now, we have calculated the values of the Grid strength starting from grid size of 1/100 = 0.001.

We will keep reducing the denominator, consequently the Grid cell size will start to increase. We will keep increasing until the cell size becomes 1/1 = 1.000. At each juncture the number of skyline points is seen and their co-ordinates changes at each iteration due to changes in the Grid cell size and collapsing of points due to domination. Given below is the step-by-step iteration of the Grid calculation until it collapses to a single point -:

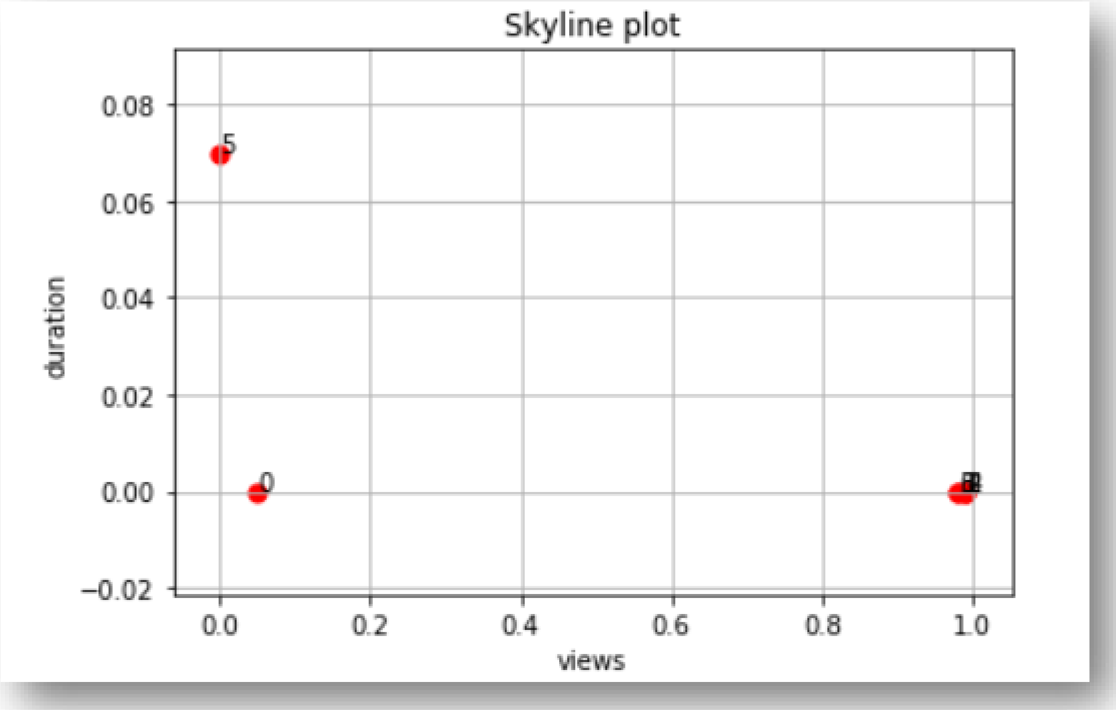


Figure 3.2: Grid plot-Youtube Dataset

1. The initial Grid size is 1/100 = 0.001. Here we can see that most of the skyline points that were in the bottom right part of the plot have collapsed since they were very close to each other. There are only two skyline points left in the Grid Mesh at the initial iteration of grid size equal to 1/99.

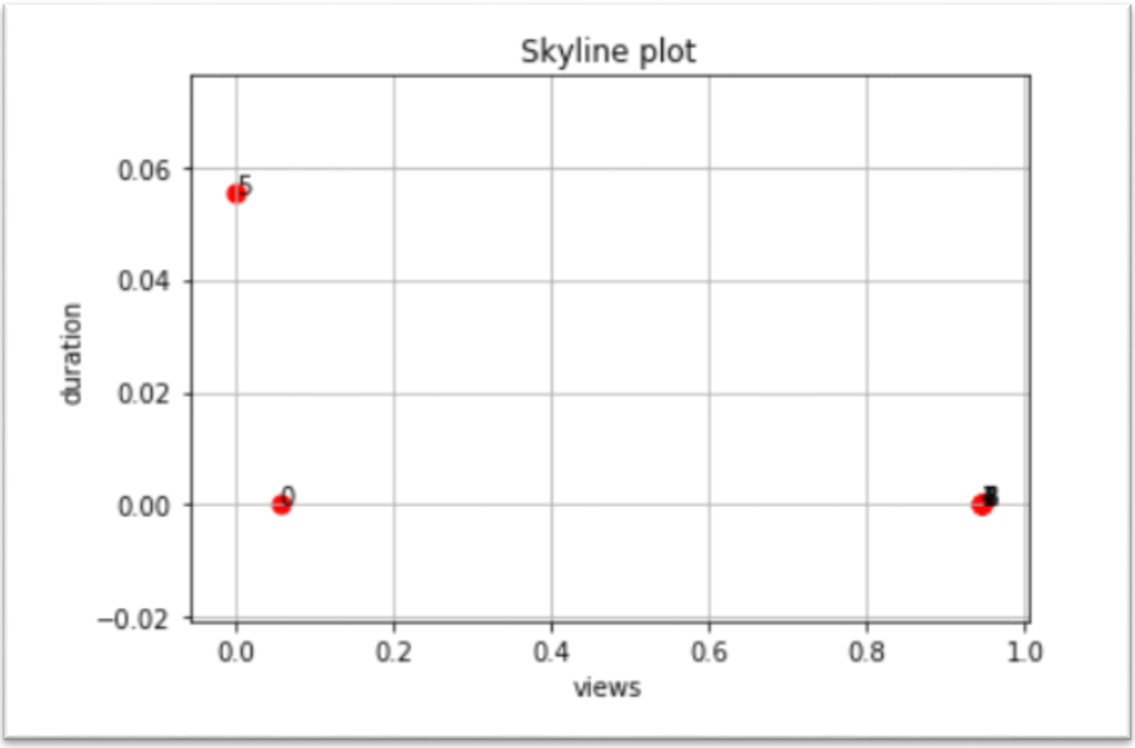
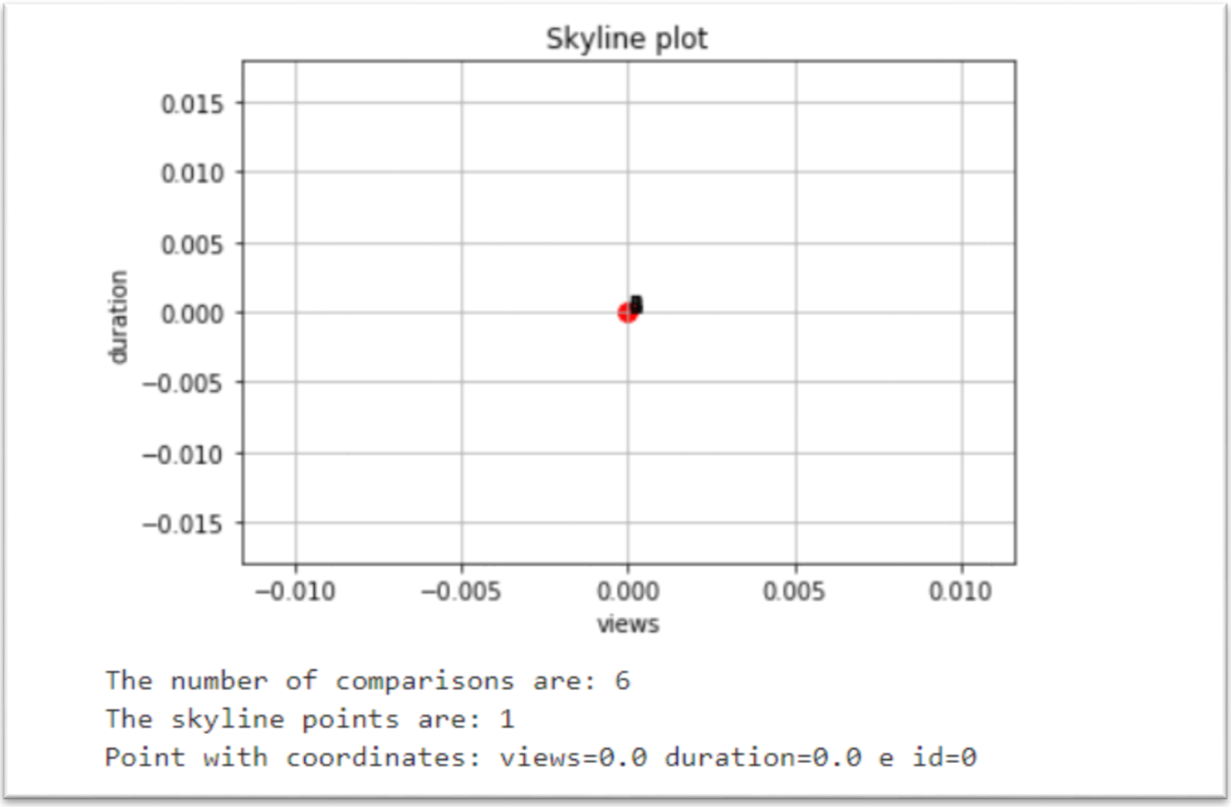


Figure 3.3: Grid plot-Youtube Dataset

2. After several iterations we reach the Grid size of 1/18 which is equal to 0.056. At this juncture we can see that none of the skyline points have collapsed and they are still intact in the Grid Mesh. There are only slight changes to the co-ordinates of the skyline points because of the increasing Grid size in the mesh.



```
The number of comparisons are: 6
The skyline points are: 1
Point with coordinates: views=0.0 duration=0.0 e id=0
```

Figure 3.4: Final Grid Youtube Dataset

3. At the last iteration we can see that finally the last point has collapsed from the Grid. The Grid cell size is equal to 1/1 which is equal to 1.000.

3.2 Increase in Domination Volume

The number of points that a given point dominates is given by the **Domination Volume.** Higher the Domination Volume indicates that it dominates many points in the skyline. The Volume Dominated is calculated for each Skyline point. The calculation of Domination Volume is made using the **Monte Carlo Method**. The conditional Domination Volume of a point is calculated by dividing the Volume Dominated by that particular point with the total Volume dominated by all the points in the skyline.

The NBA Dataset was taken to calculate the Domination volume for all the skyline points of that dataset.

The dataset consists of 18 skyline points. The skyline plot is given below -:

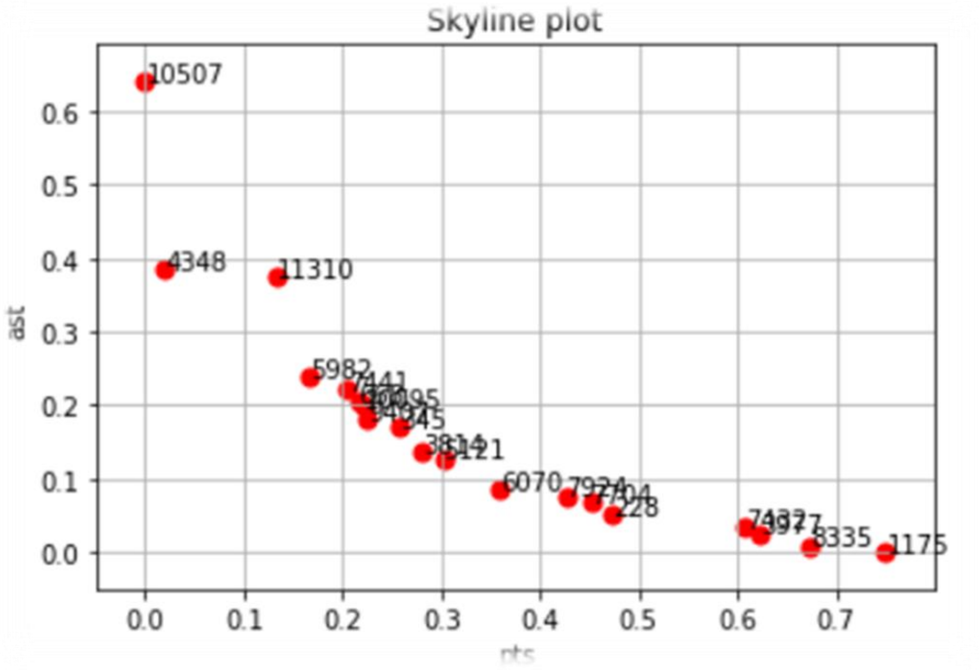


Figure 3.5: Skyline plot - NBA Dataset

Given below is the Total Volume dominated by all the skyline points. The total volume has been calculated using the Monte Carlo Method.

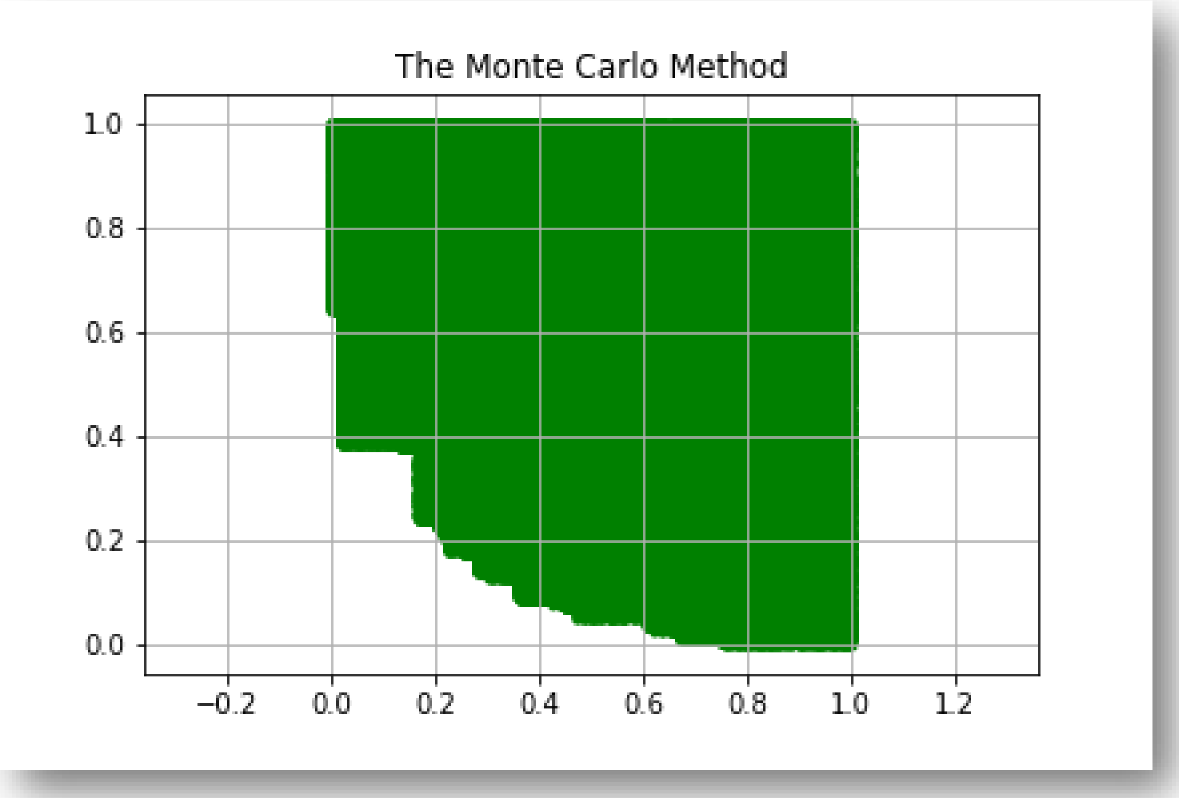


Figure 3.6: Total Volume of Domination- NBA Dataset

The total volume dominated by all the skyline points is equal to 0.879168.

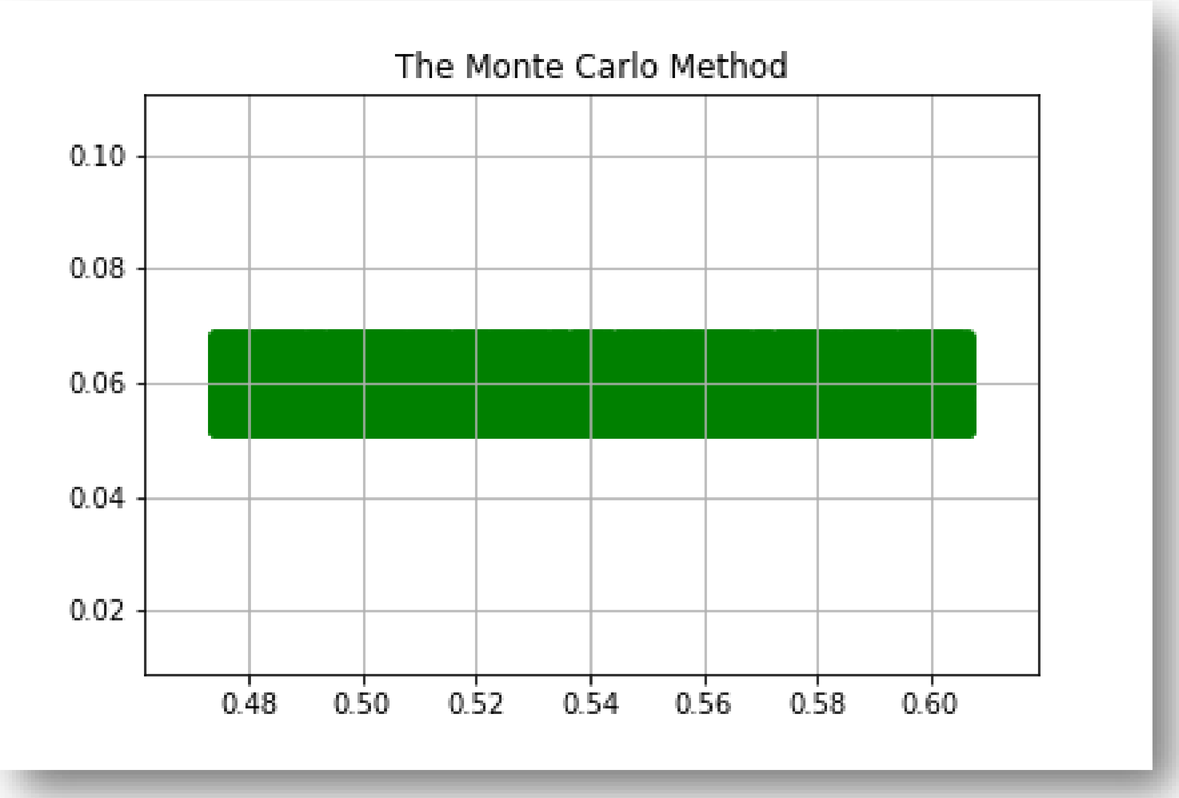


Figure 3.7: Domination Volume of particular point

The increase in Domination Volume for one of the skyline points is given above. The total number of points considered is equal to 50000. The number of points dominated by the particular skyline point is equal to 141843. The total volume dominated by the skyline point is equal to 0.00283686. The number of points dominated out of this initial value is calculated. The Domination Volume is finally calculated by the formula-:

Domination Volume(sp)=Number of points dominated/Total number of points

Here sp denotes skyline point.

3.3 Concavity Index - pIndex

The degree of the function required to convert a convex skyline point to a concave skyline point is given by the pIndex or the Concavity index. The higher the concavity index of the point, the more difficult it is to convert it to a concave point. The concavity index is based upon the notion of Convex hull explained earlier.

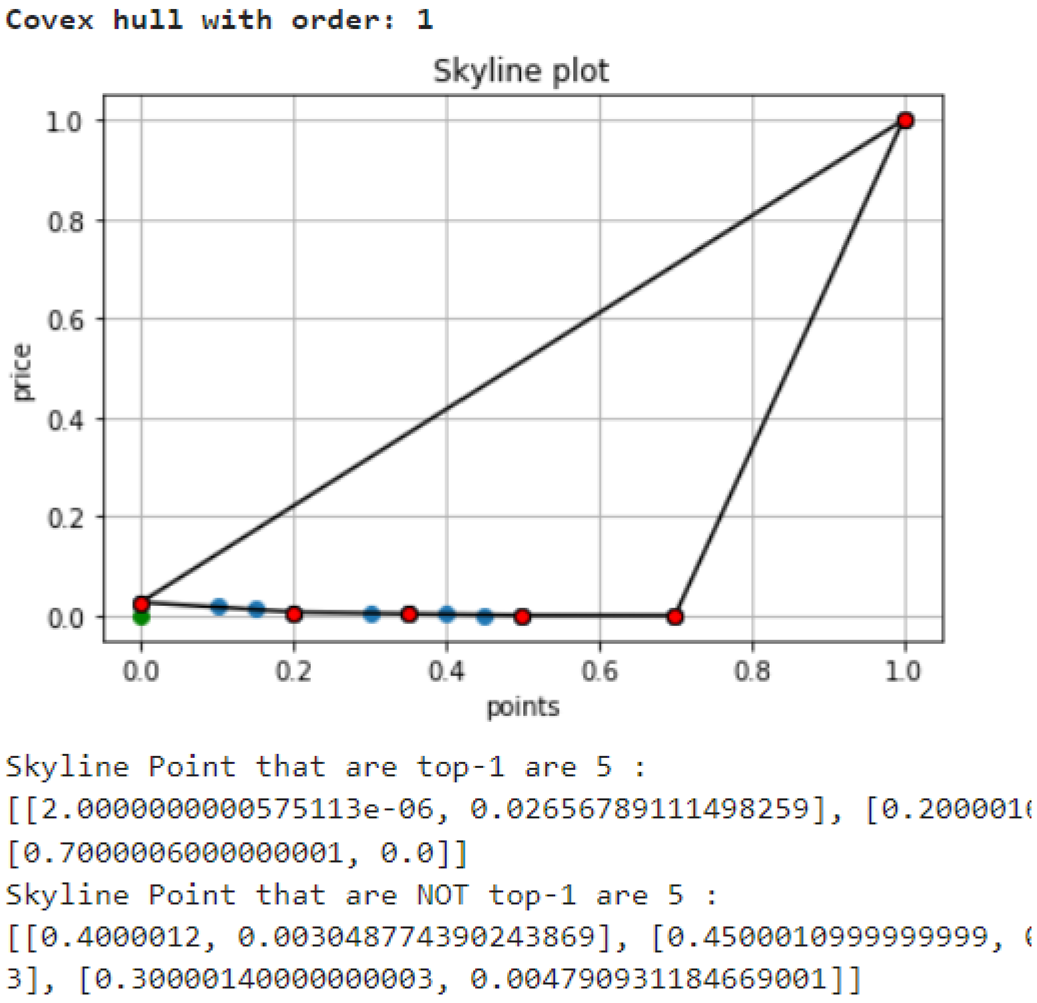
Suppose we have A to be the set of convex points then if we transform the co-ordinates to say the second degree order(to the square) then the new set of convex points would be the set A along with those points which are convex for the second order function. This transformation of the co-ordinates is only valid if we have the target point equal to 0.

Two operations are carried out on the data -:

1. Normalization of the data to value between 0 and 1

2. Converting the target point of the data to 0 so that 0 is our target point.

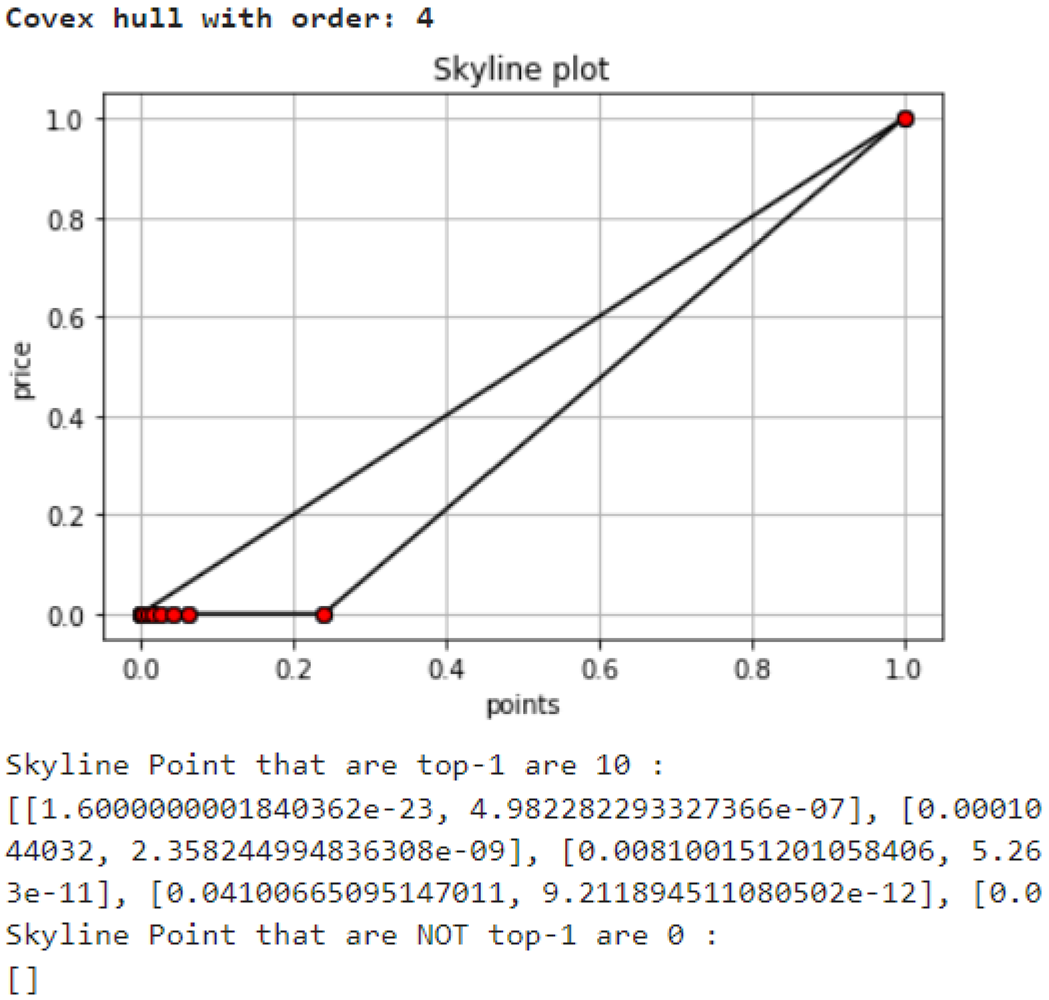
The concavity index value varies with each point and in some cases can be very high which would lead a top-1 point to be very concave.



```
Covex hull with order: 1
```

```
Skyline Point that are top-1 are 5 :
[[2.0000000000575113e-06, 0.02656789111498259], [0.200001
[0.7000006000000001, 0.0]]
Skyline Point that are NOT top-1 are 5 :
[[0.4000012, 0.003048774390243869], [0.45000109999999, 
3], [0.30000140000000003, 0.004790931184669001]]
```

Figure 3.8: Convex hull order 1



```
Covex hull with order: 4
```

```
Skyline Point that are top-1 are 10 :
[[1.6000000001840362e-23, 4.982282293327366e-07], [0.00010
44032, 2.358244994836308e-09], [0.008100151201058406, 5.26
3e-11], [0.04100665095147011, 9.211894511080502e-12], [0.0
Skyline Point that are NOT top-1 are 0 :
[]
```

Figure 3.9: Convex hull order 4

Given above are two figures depicting the pIndex calculation of the skyline points in the Wines dataset. The wines dataset has 10 skyline points in total. For the convex hull with order 1, 5 points out of 10 are top-1 whereas the other 5 points are not top-1. When we move further to the fourth order convex hull then all the points in the skyline become top-1. Now, there are no skyline points that are not top-1 or concave and all the skyline points have been converted to a convex point lying in the convex hull. Hence the points in the first iteration which were top-1 will have a pIndex equal to 1 and the points that have turned convex at the fourth iteration that is for the fourth order will have a pIndex value equal to 4. There might be other datasets wherein the points will have a higher pIndex than 4 but for the Wines dataset the maximum pIndex obtained is equal to 4.

3.4 Maxrank

The top-k query results in the tuples with the highest score with respect to a particular scoring function. In this relation, the **Maxrank** computes the maximum possible rank that can be attained by the particular data point with variations in the query space. The calculation of the Maxrank indicator is equivalent to removing points from the database so that a particular data point becomes the top result with respect to some query criterion. The Maxrank indicator is independent of the dimensionality.

There are three algorithms that calculate the Maxrank of a given tuple/record -:

1. **FCA Algorithm (First cut algorithm)**
2. **BA Approach (Basic Algorithm)**
3. **AA Approach (Advanced Algorithm)**

FCA Algorithm – The FCA Algorithm can be used in the case where the dimension is equal to 2. Here the sum of the query vector weights is assumed to be equal to 1.

The following formula is used in FCA -:

$$\sum qi = 1$$

$$S(r) = r*q1 + r*(1-q1)$$

Here S(r) denotes the score of the record or tuple. Since the sum of qi = 1, we can use the product of r*q1 and r*(1-q1). The plot of S(r) versus q1 is a line. We draw lines for each of the tuples and compute all the intersections with the tuple r whose Maxrank we would want to calculate. We then sort this based on q1. The minimum order of the record/tuple r is calculated.

BA Approach (Basic Algorithm) – Before delving into the functioning of the BA Algorithm, it is essential to introduce some concepts -:

Dominators: Consider a record/tuple p. All the tuples that dominate this record have a higher score than p. Let r be any record/tuple that dominates p. Then we have that S(r) > S(p) and the order of r is lower than the order of p.

Dominees: Consider a record/tuple p. All the tuples that the record p dominates have a lower score than p. Let r be any record/tuple that is being dominated by p. Then we have that S(r) < S(p) and the order of r is higher than the order of p.

Incomparable records: All the records/tuples that do not dominate or are not dominated by a record p is called an incomparable record.

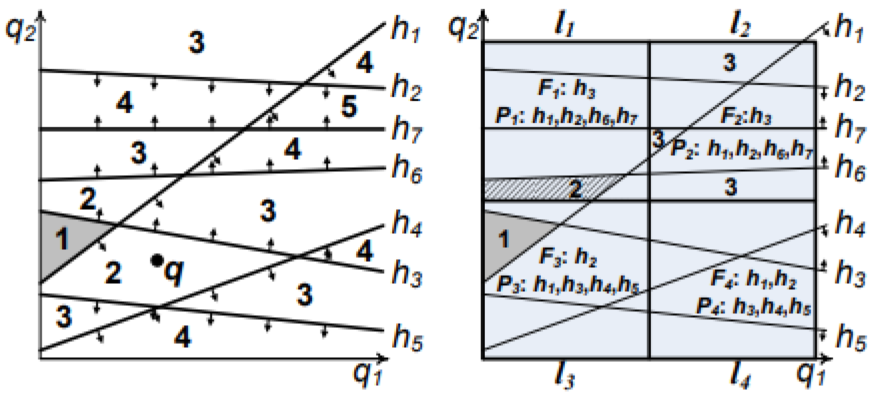
For an incomparable record whose order is lower than the record/tuple p, we have S(r) > S(p) where r is the incomparable record. This corresponds to the formula -:

$$\sum ri *qi > \sum pi * qi = \sum (ri-pi) * qi > 0$$

The above interpretation refers to a half space that passes through the origin. Through this formula, for every incomparable record we compute its corresponding half space. The arrangement of half spaces divides the space into cells. For a given cell c , the number of half spaces that it contains is given by |Hc| where Hc is number of half spaces contained in the cell. If we denote |D+| by the number of dominators, then the formula for the Maxrank is given by -:

$$k^* = |D+| + |Hc| + 1 = |D+| + 2$$

Here, k* is the minimum order, |Hc| is the number of half spaces in cell c and |D+| is the number of dominators.



(a) Half-space arrangement (b) BA example

Figure 3: *MaxRank* in (reduced) query space, $d = 3$

Figure 3.10: Maxrank in reduced query space

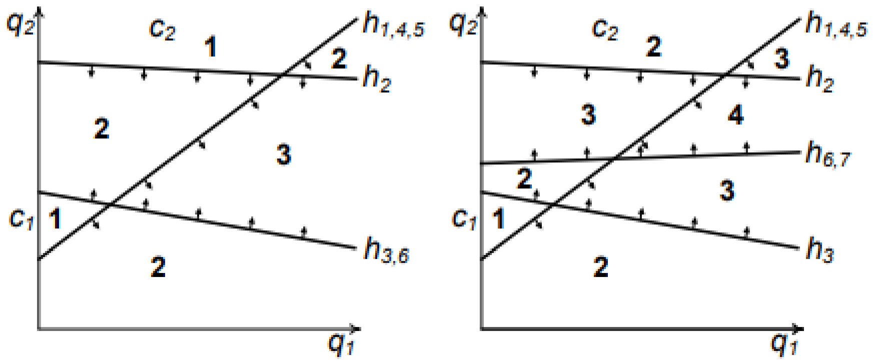
The above figure shows the half space arrangement with the cells and an example for the BA algorithm for d = 3 and with the reduced dimensional space of 2.

AA Approach (Advanced Algorithm): The AA algorithm resolves the problem faced by BA which becomes inefficient as the number of incomparable records increase which is the case when there are datasets that are considerably large. The

AA approach includes the records/tuples only when they can affect the Maxrank processing.

AA applies the dominance relation between points to avoid processing all the points in the dataset. This proves to be efficient and less time consuming than BA.

Suppose a record/tuple A dominates another record B , then the order of A is surely going to be lesser than the order of tuple B. If P is the record whose Maxrank we would want to calculate, then B can score higher than P if and only if A scores higher than P. There is a subsumption of the half space of B induced by A. **The main objective of AA is to find the minimum order of the cells in the half space arrangement without considering the subsumed half spaces.**



(a) First iteration (b) Second iteration

Figure 5: AA example in (reduced) query space, $d = 3$

Figure 3.11: AA example in reduced query space

The above figure shows an example for AA algorithm with the reduced query space where d=3

Singular: The half spaces of records that subsumes no other record is called a singular record.

Augmented: The half spaces of records that subsumes at least one record is called an augmented half space.

Mixed arrangement: An arrangement consisting of both Singular and Augmented half spaces.

If a cell is not under any augmented half space, then its extent and order are accurate. The AA algorithm continuously divides the half spaces to find in the minimum order of the cells in the Augmented Half space arrangement.

4. Experimentation

Python was used in the following experiments. Python is the most widely used programming language for data related tasks with extensive libraries available to the user to handle data. Jupyter Notebook was used as the platform for programming. With Jupyter Notebook , it is easy to write code on the web browser and also it is possible to share the notebook through the cloud.

CLion was used to carry out the calculation of the Maxrank indicator. CLion is an IDE used for programming mainly in the C and C++ languages. It is a powerful IDE from the JetBrains Company.

The main tasks on the datasets carried out were -:

1. Data preprocessing
2. Calculation of the skyline points
3. Calculation of the Grid Strength indicator
4. Calculation of the Increase in Domination Volume
5. Preparing the data for Maxrank program
6. Calculation of Maxrank for the datasets
7. Creating a summary of the indicators in Jupyter Notebook

The main type of datasets used for the below experiments are 2-D Datasets.

The 2-D Datasets used are given below -:

1. Wines Dataset
2. NBA Dataset
3. NBA Raptors Dataset
4. Youtube Videos Dataset
5. Undergrad Universities Dataset
6. Skyrim Weapons Dataset
7. USA Cars Dataset
8. Games of All Time Dataset
9. Perth Housing Dataset
10. Best Bowling Stats Dataset

    All the indicators have been calculated for the above datasets.

4.1 2-D Datasets

4.1.1 Wines Dataset

General Description:

Number of tuples/records: 150930

Number of Skyline points: 10

Attributes used: ‘points and ‘price’

Range for points data: 20

Range for price data: 2296.0

Skyline:

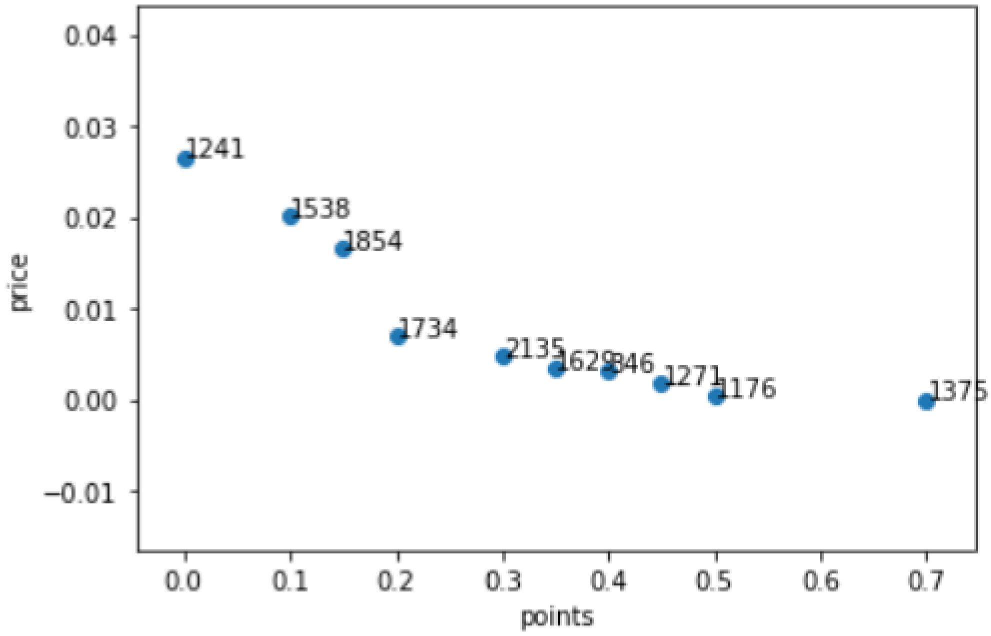
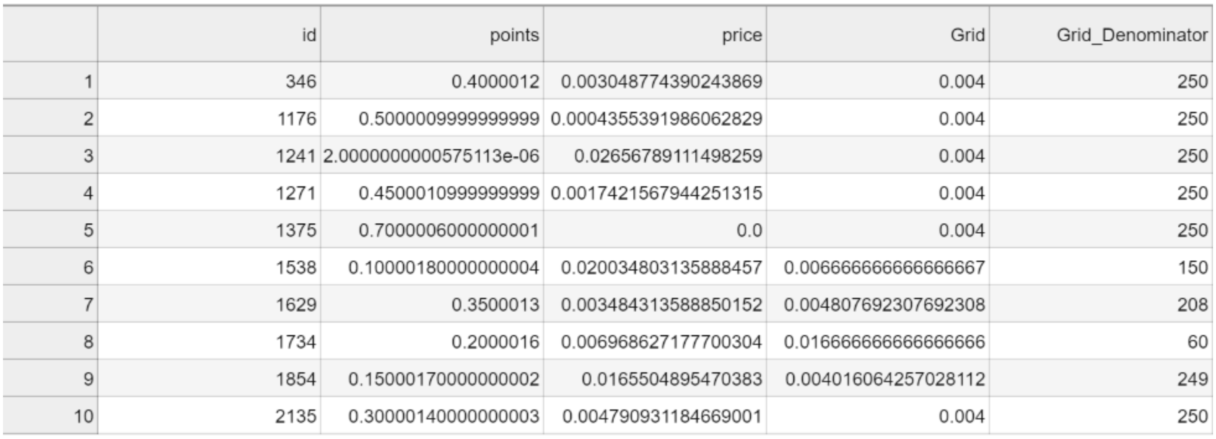


Figure 4.1: Skyline plot-Wines Dataset

The above plot represents the skyline points for the Wines Dataset. We have plotted points vs price. There are 10 skyline points in this plot.

Grid Strength:

The table below specifies the Grid strength for all the skyline points in the dataset. From the table it can be seen that the points with id 1241 and 1734 have the highest Grid strength value and seem to be robust with respect to the Grid mesh.



	id	points	price	Grid	Grid_Denominator
1	346	0.4000012	0.003048774390243869	0.004	250
2	1176	0.5000009999999999	0.0004355391986062829	0.004	250
3	1241	2.0000000000575113e-06	0.02656789111498259	0.004	250
4	1271	0.4500010999999999	0.0017421567944251315	0.004	250
5	1375	0.7000006000000001	0.0	0.004	250
6	1538	0.10000180000000004	0.020034803135888457	0.006666666666666667	150
7	1629	0.3500013	0.003484313588850152	0.004807692307692308	208
8	1734	0.2000016	0.006968627177700304	0.016666666666666666	60
9	1854	0.15000170000000002	0.0165504895470383	0.004016064257028112	249
10	2135	0.30000140000000003	0.004790931184669001	0.004	250

Figure 4.2: Grid Strength -Wines Dataset

IncreaseinDominationVolume:

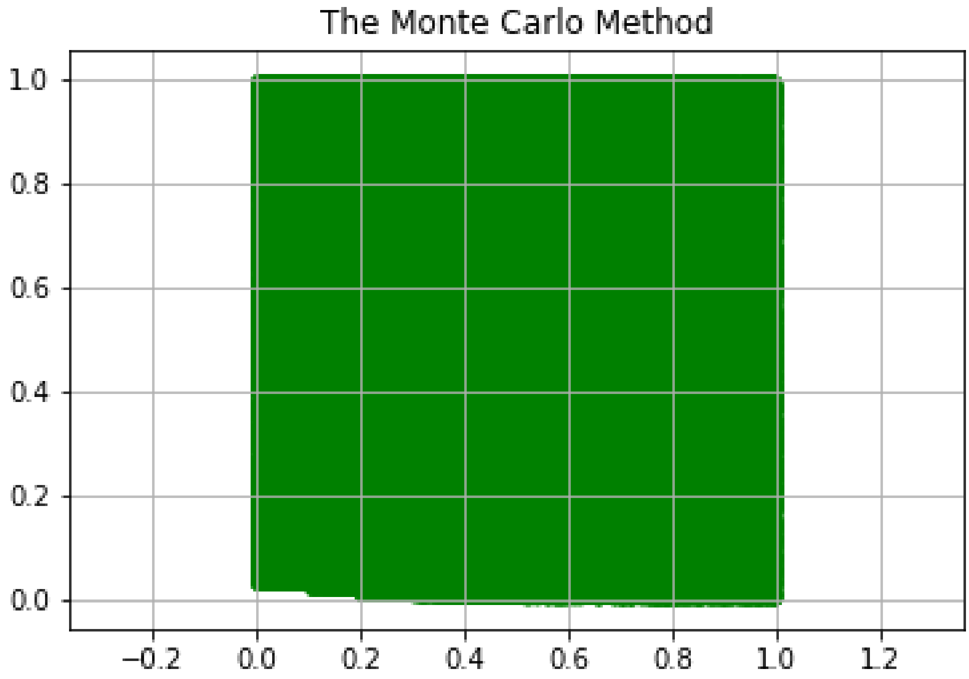
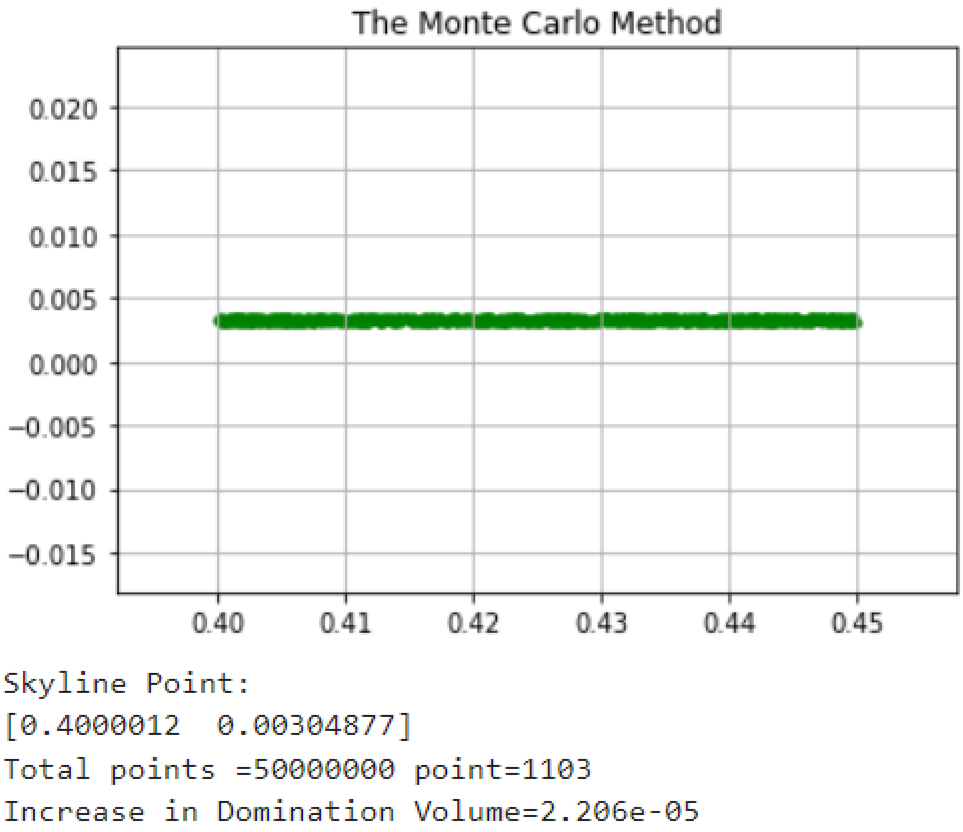


Figure 4.3: Total Volume of Domination - Wines Dataset

The above image represents the Total Volume dominated by all the skyline points. The total volume of domination is equal to 0.99405. This has been calculated using the Monte Carlo Method as mentioned previously.



```
Skyline Point:
[0.4000012  0.00304877]
Total points =50000000 point=1103
Increase in Domination Volume=2.206e-05
```

Figure 4.4: Domination Volume of point-Wines Dataset

The above diagram represent some of the Domination Volume values for one of the skyline points. The total number of points used for this is equal to 5 * (10^7). The number of points that the skyline point dominates out of the total number of points is equal to 1075.

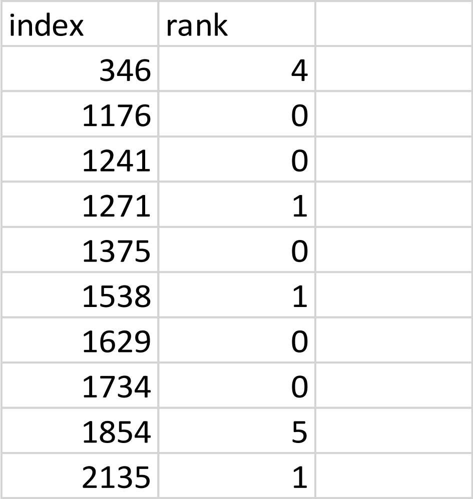
pIndex:

index	pIndex
346	4
1176	1
1241	1

1271	2
1375	1
1538	2
1629	1
1734	1
1854	3
2135	2

Table 4.1: PIndex for Wines Dataset

The above table shows the pIndex values for the wines dataset. We can see that the points 346 and 1854 have the highest pIndex values with 4 and 3 respectively. However most of the points in the skyline seem to be convex already.



index	rank	
346	4	
1176	0	
1241	0	
1271	1	
1375	0	
1538	1	
1629	0	
1734	0	
1854	5	
2135	1	

Table 4.2.: Maxrank for Wines Dataset

Maxrank calculation:

The above table represents the Maxrank corresponding to the different skyline points. The points that have the maxrank equal to 0 may be found from a linear function. We are mainly interested in those points whose maxrank is higher than 0. We have to compare the Maxrank values with the values of the Grid Strength and Increase in Domination Volume for these particular skyline points.

Summary:

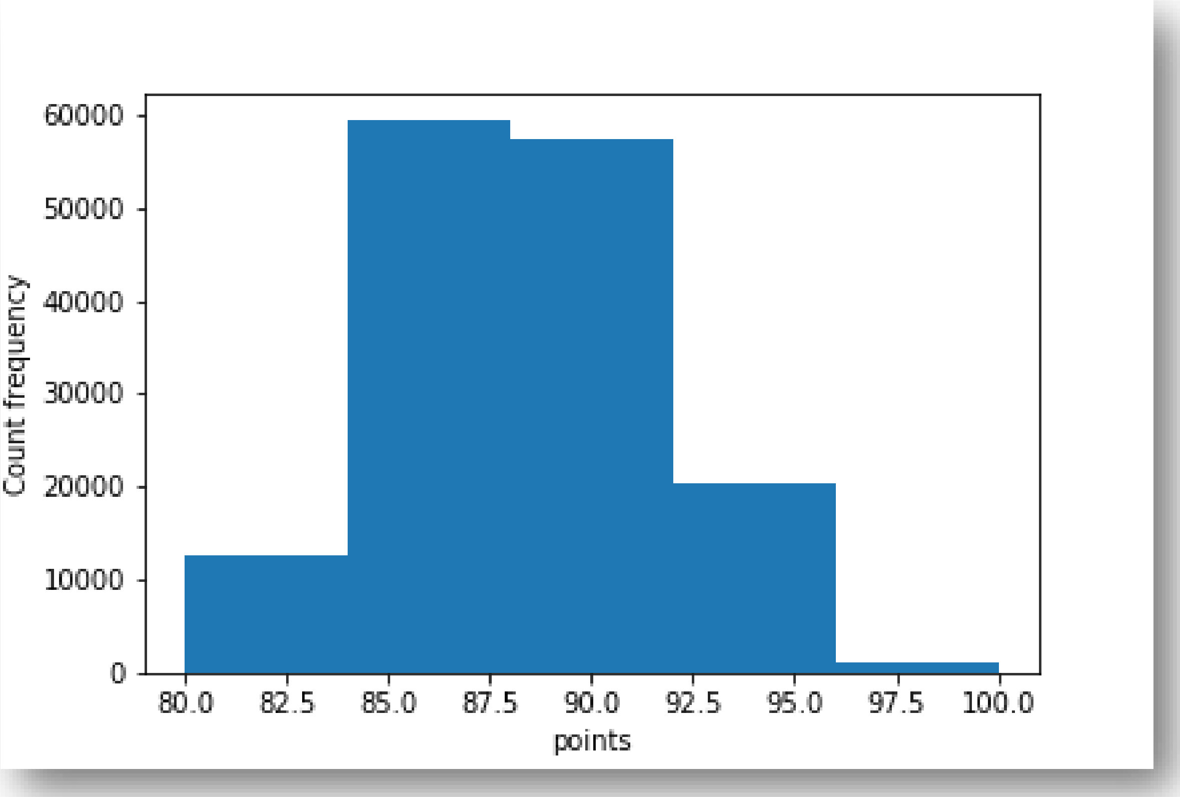


Figure 4.5: Histogram points data

The histogram represents the distribution of the points data in the dataset. We can see that the points data varies from the values between 80.0 and 100.0. A majority of the data values lie between the value of 85.0 and 87.5 as can be seen from the distribution as it corresponds to the highest frequency.

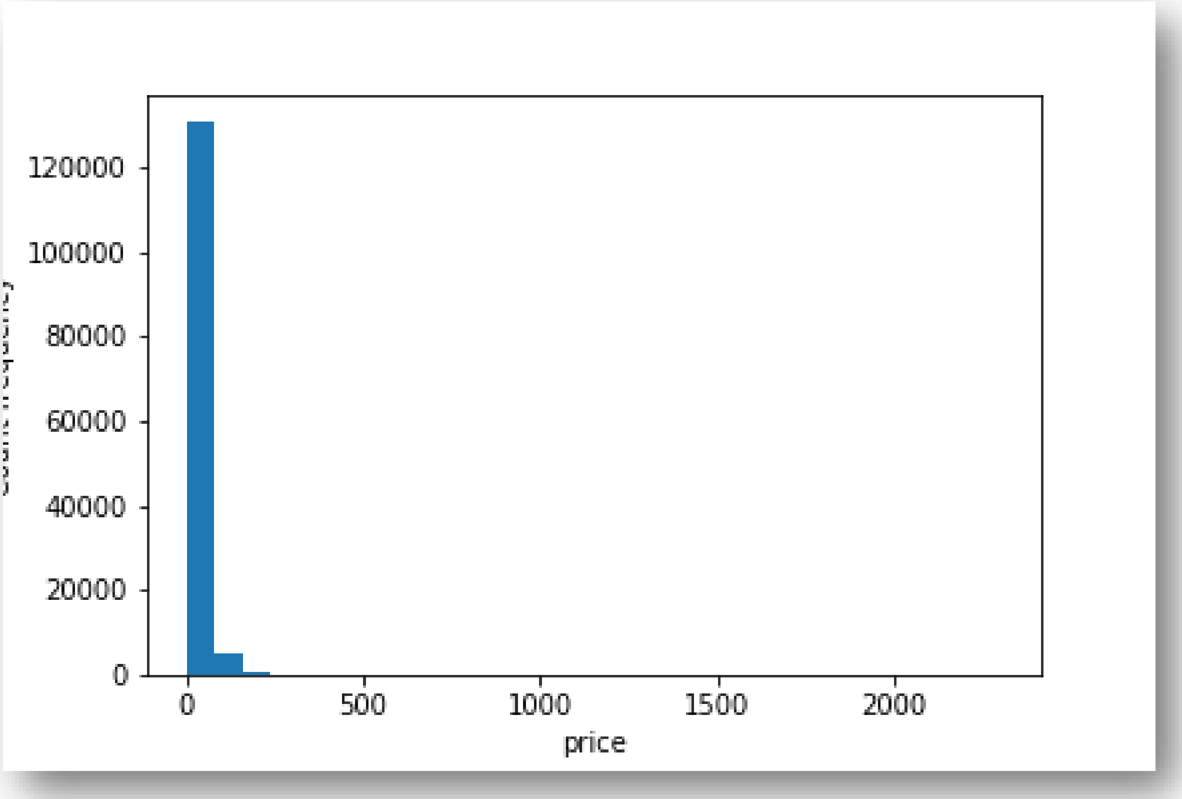


Figure 4.6: Histogram price data

This histogram represents the distribution of the values for the price data in the dataset. From the histogram we can see that the values of price fluctuate between 0 and 500. The x axis scale is between 0 and 2000. The y axis scale is between 0 and 120000.

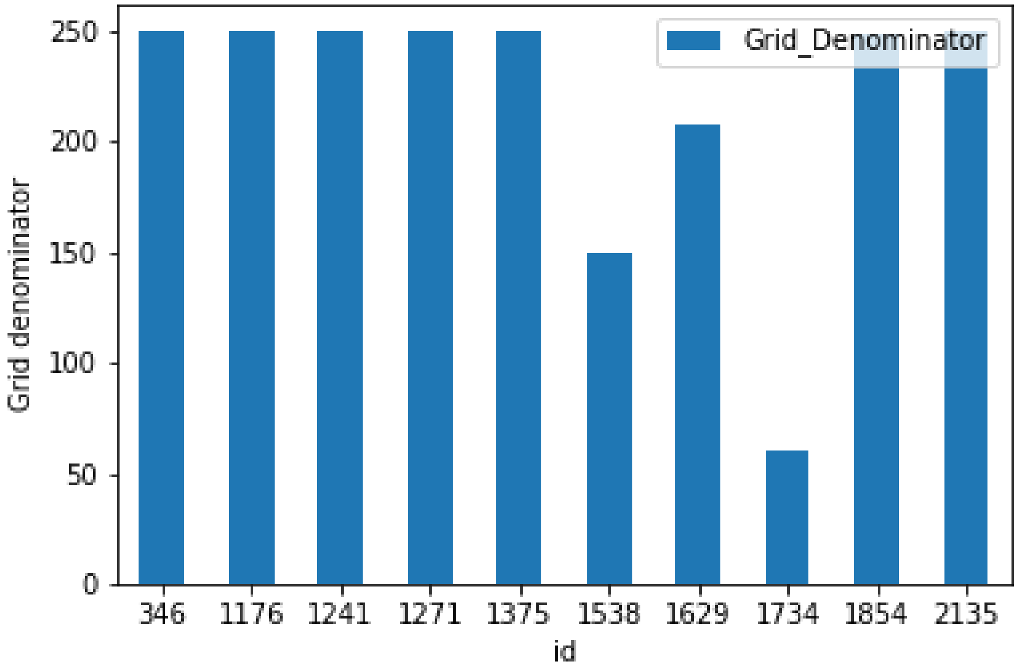


Figure 4.7: Grid Denominator bar plot - Wines Dataset

The bar plot represents the Grid denominator for the skyline points. The lower the Grid Denominator, the better the are the points. Lower Grid Denominator means that the skyline point is more robust to the Grid size variation. Here we can see that the point with id 1241 is has the lowest value for the Grid Denominator which means that it is the most robust with respect to the other skyline points in the Grid Mesh. The other skyline point that seems to be robust with the Grid mesh is the point with id 1734. It would be interesting to check if this point has a relatively high Domination Volume and a Maxrank value which would help us reach a conclusion that this point might be interesting to consider.

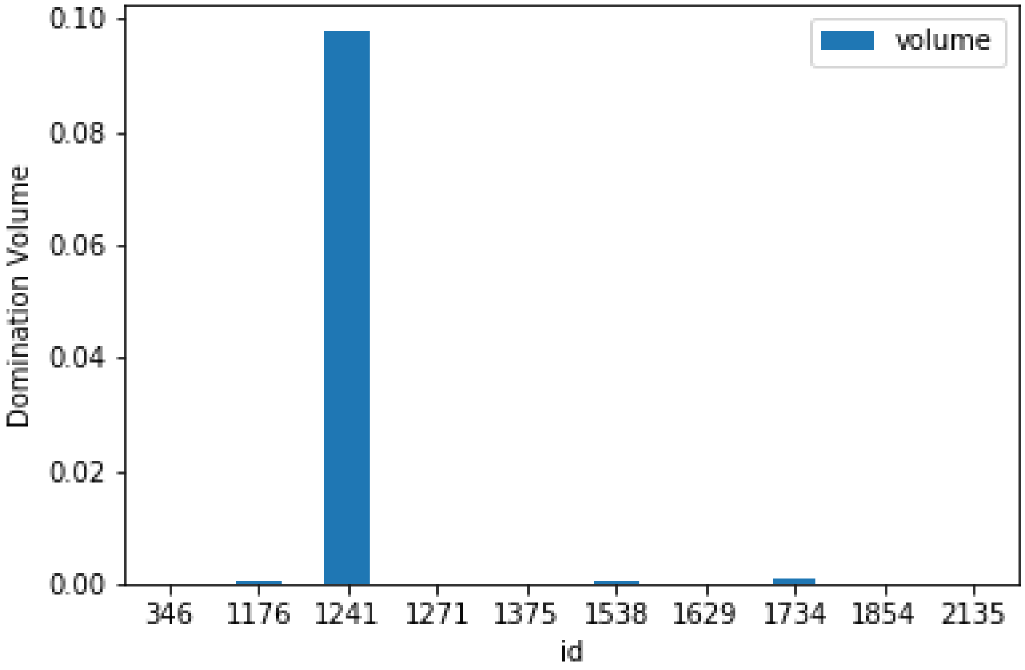


Figure 4.8: Domination Volume bar plot-Wines Dataset

The bar plot represents the Increase in Domination Volume for each of the skyline points. Here, there is no surprise that the point with id 1241 has the highest Domination Volume, it did have a low Grid Denominator value, so this result gives proof to what we have found in the previous bar plot. The other points that seem to have a relatively small but non-negligible value for the Volume are the points with the ids 1734,1854,1538 and 1176. The rest of the skyline points seems to have a pretty much negligible or we can say almost zero increase in Domination Volume.

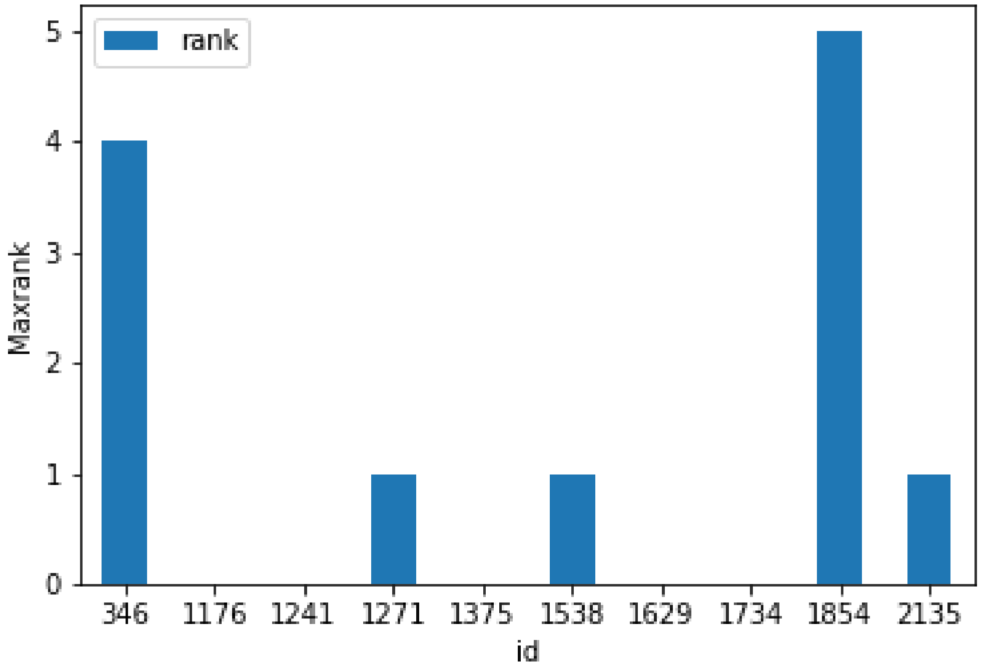


Figure 4.9: Maxrank bar plot - Wines Dataset

The bar plot represents the Maxrank values for the skyline points. Here the points that have a zero rank are 1176,1241,1375,1629 and 1734. The points that seem to have a non-zero value for the Maxrank are 346,1271,1538,1854 and 2135. From this plot we can conclude that probably 1241 is the only point which was dominant in the dataset. The expected maxrank of 1734 is equal to 0. The points 346,1271,1538,1854 and 2135 have a non-zero Maxrank value but all of these points did not have a significant Domination Volume value or Grid Denominator value. Therefore, for this particular dataset the point with the id 1241 was the only dominant point in the dataset.

Data Insights: From the above plots we can possibly conclude that the main points of interest in the wines dataset are 1538 and 1734 since they have a low grid denominator value corresponding to a high grid strength. Out of the two points we can see that the maxrank of 1734 is 0 and the maxrank of 1538 is equal to 1. Hence 1734 can be found on the convex hull while 1538 is probably not. This leads to the conclusion that the point with id 1538 is probably the point we are looking for.

4.1.2 NBA Dataset

Description of Dataset:

Number of records/tuples:

Number of skyline points:

The attributes chosen for analysis:

The range for 'pts' data:

The range for 'ast' data:

Skyline:

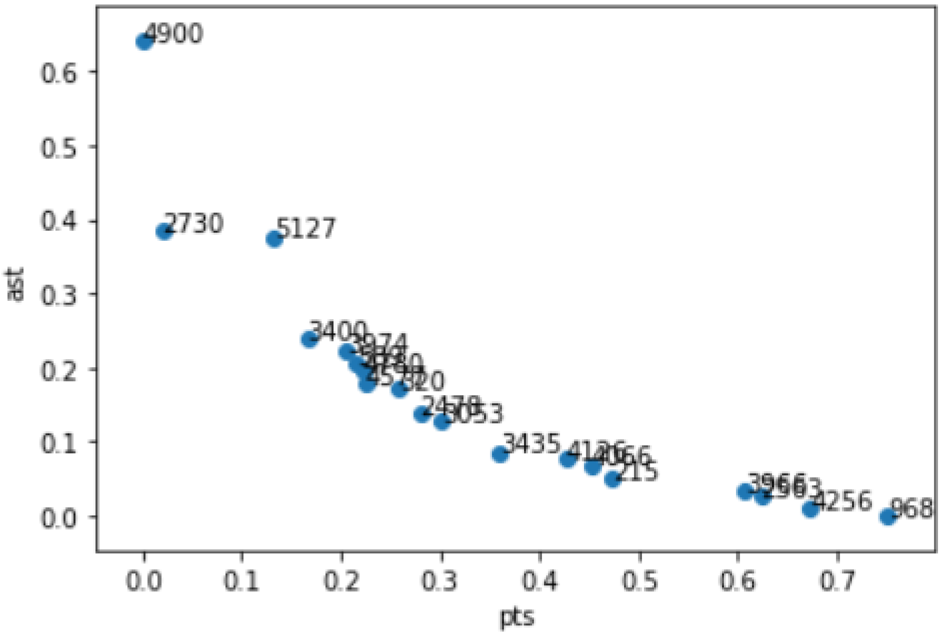
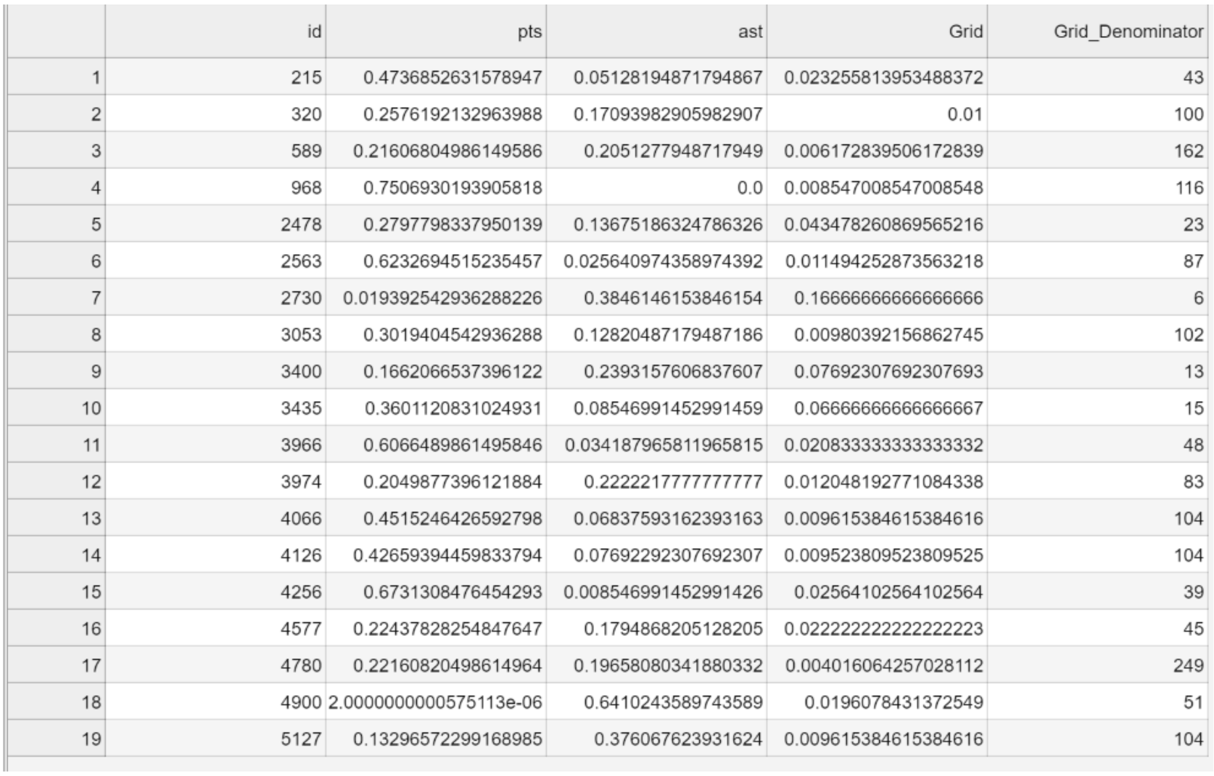


Figure 4.10: Skyline plot - NBA Dataset

The plot shows the skyline points. The point with id **4900** is at the highest point on the y-axis and the point with id **968** is at the lowest point corresponding to the y-axis.

Grid Strength:

The figure represents the Grid Denominator for the skyline points along with their Grid strength. It can be seen that the point with id 2730 has the lowest Grid denominator with a value equal to 6. The reason for such a low value for the Grid Denominator value of id 2730 is that it is at a larger distance with respect to the point 4900, so it probably collapses much later in the Grid Mesh. The point with id 4780 collapses the fastest in the grid as can be seen in its Grid Denominator that is equal to 249.

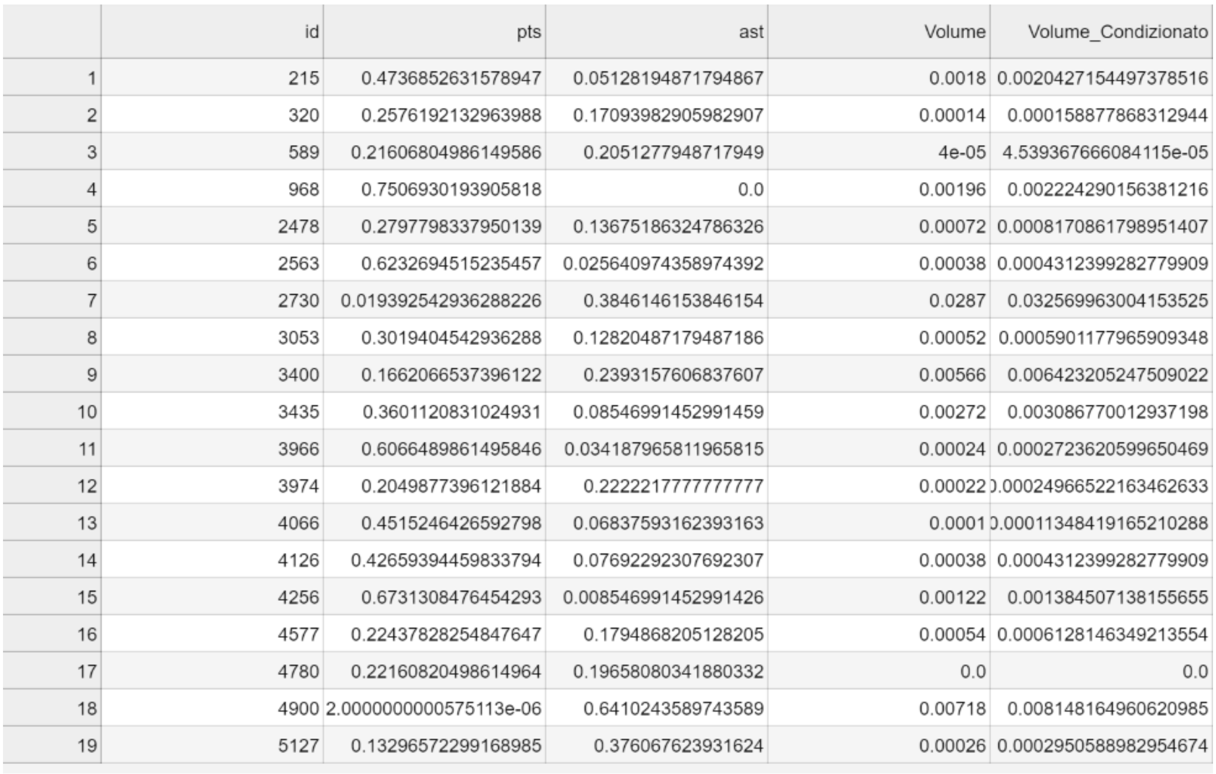


	id	pts	ast	Grid	Grid_Denominator
1	215	0.4736852631578947	0.05128194871794867	0.023255813953488372	43
2	320	0.2576192132963988	0.17093982905982907	0.01	100
3	589	0.21606804986149586	0.2051277948717949	0.006172839506172839	162
4	968	0.7506930193905818	0.0	0.008547008547008548	116
5	2478	0.2797798337950139	0.13675186324786326	0.043478260869565216	23
6	2563	0.6232694515235457	0.025640974358974392	0.011494252873563218	87
7	2730	0.019392542936288226	0.3846146153846154	0.16666666666666666	6
8	3053	0.3019404542936288	0.12820487179487186	0.00980392156862745	102
9	3400	0.1662066537396122	0.2393157606837607	0.07692307692307693	13
10	3435	0.3601120831024931	0.08546991452991459	0.06666666666666667	15
11	3966	0.6066489861495846	0.034187965811965815	0.020833333333333332	48
12	3974	0.2049877396121884	0.2222217777777777	0.012048192771084338	83
13	4066	0.4515246426592798	0.06837593162393163	0.009615384615384616	104
14	4126	0.42659394459833794	0.07692292307692307	0.009523809523809525	104
15	4256	0.6731308476454293	0.008546991452991426	0.02564102564102564	39
16	4577	0.22437828254847647	0.1794868205128205	0.022222222222222223	45
17	4780	0.22160820498614964	0.19658080341880332	0.004016064257028112	249
18	4900	2.0000000000575113e-06	0.6410243589743589	0.0196078431372549	51
19	5127	0.13296572299168985	0.376067623931624	0.009615384615384616	104

Figure 4.11: Grid Strength- NBA Dataset

Domination Volume:

The figure represents the Domination Volume for all the skyline points. Again we can see that the point with id 2730 has an domination volume equal to 0.0287 which is the highest among all the other skyline points in the dataset. This result is in sync with our result for the Grid denominator value wherein id 2730 had a relatively low Grid Denominator value compared to the other skyline points.



	id	pts	ast	Volume	Volume_Condizionato
1	215	0.4736852631578947	0.05128194871794867	0.0018	0.0020427154497378516
2	320	0.2576192132963988	0.17093982905982907	0.00014	0.000158877868312944
3	589	0.21606804986149586	0.2051277948717949	4e-05	4.539367666084115e-05
4	968	0.7506930193905818	0.0	0.00196	0.002224290156381216
5	2478	0.2797798337950139	0.13675186324786326	0.00072	0.0008170861798951407
6	2563	0.6232694515235457	0.025640974358974392	0.00038	0.0004312399282779909
7	2730	0.019392542936288226	0.3846146153846154	0.0287	0.032569963004153525
8	3053	0.3019404542936288	0.12820487179487186	0.00052	0.0005901177965909348
9	3400	0.1662066537396122	0.2393157606837607	0.00566	0.006423205247509022
10	3435	0.3601120831024931	0.08546991452991459	0.00272	0.003086770012937198
11	3966	0.6066489861495846	0.034187965811965815	0.00024	0.0002723620599650469
12	3974	0.2049877396121884	0.2222217777777777	0.00022	0.00024966522163462633
13	4066	0.4515246426592798	0.06837593162393163	0.0001	0.00011348419165210288
14	4126	0.42659394459833794	0.07692292307692307	0.00038	0.0004312399282779909
15	4256	0.6731308476454293	0.008546991452991426	0.00122	0.001384507138155655
16	4577	0.22437828254847647	0.1794868205128205	0.00054	0.0006128146349213554
17	4780	0.22160820498614964	0.19658080341880332	0.0	0.0
18	4900	2.0000000000575113e-06	0.6410243589743589	0.00718	0.008148164960620985
19	5127	0.13296572299168985	0.376067623931624	0.00026	0.0002950588982954674

Figure 4.12: Domination Volume - NBA Dataset

The total volume of domination is equal to 0.88118.

pIndex :

id	pIndex
215	1
320	10
589	7
968	1
2478	1
2563	2
2730	1
3053	3

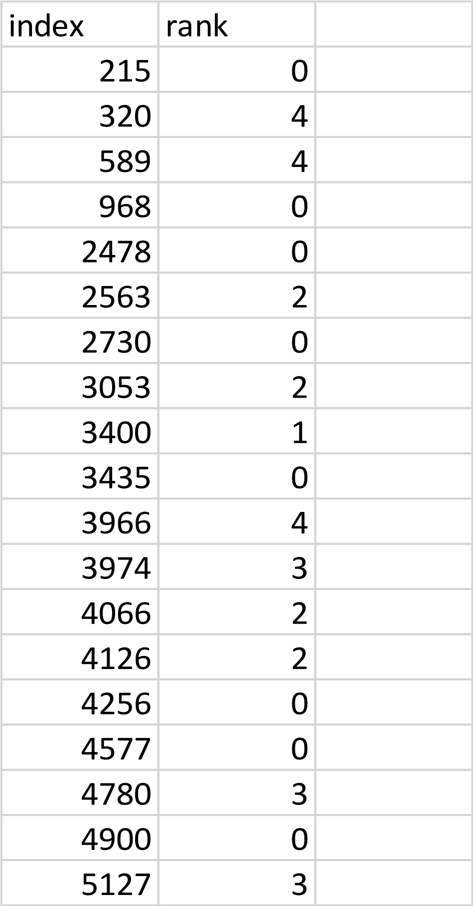
3400	2
3435	1
3966	4
3974	8
4066	5
4126	6
4256	1
4577	1
4780	-1
4900	1
5127	8

Table 4.3: PIndex for NBA Dataset

The above table gives the pIndex values for the NBA Dataset. Here we can see that the points with the highest pIndex values are 320,3974,4126 and 5127 which are concave skyline points.

Maxrank:

The table shows the Maxrank values for all the skyline points in the dataset. Here it is worth noting that the point that had a highest increase in domination volume has a maxrank value equal to 0. The other points that have a non-zero maxrank value are – 589,2563,3053,3400,3966,3974,4066,4126,4780 and 5127. We need to look at those skyline points who have a non zero value for the maxrank as these points have a potential of being interesting providing that they have a considerable high value for increase in domination volume and a relatively low value for the Grid Denominator/relatively high value for the Grid strength with respect to the Grid Mesh.



index	rank	
215	0	
320	4	
589	4	
968	0	
2478	0	
2563	2	
2730	0	
3053	2	
3400	1	
3435	0	
3966	4	
3974	3	
4066	2	
4126	2	
4256	0	
4577	0	
4780	3	
4900	0	
5127	3	

Table 4.4: Maxrank for NBA Dataset

Summary:

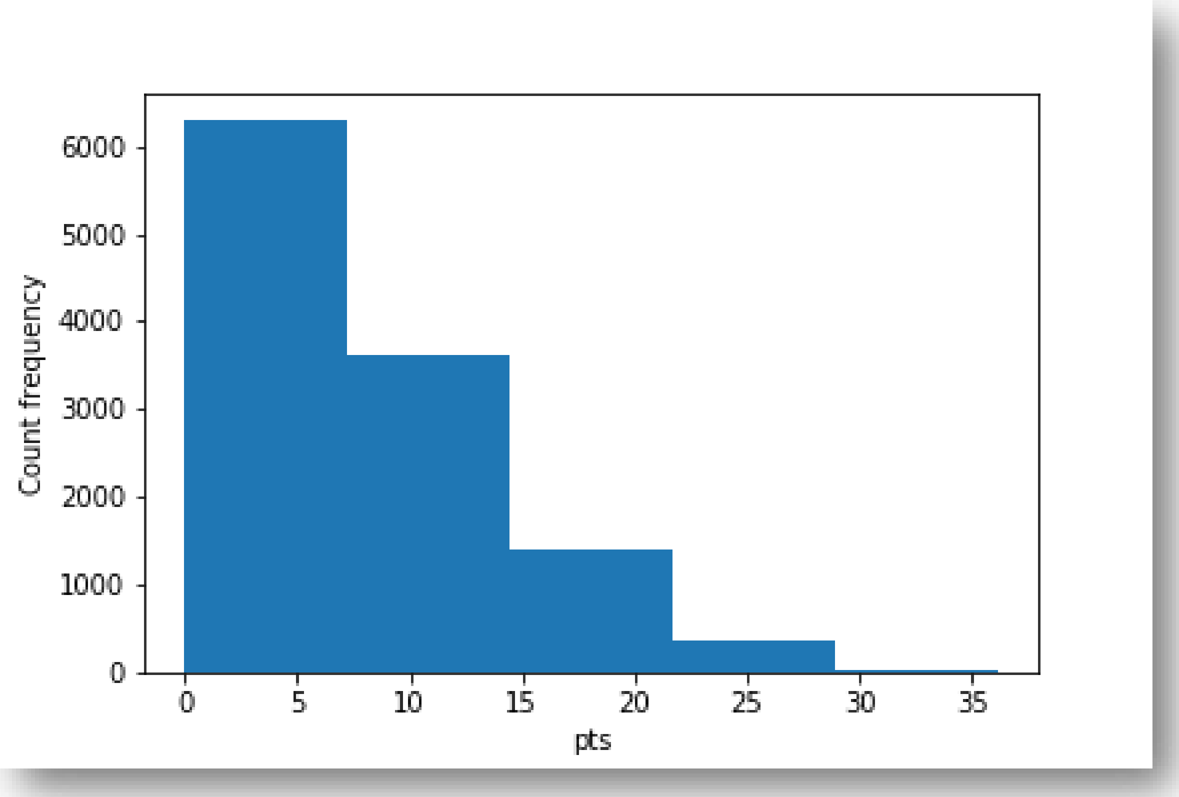


Figure 4.13: Histogram pts data

The histogram shows the distribution of the pts data of the NBA Dataset. It can be seen that the values for the pts data varies from 0 to 40. The highest number of values seem to be between the value of 0 and a little less than 10. The histogram gives the distribution of data with respect to the frequency of each data value specified on the y-axis. It is a good indicator of the distribution of values in a given dataset.

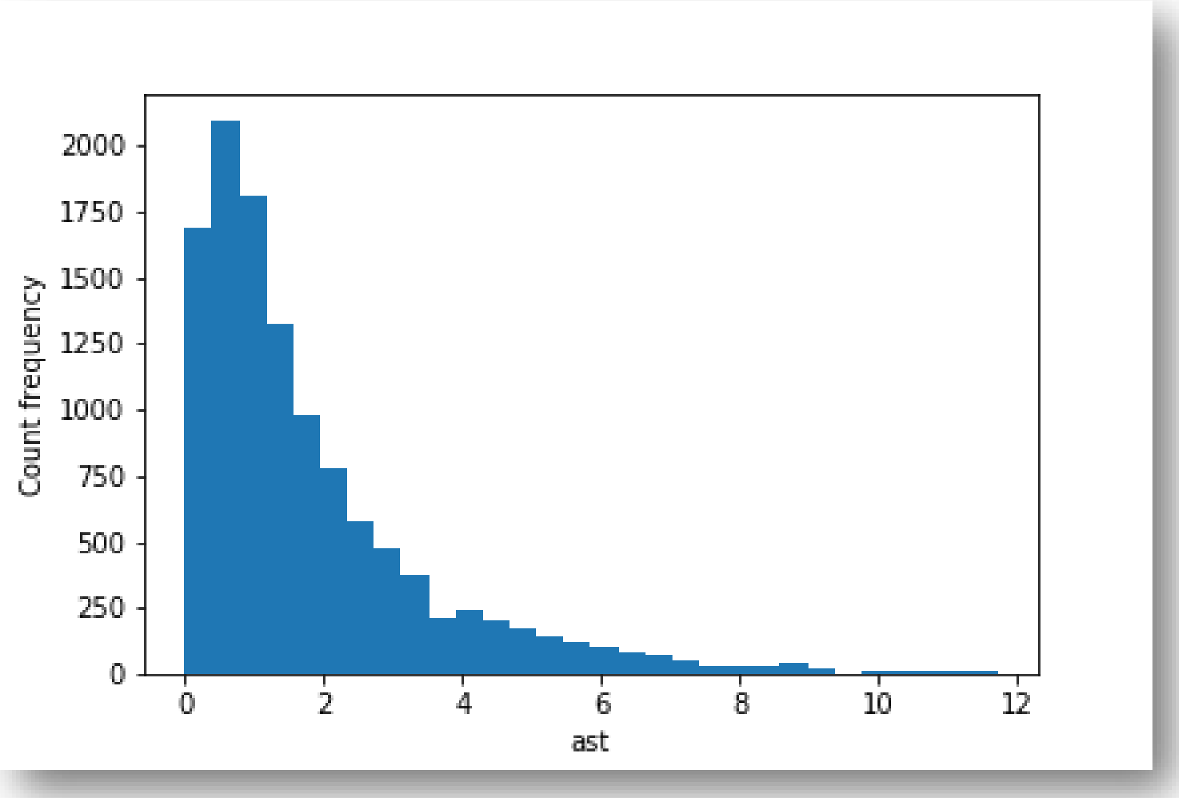


Figure 4.14: Histogram ast data

The histogram shows the distribution of values for the ast data in the NBA Dataset. We can see that the values for the ast data vary from 0 and 12. The highest frequency of ast data values is between 0 and 2 as can be seen from the plot above. The histogram has been plotted using the plot.hist function provided by Python.

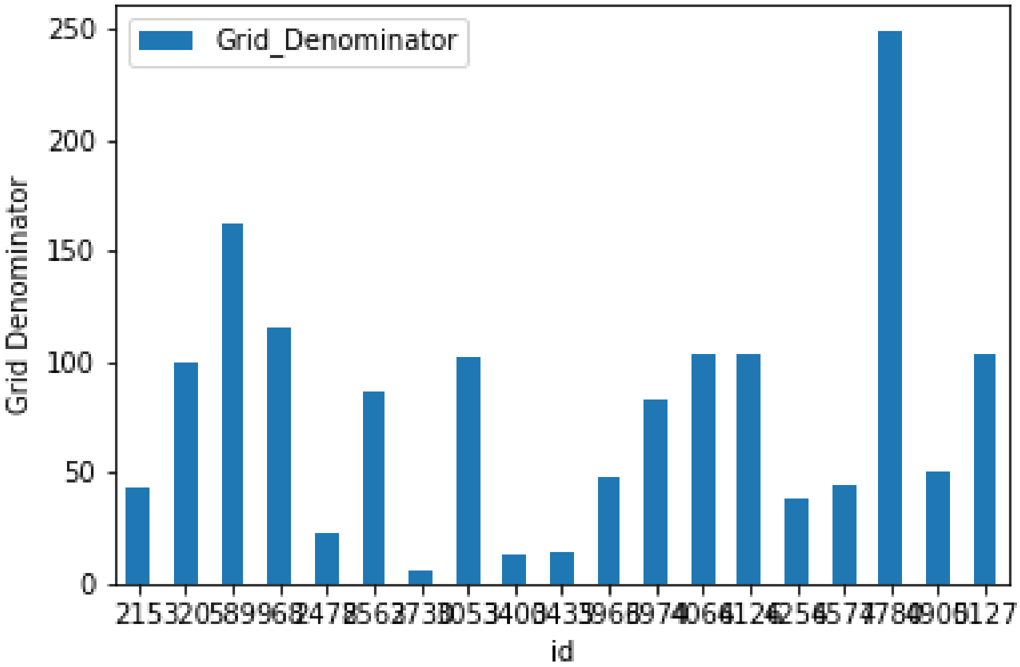


Figure 4.15: Grid denominator bar plot -NBA Dataset

The bar plot represents the various Grid Denominator values for the skyline points in the NBA Dataset. The plot has been generated using the plot.bar function in the Python programming language. With the bar plot it is easier to locate which of the skyline point has a low value for the Grid Denominator in our case. The Grid denominator here obviously varies from 0 to the value of 250 which was our initial value for the Grid cell size which corresponded to 1/250 = 0.004

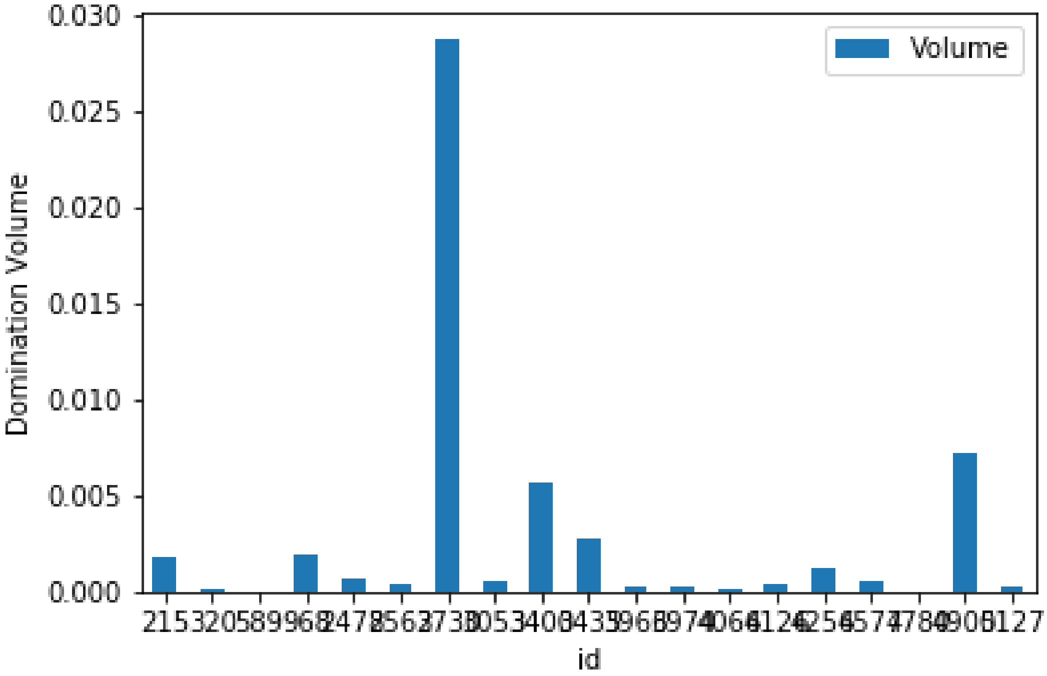


Figure 4.16: Domination Volume bar plot - NBA Dataset

The bar plot indicates the increase in Domination Volume for the various skyline points in the NBA Dataset. With the bar plot we can easily pinpoint which of the skyline points has a high value for the increase in domination volume. The domination volume has been calculated by using plot.bar function in Python programming language. The value for the increase in domination volume varies from 0.00 and 0.030 according to the y-axis. The index of the skyline points are given in the x-axis. The increase in domination volume values are given in the y-axis.

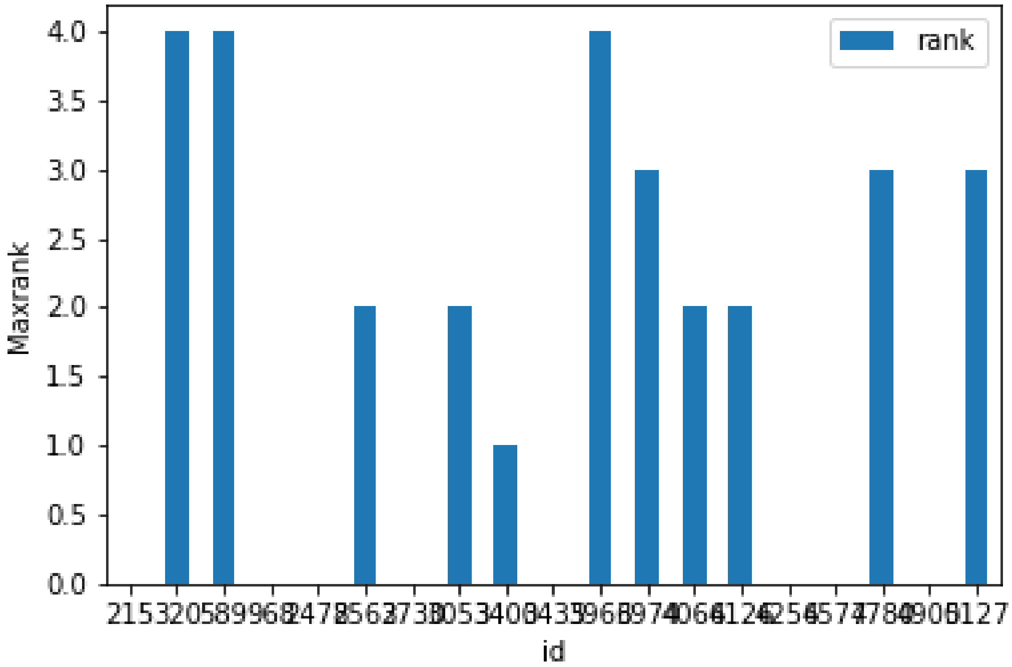


Figure 4.17: Maxrank bar plot - NBA Dataset

The bar plot indicates the Maxrank values for the skyline points in the dataset. The index values are given on the x-axis. The maxrank values are given in the y-axis. The bar plot makes it easier for us to see which of the points have a large maxrank value which might indicate that they are particularly interesting from a human standpoint.

Data Insight: Out of all the skyline points observed the point with id 3400 seems to be the best point according to our criterion. When all the skyline points with non-zero maxrank were observed it resulted in 3400 being dominant because of a relatively high domination volume of 0.00566 and a grid denominator value equal to 13. The rest of the skyline points did have a low grid denominator value but their domination volume value was low in comparison to the point 3400. Therefore, in the NBA Dataset the point with id 3400 seems to be the best point according to our criterion.

In the below datasets the Grid denominator, Increase in Domination Volume and the Maxrank values of the data points have been shown in a single plot along with the skyline of the dataset. The following notation has been used in the datasets given below -:

1. Grid denominator
   Points with red color – Grid denominator greater than 1 and less than or equal to 100
   Points with green color – Grid denominator greater than 100 and less than or equal to 200
   Points with blue color – Grid denominator greater than 200
2. Increase in Domination Volume
   Points which have a significant domination Volume with respect to the others are shown with a larger size.
3. Maxrank
   Points with plus (+) symbol – Maxrank value equal to 0
   Points with a upper triangle(^) symbol – Maxrank greater than 0 and less than or equal to 10.
   Points with a star (*) symbol – Maxrank value greater than 10

For the remaining 8 datasets, we have given their skyline plot along with the indicators and described the dataset with a short description. The interesting points to consider are also given for each dataset.

4.1.3 NBA Raptors Dataset

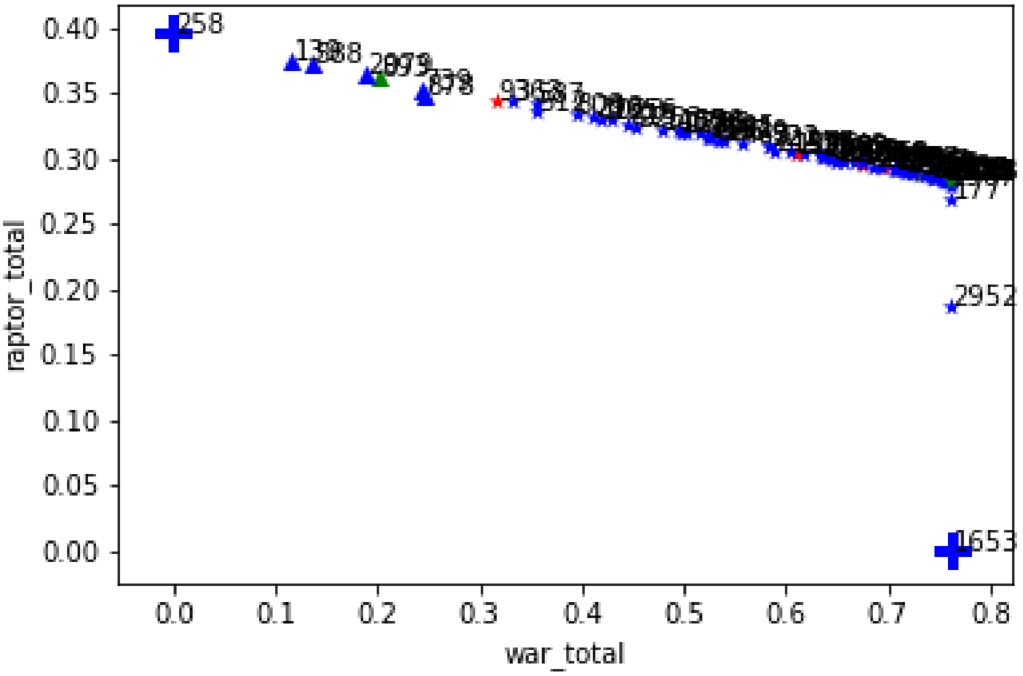


Figure 4.18: Skyline plot - NBA Raptors

Dataset Link: **https://www.kaggle.com/datasets/anandaramg/nba-players-according-to-raptor?select=train.csv**

Number of tuples/records: 340330

Number of attributes: 95

Attributes selected for analysis: poss_y and poss_x

Number of distinct values for attributes selected: 3311

Dataset Description: This dataset is large with about 340330 tuples. The dataset consists of NBA player ratings according to Raptor. The player ratings are taken from the year of 2014. The ratings consist of scores like the box score, plus or minus score and the overall Raptor total score. We have used the attributes poss_y and poss_x for our analysis.

Points of Interest: The point with id 258 seems to be the best point according to the above plot. However, it is a convex point. The points next to it with an upper triangle may be the points that we are looking for since they are concave and have a significant domination volume although lesser than the point 258.

4.1.4 Youtube Dataset

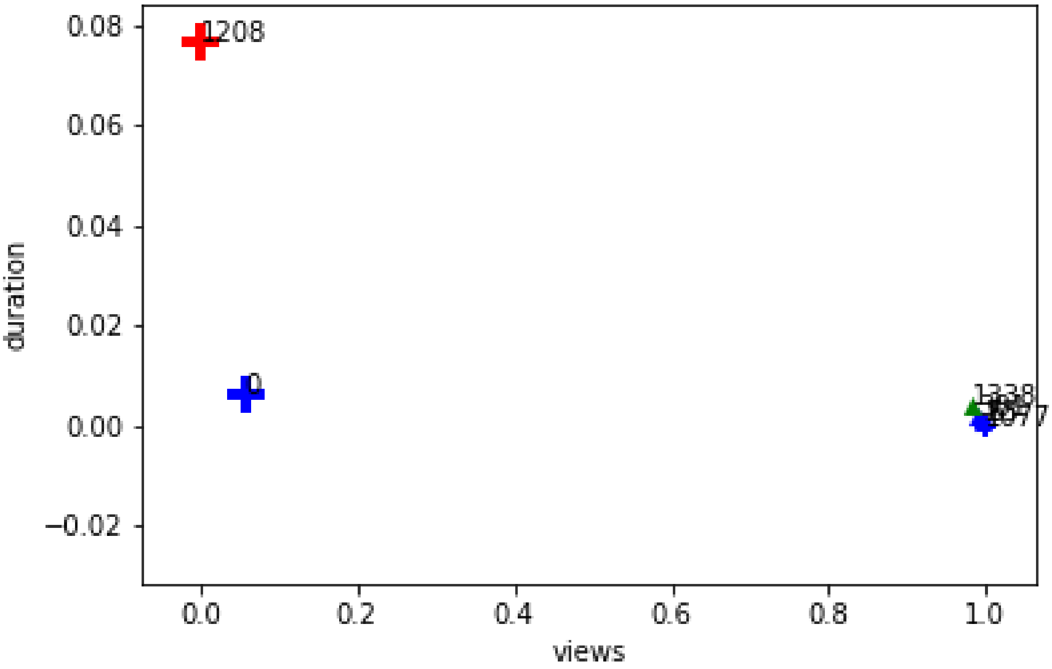


Figure 4.19: Skyline plot - Youtube Dataset

Dataset Link: **https://www.kaggle.com/datasets/demko1/youtube-oldest-videos2005-dataset**

Number of tuples/records: 1700

Number of attributes: 6

Attributes selected for analysis: views and duration

Number of distinct values for attributes selected: 1979

Dataset Description: This Dataset has been taken from Kaggle which is a website popular for data science. This is a dataset which consists of the Youtube's oldest videos from the year of 2005. This is a relatively small dataset with 1700 tuples and 6 attributes. The attributes of views and duration are taken for analysis. We want to find the skyline of the videos for which views are the high and the duration is low.

Points of Interest: The points 1208 and 0 have a maxrank value equal to zero(shown by a plus). These points have a large domination volume, but they are convex points. A large number of points are concentrated at the bottom right of the plot and it is difficult to predict which of the points are suitable as they might have a very low domination volume despite being concave. In this dataset possibly there are no such points present.

4.1.5 Undergrad Universities Dataset

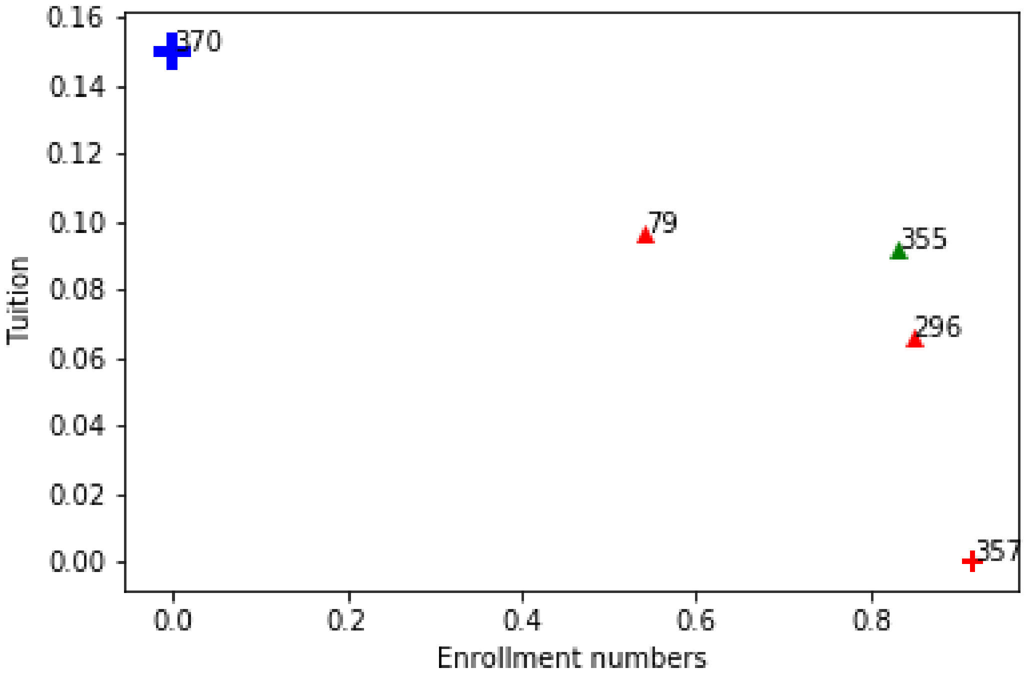


Figure 4.20: Skyline plot - Undergrad Universities Dataset

Dataset: https://www.kaggle.com/datasets/neelgajare/2022-usa-college-rankings-more

Total number of tuples: 392

Total number of attributes: 3

Selected attributes: tuition and enrollment numbers

Number of distinct values for tuition and enrollment numbers: 772

Description of Dataset – This dataset has been taken from the popular data science website Kaggle which has thousands of datasets posted every day. This particular dataset can be found using the link provided above. The dataset represents the ranking of universities in the United States of America as provided by US News. This dataset contains a total of 392 American universities ranked according to a given criteria. We have calculated the skyline points for this dataset using the Tuition and enrollment numbers. We are basically looking for universities with a low Tuition fee and a high enrollment number.

Points of Interest: The point with id 370 has a large domination volume but it is convex. The main point of interest here would be the point with id 79 since it is concave and it might dominate the other skyline points below it given that it has a relatively high domination volume compared those points.

4.1.6 Skyrim Weapons Dataset

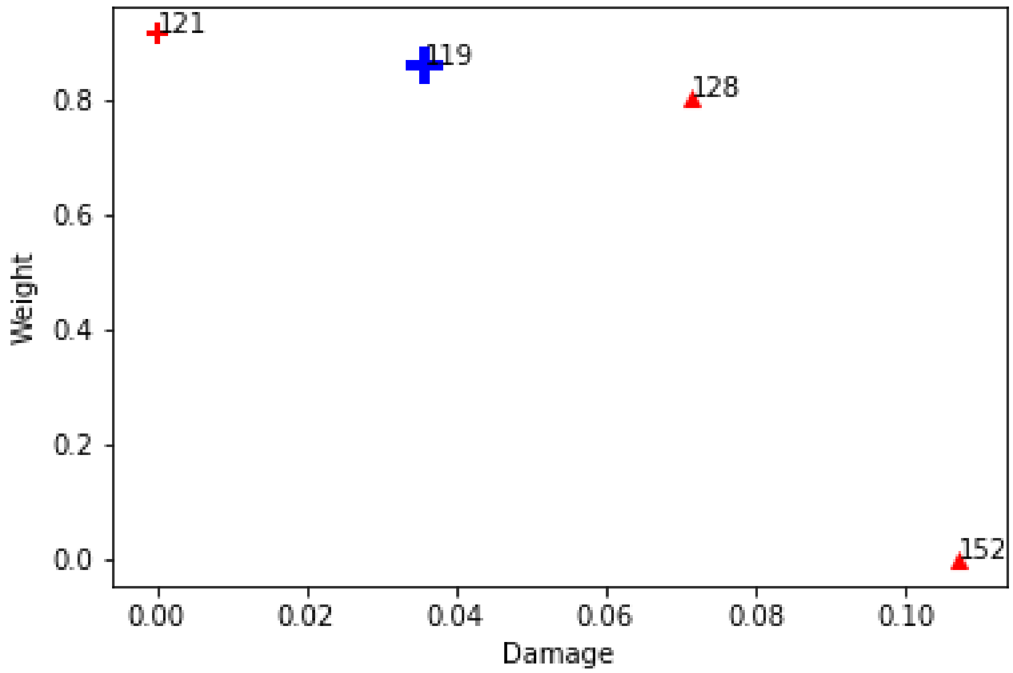


Figure 4.21: Skyline plot - Skyrim Weapons Dataset

Dataset link: https://www.kaggle.com/datasets/elmartini/skyrim-weapons-dataset

Number of tuples/records: 293

Total number of skyline points: 4

Total number of attributes:9

Selected attributes: Damage and Weight

Number of distinct values for Damage and Weight: 39

Description of Dataset: This dataset was taken from Kaggle. Skyrim is a game created by Bethesda Game Studios and it was produced by Bethesda Soft Works. It is the fifth game in the Skyrim series which was released after the fourth game Skyrim : Oblivion. The game was released in November 2011. This dataset specifies all the weapons that can be used by the player in the game.

Points of Interest: In this dataset the points 121 and 119 have a significant domination volume, they are convex points. The main point to check here would be the point with an id of 128 although this point might have a low domination volume since it is seen to dominate only one point with an id of 152 which is possibly the worst point in the dataset.

4.1.7 USA Cars Dataset

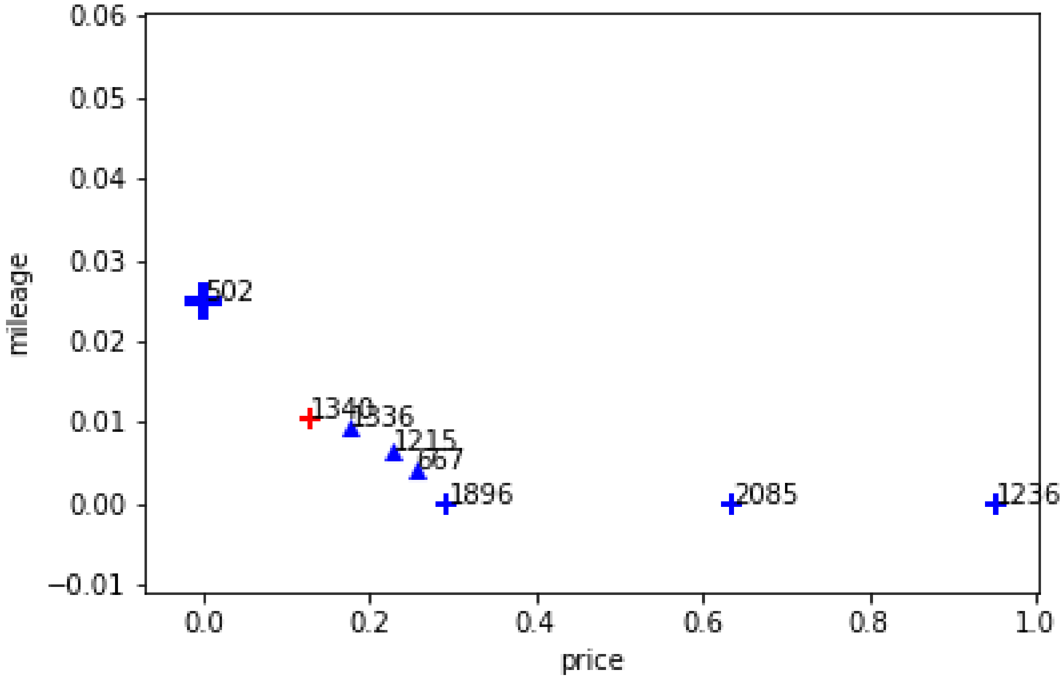


Figure 4.22: Skyline plot - USA Cars Dataset

Link to the dataset: https://www.kaggle.com/datasets/doaaalsenani/usa-cers-dataset

Number of tuples/records: 2499

Number of attributes: 12

Number of skyline points: 8

Attributes selected: price and mileage

Total number of distinct values for attributes chosen: 3205

Description of the dataset: The data for this dataset was scraped from AUCTION EXPORTS.com. This dataset represents the Automobiles for sale in the United States of America. This data consists of 12 features along with 2499 records/tuples. This is a relatively small dataset than what we have analyzed before.

Points of Interest: The point 502 has a large domination volume but it is a convex point. However, the main points of interest here would be 1336, 1215 and 567. The domination volume of these concave points have to be checked in order to conclude that these points are indeed interesting to consider.

4.1.8 Games of All Time Dataset

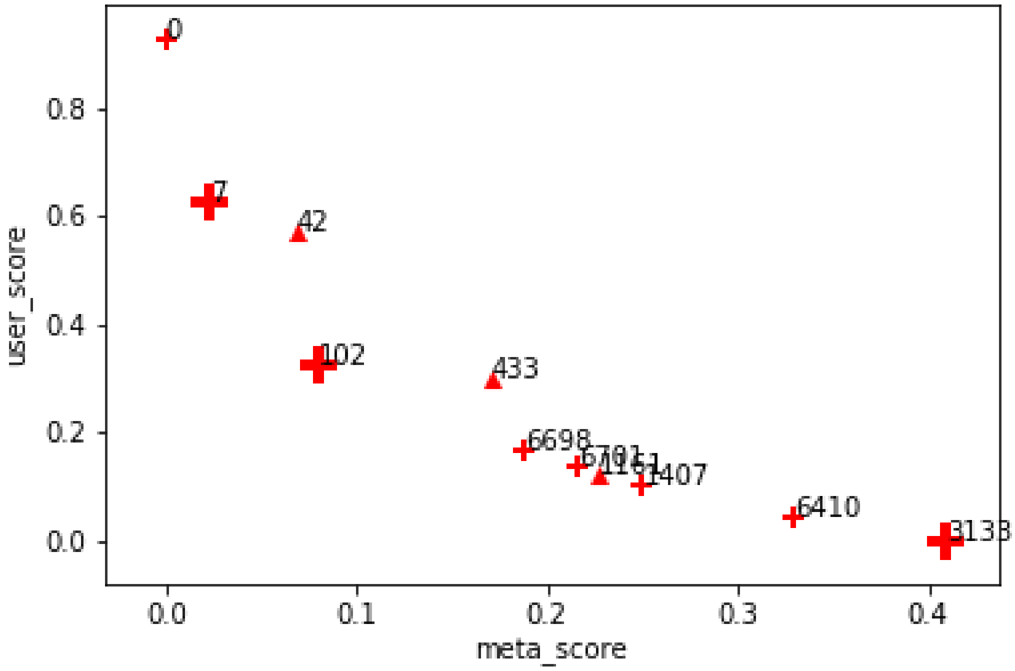


Figure 4.23: Skyline plot - Games Dataset

Dataset link: https://www.kaggle.com/datasets/xcherry/games-of-all-time-from-metacritic

Number of tuples/records: 8831

Number of attributes: 9

Number of skyline points: 11

Selected attributes: meta_score and user_score

Number of distinct values for the attributes: 463

Dataset description: This dataset gives a description of all the games which is reviewed by the popular Metacritic which is a website which reviews any new game that is released. This dataset consists of about 8000 data records with 9 attributes out of which 2 attributes are selected for the analysis of Skyline points.

Points of Interest: In this dataset, the points with id 7 and 102 have a significant domination volume but these are convex points. The main points of interest here would be the points with id 42 and id 433 since these are concave and they might have a relatively higher domination volume than the points that are below it.

4.1.9 Perth House Prices Dataset

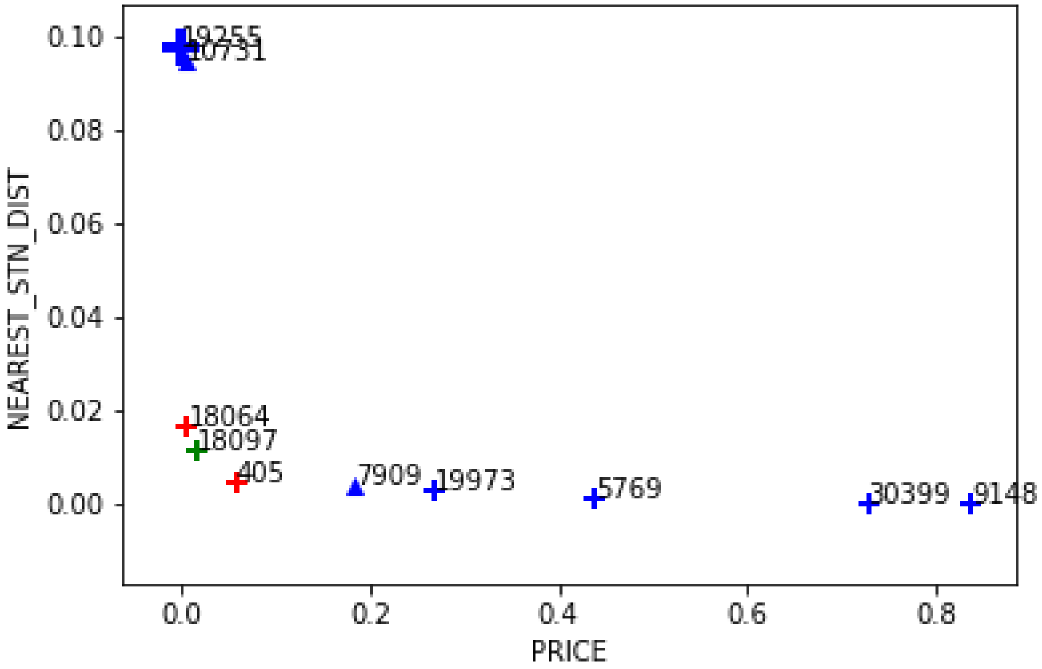


Figure 4.24: Skyline plot - Perth Housing Dataset

Dataset Link: https://www.kaggle.com/datasets/syuzai/perth-house-prices

Number of tuples/records: 33656

Number of attributes: 18

Attributes selected for analysis: PRICE and NEAREST_STN_DIST

Number of distinct values for attributes selected: 3486

Dataset Description: This Dataset has been taken from Kaggle which is a website popular for data science. This is a relatively bigger dataset with about 30,000 tuples/records. The dataset consists of information about the house prices in 322 suburbs in Perth in Austrailia. There are 100 rows in the dataset per suburb approximately. We have taken two attributes namely PRICE and NEAREST_STN_DIST and we try to find and analyze skyline points depending on the houses that have a high price and are closest to the nearest station in Perth in the dataset.

Points of Interest: The points with id 19255 and id 10731 seem to have a large domination volume. They are convex points. Most of the points in this dataset are seen to be convex and if there are any concave points here it does not seem likely that they have a significant domination volume.

4.1.10 Best Bowling Dataset

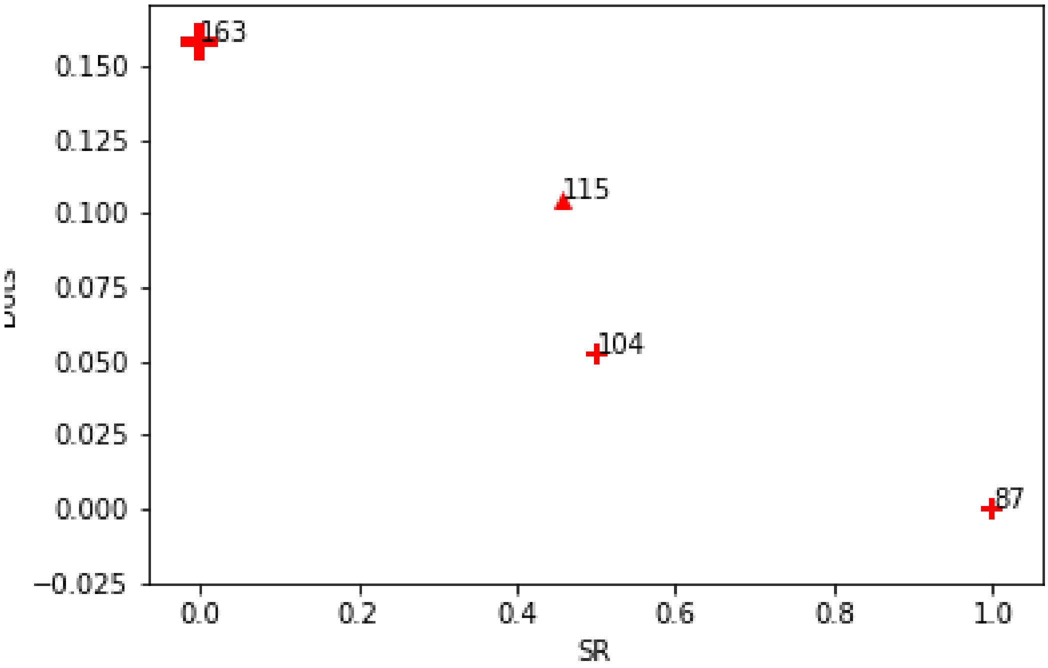


Figure 4.25: Skyline plot - Bowling Dataset

Dataset Link: -

Total number of tuples/records: 2100

Total number of skylines: 4

Total number of attributes: 10

The attributes selected: Runs and SR (Strike Rate)

The number of distinct values for the attributes selected: 44

Dataset description: This dataset contains almost all the greatest players to have ever played the game of Cricket. This dataset was taken from Kaggle. It consists of 2100 records and 10 attributes out of which Runs and Strike Rate were the two attributes that are chosen for our analysis.

Points of Interest: The point with id 163 has a large domination volume but this point is convex. The only interesting point to consider in this dataset is the point 115 which is concave and which might have a significant domination volume since it seems to dominate the points with id 104 and id 87.

From the experiments we have proved that there are points that are interesting from a human perspective that can provide a good tradeoff considering the options that we have with us. We essentially focused on points that had a high value for the Grid strength, high value for Increase in Domination Volume and a considerable value for the maxrank indicating that it does not belong on the convex hull and therefore is hard to be found by a top-1 linear function. As the number of dimensions increase, it becomes increasingly difficult to pinpoint the points in the skyline that can be potentially interesting by just looking at the plot. A mere 3-d dataset can be hard to infer as we have three axes x,y, and z in the three dimensions.

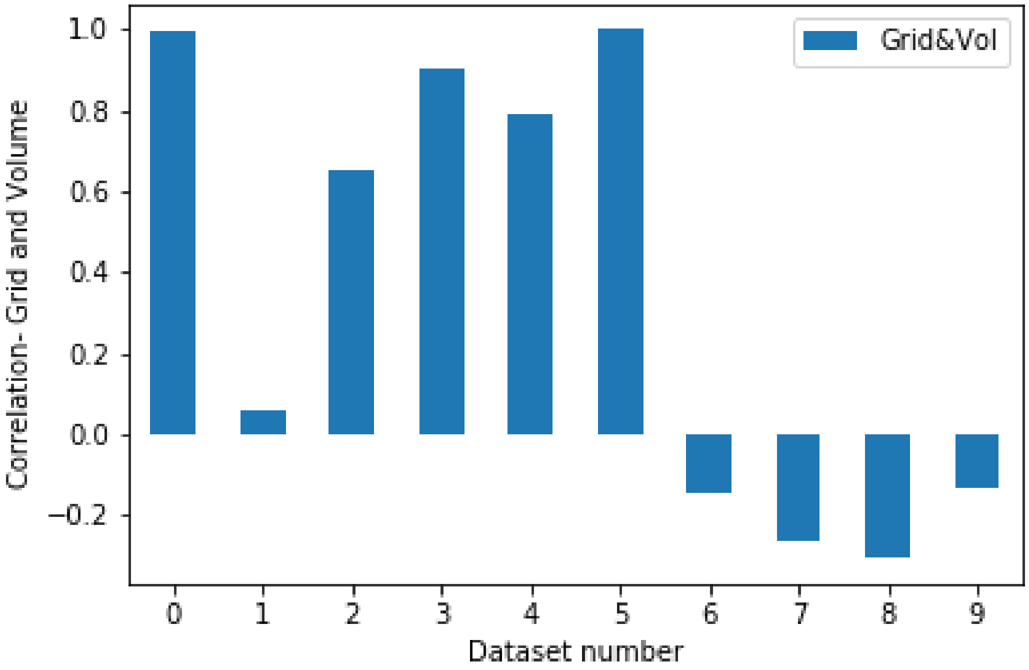


Figure 4.26: Correlation Grid and Volume

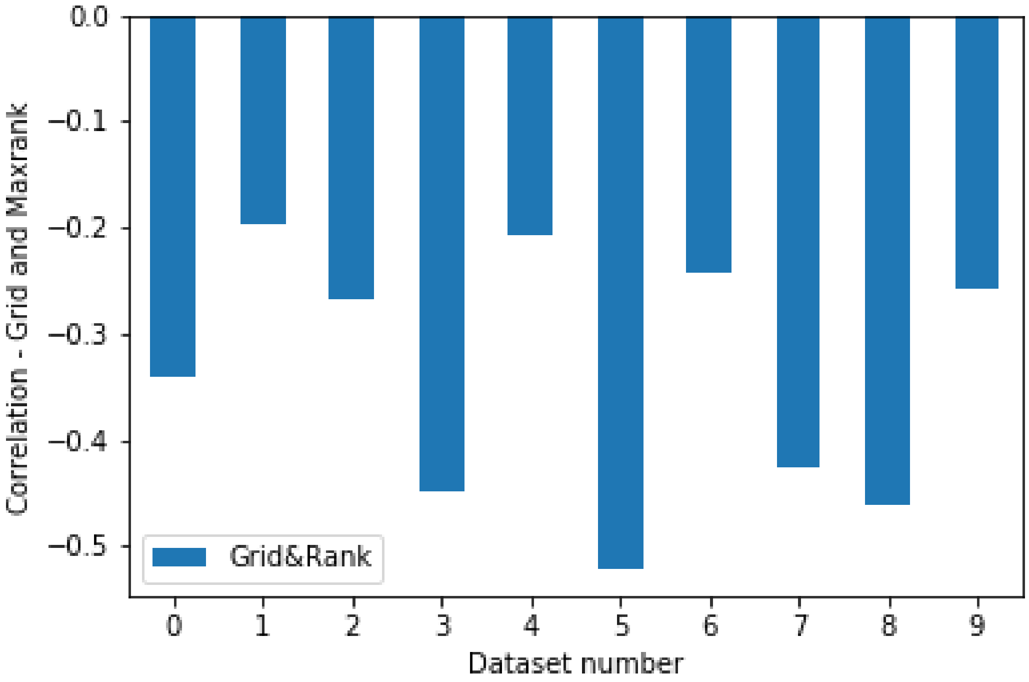


Figure 4.27: Correlation Grid and Maxrank

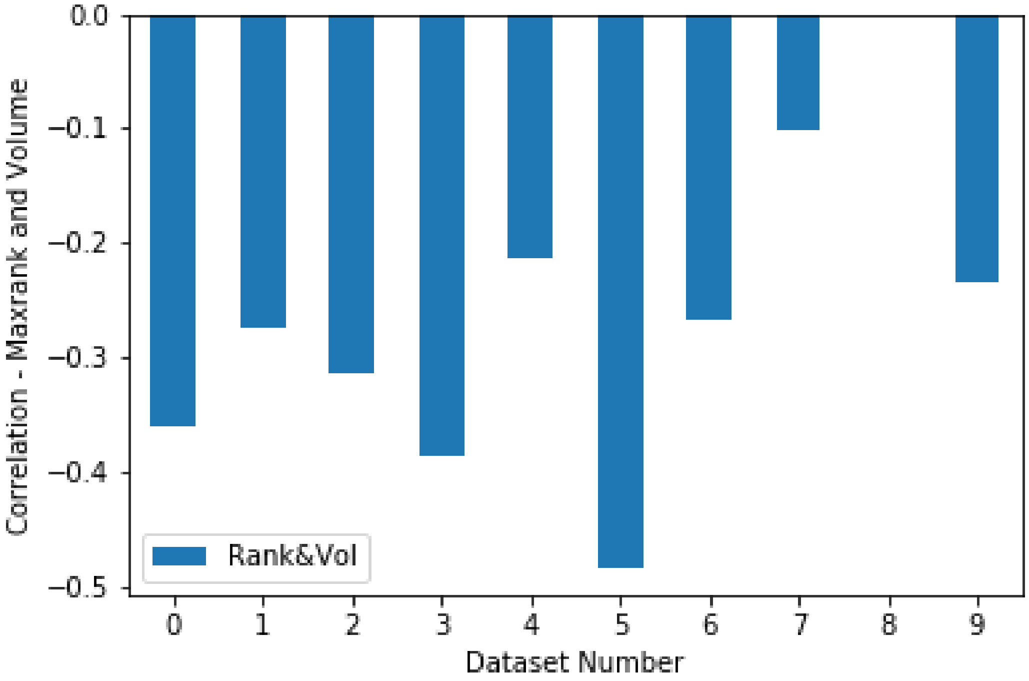


Figure 4.28: Correlation Maxrank and Volume

The above bar plots depict the correlation between -:

1. Grid strength and Domination Volume
2. Grid strength and Maxrank
3. Maxrank and Domination Volume

The Datasets and their corresponding numbers are given below -:

0 – Bowling Dataset

1 – Perth Housing Dataset

2 – Games of All Time Dataset

3 – NBA Dataset

4 – NBA Raptors Dataset

5 – Undergrad Universities Dataset

6 – USA Cars Dataset

7 – Youtube Videos Dataset

8 – Skyrim Dataset

9 – Wines Dataset

Pearson r – correlation coefficient was calculated for the three cases in all the 10 2-D datasets. A positive correlation indicates that as x increases y also increases and a negative correlation between x and y indicates that if x increases then y is more likely to decrease.

1. Grid strength and Domination Volume

   The correlation between Grid strength and Domination Volume is mostly positive as seen from the bar plot. Only the datasets – USA Cars, Youtube, Skyrim and Wines have a negative correlation value. We can therefore conclude that as Grid strength increases, an increase in Domination Volume is expected.

2. Grid strength and Maxrank

   The correlation between Grid strength and Maxrank is negative as seen from the bar plot. There is a variation in the amount of correlation between the two indicators for the 10 datasets but the value has been negative throughout. Hence, we can come to a conclusion that as the Grid strength increases the Maxrank of the datapoint is expected to decrease.

3. Maxrank and Domination Volume

   The correlation between Maxrank and Domination Volume is negative for all the datasets observed. As the Maxrank value increases this would lead to a decrease in Domination Volume for that data point.

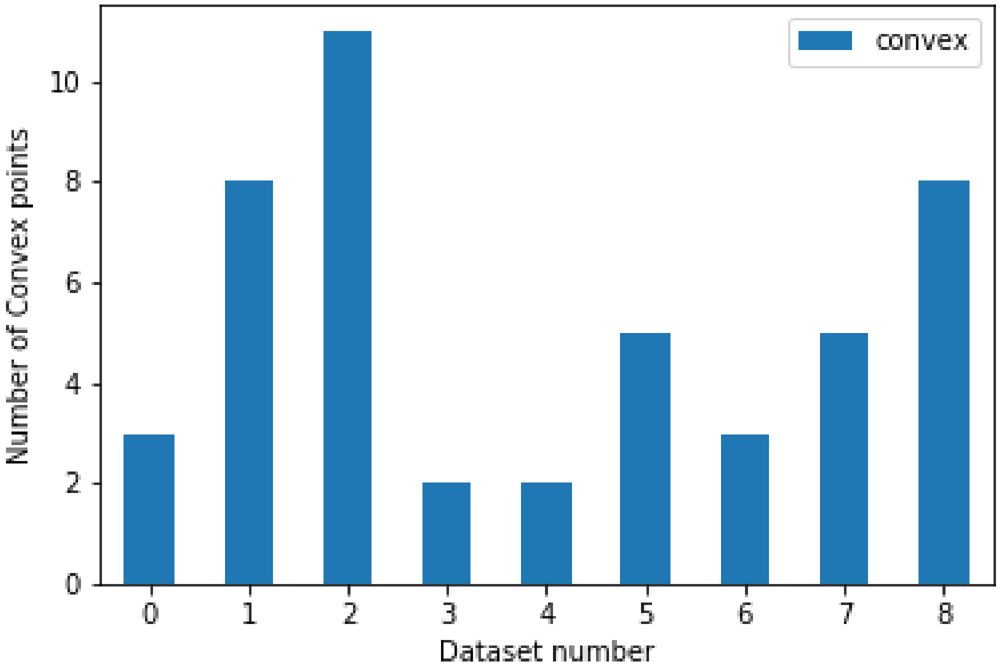


Figure 4.29: Bar plot number of convex points

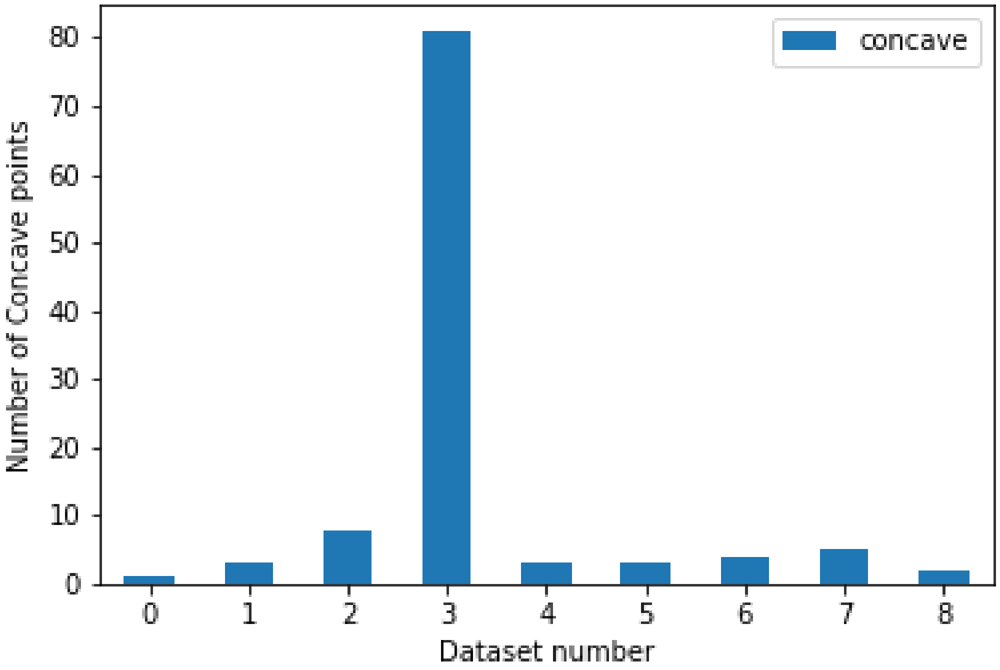


Figure 4.30: Bar plot number of concave points

The above bar plot for the 10 datasets indicates -:

1. Number of convex skyline points
2. Number of concave skyline points

The number of concave skyline points is significantly lower than the number of convex skyline points among all the skyline points in the dataset.

In the next part of the thesis we move on to the conclusion where we make a final statement on the results of the experiments done so far.

5. Conclusion

In conclusion we can say, given by the experiments performed that points that are interesting from the user's perspective and that seem to provide a good compromise do exist in the skyline of the dataset. Out of the 10 datasets seen, most of the datasets had some interesting points to consider that might be potentially good options for the user. We have also seen the correlation between the different indicators and it shows that the correlation between Grid strength and Volume is mainly positive and that between Grid Strength and Maxrank and Maxrank and Volume was mainly negative.

We have only considered datasets of 2 Dimensions here and it might be difficult to predict the existence of these points in higher dimensions (3 or more) given the indicators. For a 3-D dataset it becomes difficult to visually see from the skyline plot which of the points may be convex or concave and there might be other indicators that have to be introduced in order to make the detection of such points in higher dimensions easier. Therefore, a possible continuation of this search for points can be to introduce new indicators that would help in higher dimensions.

6. References

1. Paolo Ciaccia , Davide Martinenghi : Reconciling Skyline and Ranking Queries Proc VLDB Endow 10 (11) : 1454 – 1465 (2017)
2. Kyriakos Mouratidis , Bo Tang : Exact Processing of Uncertain Top-k Queries in Multi – Criteria settings. Proc VLDB Endow 11 (8) : 866 – 879 (2018)
3. Paolo Ciaccia , Davide Martinenghi – FA + TA < FSA : Flexible Score Aggregation. CIKM 2018: 57 – 66.
4. Paolo Ciaccia , Davide Martinenghi : Flexible Skylines : Dominance of Arbitrary Sets of Monotone Functions. ACM Trans Database Systems 45(4): 18 : 1 – 18:45 (2020)
5. Kyriakos Mouratidis , Keming Li, Bo Tang : Marrying Top-k with Skyline Queries : Relaxing the Preference Output while producing output of considerable size. SIGMOD Conference 2021- 1317 – 1330.
6. Jan Chomicki, Paolo Ciaccia, Nicolo Meneghetti : Skyline Queries, Front and Back ACM SIGMOD Record September 2013 pp -6-18
7. Chirstian Bohm, Frank Fiedler, Annahita Oswald, Claudia Plant, Bianca Wackerseuther : Probabilistic Skyline queries CIKM ’09 Proceedings of the 18th ACM conference on information and knowledge management November 2009 – pages 651-660
8. Louis Woods, Gustavo Alonso, Jens Teubner: Parallel Computation of Skyline Queries 2013 IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines
9. Matteo Magnani, Ira Assent: From stars to galaxies: skyline queries on aggregate data EDBT '13: Proceedings of the 16th International Conference on Extending Database Technology March 2013 Pages 477–488

10. Muhammad Aamir Cheema, Xeumin Lin, Wenjie Zhang, Ying Zhang: A safe zone-based approach for monitoring moving skyline queries EDBT '13: Proceedings of the 16th International Conference on Extending Database Technology March 2013 Pages 275–286
11. Laura Lampariello - Indicatori Originali per Caratterizare La Rilevanza Dei Punti Dello Skyline Alma Mater Studorium Universita Di Bologna Sessione 3 – Anno Accademico – 2020/21.
    This thesis is the continuation of the above referenced thesis.

List of Figures

List of Tables