



Thales

ANALYTICS



Veri
Ambarı



Veri
Analizi



Veri
Yönetiřimi



İř
Zekası

Veri Analizi

"Veri Analizi: Bilgiye Dayalı Kararlar Alınmasını Sağlar!"

Her gün iş dünyasında yığınla veri üretilmekte ve bu veriler hızla birikmektedir. Ancak verilerin yalnızca toplanması yeterli değildir; bu verilerin analiz edilip değerli bilgilere dönüştürülmesi gerekir. Tam da burada Veri Analizi devreye girmektedir!

Veri Analizi Nedir?

Veri analizi, işletmenin sahip olduğu verilerin sistematik bir şekilde incelenmesini, içerdikleri anlamlı bilgilerin açığa çıkarılmasını ve iş kararlarına yön verecek içgörüler elde edilmesini sağlayan bir süreçtir. Ham veriler üzerinde yapılan analizlerle işletmenin farklı alanlarına dair detaylı bilgi sağlanır ve stratejik bir bakış açısı geliştirilir.

Veri Analizinin Faydaları Nelerdir?

Hızlı ve Doğru Karar Verme

Veri analizi sayesinde işletme verileri detaylı bir şekilde incelenir ve bu incelemeler ışığında güvenilir ve doğru kararlar alınması sağlanır. İşletmenin performansı ölçülerek süreçlerdeki iyileştirme alanları tespit edilir.

Müşteri Davranışlarının Anlaşılması

Müşterilere dair veriler analiz edilerek onların davranışları ve tercihleri anlaşılır. Bu sayede müşteri ihtiyaçlarının önceden öngörülmesi ve kişiselleştirilmiş deneyimler sunulması kolaylaştırılır.

Rekabet Üstünlüğü Sağlanır

Pazar verilerinin analiz edilmesi sayesinde sektördeki değişiklikler, müşteri trendleri ve rekabet koşulları hızla takip edilir. Bu analizler, işletmenin rekabette bir adım öne geçmesine olanak tanır.

Operasyonel Verimlilik Artırılır

Veri analizleri ile süreçlerdeki gereksiz maliyetler ve verimsizlikler tespit edilir. Bu sayede kaynakların daha etkin kullanılması sağlanır ve operasyonel verimlilik artırılır.

Veri Ambarı

Veri Ambarı: Kararların Veriye Dayandırılmasını Sağlar!

Her gün iş dünyasında binlerce veri üretilmektedir: satış raporları, müşteri geri bildirimleri, pazarlama kampanya sonuçları ve daha fazlası. Ancak bu veriler farklı kaynaklarda ve sistemlerde saklandığında, değerli bilgilerin keşfedilmesi zor hale gelir. Tam da burada Veri Ambarı devreye girmektedir!

Veri Ambarı Nedir?

Veri ambarı, firmanın tüm verilerinin tek bir yerde toplanmasını, organize edilmesini ve analiz edilebilir hale getirilmesini sağlayan bir sistemdir. Farklı kaynaklardan gelen veriler (satışlar, müşteri bilgileri, finansal veriler, vb.), veri ambarında bir araya getirilerek işletmenin tüm veriyi tek bir platformdan anlamlı bilgiye dönüştürmesine olanak tanınır.

Veri Ambarının Faydaları Nelerdir?

Güçlü ve Hızlı Karar Verme

Veri ambarı sayesinde tüm verilere tek bir yerden ulaşıp analiz yapılabilir. Bu sayede karar alırken zaman kazandırılır ve daha isabetli, veriye dayalı kararlar alınabilir.

Tüm Veriler Tek Çatı Altında

Farklı departmanlardan gelen veriler bir araya getirilir ve bütüncül bir bakış açısı sunulur. Pazarlama, finans, satış gibi birimlerin verileri entegre edilir, böylece iş hakkında daha geniş bir resim elde edilmesi sağlanır.

Müşteri Deneyimi İyileştirilir

Veri ambarı sayesinde müşteri davranışlarının analiz edilmesi kolaylaştırılır. Bu analizler doğrultusunda müşterilere daha kişiselleştirilmiş deneyimler sunulması ve ihtiyaçlarının önceden tahmin edilmesi mümkün hale getirilir.

Rekabet Avantajı Sağlanır

Veri ambarı ile pazardaki değişimlerin ve müşteri trendlerinin daha hızlı analiz edilmesi sağlanır. Bu durum, rakiplerin önüne geçilmesine katkıda bulunur.

iş Zekası

İş Zekası: Verinin Anlamalı Bilgiye Dönüştürülmesi Sağlanır!

Veri yığınlarının içinde saklı olan değerli bilgilerin gün yüzüne çıkarılması, işletmelerin başarıya ulaşmasında kritik bir rol oynar. İş zekası (BI), bu verilerin analiz edilerek anlamalı bilgilere dönüştürülmesini sağlar ve işletme stratejilerinin güçlenmesine katkıda bulunur.

İş Zekası Nedir?

İş zekası, işletmeye ait verilerin toplanması, işlenmesi ve analiz edilmesi sürecidir. Bu süreçte gelişmiş raporlama ve analiz araçları kullanılarak, işletme performansını ölçen, öngörüler sunan ve karar destek sistemleriyle yöneticilere stratejik bilgiler sağlayan bir yapı oluşturulur.

İş Zekasının Faydaları Nelerdir?

Veriye Dayalı Karar Alma İmkânı Sağlanır

İş zekası ile işletme verileri detaylı analiz edilerek yöneticilerin doğru ve zamanında karar alması sağlanır. Bu durum, işletmenin riskleri azaltmasına ve fırsatları değerlendirmesine yardımcı olur.

Performans İzlemesi Kolaylaştırılır

İş zekası araçları sayesinde, işletmenin tüm departmanlarındaki performans izlenir. Belirlenen hedeflere ne kadar yaklaşıldığı analiz edilir ve gerekli iyileştirmeler yapılabilir.

Müşteri Davranışları Hakkında Derin Bilgiler Elde Edilir

Müşterilere dair veriler analiz edilerek onların ihtiyaçları ve beklentileri hakkında detaylı bilgi sağlanır. Bu sayede, müşteri memnuniyeti artırılır ve müşteri sadakati güçlendirilir.

Rekabet Üstünlüğü Sağlanır

Pazar trendleri ve rakip analizleri ile sektördeki değişimler izlenir, böylece işletmenin rekabette avantaj elde etmesi sağlanır.



Veri Yönetiřimi

Veri Yönetiřimi: Verinin Doğru Yönetilmesi Sağlanır!

Günümüz iş dünyasında verinin stratejik bir kaynak olarak etkin bir şekilde yönetilmesi büyük önem taşır. Ancak, sadece veriyi toplamak ve analiz etmek yeterli değildir; bu verinin doğru, güvenli ve uyumlu bir şekilde yönetilmesi gerekir. İşte bu noktada Veri Yönetiřimi devreye girmektedir!

Veri Yönetiřimi Nedir?

Veri yönetiřimi, işletmenin sahip olduđu verinin doğru şekilde toplanmasını, saklanmasını, düzenlenmesini ve kullanımını sağlamak için uygulanan standartlar ve süreçlerin bütünüdür. Veri kalitesini, güvenliğini ve uyumluluğunu sağlamak amacıyla politikalar, prosedürler ve teknolojiler kullanılarak verinin işletme içindeki yaşam döngüsünün her aşaması kontrol altına alınır.

Veri Yönetişiminin Faydaları Nelerdir?

Veri Kalitesinin Güvence Altına Alınması

Veri yönetiřimi sayesinde, iřletmenin verilerinin güncel, doğru ve güvenilir olması saęlanır. Bu sayede, alınacak kararların güvenilir verilere dayanması saęlanır.

Veri Güvenlięinin Saęlanması

Veri yönetiřimi politikalarıyla, iřletmenin sahip olduęu verilerin yetkisiz erişimlerden korunması ve veri güvenlięinin saęlanması hedeflenir. Özellikle müşteri ve çalışan verilerinin gizlilięi güvence altına alınır.

Yasal Uyumun Kolaylaştırılması

Veri yönetiřimi ile yasal ve düzenleyici gereksinimlere uygun veri yönetimi saęlanır. Bu durum, veri gizlilięi ve koruma yasalarına uyum sürecini kolaylaştırır ve olası cezai yaptırımlardan kaçınılmasına yardımcı olur.

Verinin Deęerli Bir Kaynaęa Dönüřtürülmesi

Veri yönetiřimi ile verinin anlamlı ve stratejik bir kaynaęa dönüřtürülmesi saęlanır. İřletmenin veri kullanımı optimize edilir ve iş süreçlerine entegre edilerek katma deęer oluşturulur.



Nasıl Yapıyoruz ?

Data Governance

Data Sources

Enterprise Data Hub

Apps, Portals & Mobile BI

Extract Transform Load

1

Staging

2

DWH

3

A

A

A

Master Data

Data Lake

Data Analysis

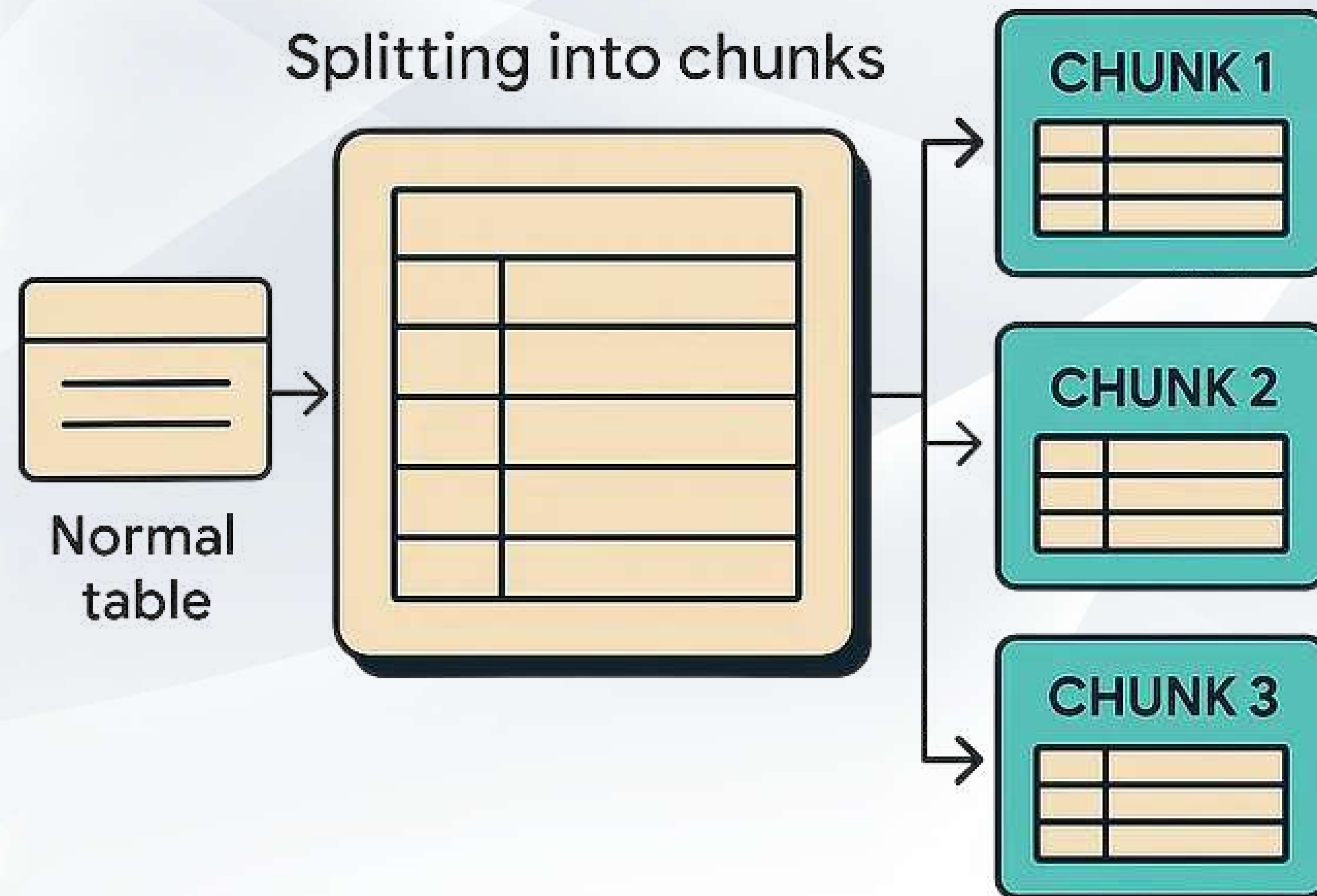


Timescale

*Thales suyun akışında düzen görmüştü.
Biz o düzeni verinin akışında arıyoruz.*

HYPERTABLES

IN TIGERDATA



Örnek Bir Uygulama...

---Tablo Boyutu

```
SELECT
    schemaname || '.' || relname AS table_name,
    pg_size_pretty(pg_total_relation_size(relid)) AS total_size,
    pg_size_pretty(pg_relation_size(relid)) AS table_size,
    pg_size_pretty(pg_total_relation_size(relid) -
    pg_relation_size(relid)) AS index_size
FROM
    pg_catalog.pg_statio_user_tables
where schemaname = 'schema_name' and relname = 'table_name'
ORDER BY
    pg_total_relation_size(relid) DESC;
```

---Insert edilen örnek verinin kaç günlük olduğu

```
select max(transaction_time) - min(transaction_time)
from tmp_schema.coin_data
```

---Günler içerisindeki satır sayılarının dağılımını öğrenme

```
select time_bucket('1 day', transaction_time), count(*)
from tmp_schema.coin_data
group by 1
```

$$\left(\frac{\text{Days}}{\text{TableSize}} \right) \cdot \text{Memory} \cdot 0.25 = \text{chunk_time_interval}$$

Approximate chunt_time_interval estimate

```
CREATE TABLE tmp_schema.coin_data (
  transaction_time timestamp NULL,
  symbol varchar(15) NULL,
  open_value float4 NULL,
  high_value float4 NULL,
  low_value float4 NULL,
  close_value float4 NULL,
  volume_coin float4 NULL,
  volume_usdt float8 NULL,
  trade_count int4 NULL
);
```

---Hypertable oluşturma (Tablo boşsa)

```
SELECT create_hypertable('tmp_schema.coin_data', by_range('transaction_time', INTERVAL '1 week'));
```

---Hypertable oluşturma (Tabloda hali hazırda veri varsa)

```
SELECT create_hypertable('tmp_schema.coin_data' by_range('transaction_time', INTERVAL '1 week'),migrate_data => true);
```

Query	insert into tmp_schema.coin_data select *
Updated Rows	from tmp_schema.coin_data_rep 23701328
Execute time	44s
Start time	Tue Jul 15 17:14:55 TRT 2025
Finish time	Tue Jul 15 17:15:39 TRT 2025

_timescaledb_internal	
Tables	
> _hyper_3_1993_chunk	25M
> _hyper_3_1994_chunk	28M
> _hyper_3_1995_chunk	31M
> _hyper_3_1996_chunk	31M
> _hyper_3_1997_chunk	32M
> _hyper_3_1998_chunk	32M
> _hyper_3_1999_chunk	32M
> _hyper_3_2000_chunk	32M
> _hyper_3_2001_chunk	32M
> _hyper_3_2002_chunk	32M
> _hyper_3_2003_chunk	32M
> _hyper_3_2004_chunk	32M
> _hyper_3_2005_chunk	32M
> _hyper_3_2006_chunk	28M
> _hyper_3_2007_chunk	15M
> _hyper_3_2008_chunk	20M
> _hyper_3_2009_chunk	20M

---Tabloya ait chunk tabloları gösterme

```
SELECT show_chunks('tmp_schema.coin_data');
```

A-Z show_chunks
_timescaledb_internal.hyper_3_1993_chunk
_timescaledb_internal.hyper_3_1994_chunk
_timescaledb_internal.hyper_3_1995_chunk
_timescaledb_internal.hyper_3_1996_chunk
_timescaledb_internal.hyper_3_1997_chunk
_timescaledb_internal.hyper_3_1998_chunk
_timescaledb_internal.hyper_3_1999_chunk
_timescaledb_internal.hyper_3_2000_chunk
_timescaledb_internal.hyper_3_2001_chunk
_timescaledb_internal.hyper_3_2002_chunk
_timescaledb_internal.hyper_3_2003_chunk
_timescaledb_internal.hyper_3_2004_chunk
_timescaledb_internal.hyper_3_2005_chunk
_timescaledb_internal.hyper_3_2006_chunk

Select Kolaylığı

```
select time_bucket('1 year',transaction_time) as transaction_year,  
symbol as coin_name,  
round(avg(open_value)::numeric,2) as avg_open_value  
from tmp_schema.coin_data  
where symbol = 'BTC/USDT'  
group by 1,2  
order by 1
```

transaction_year	A-Z coin_name	123 avg_open_value
2019-01-01 00:00:00.000	BTC/USDT	8,290.48
2020-01-01 00:00:00.000	BTC/USDT	11,064.88
2021-01-01 00:00:00.000	BTC/USDT	47,387.39

```
select time_bucket('1 day',transaction_time) as transaction_year,  
symbol as coin_name, round(avg(open_value)::numeric,2) as avg_open_value,  
round(avg(close_value)::numeric,2) as avg_close_value,  
round(avg((high_value+low_value)/2)::numeric,2) as avg_value_high_low  
from tmp_schema.coin_data  
where time_bucket('1 year',transaction_time) between '2020-01-01' and '2021-01-01'  
group by 1,2
```

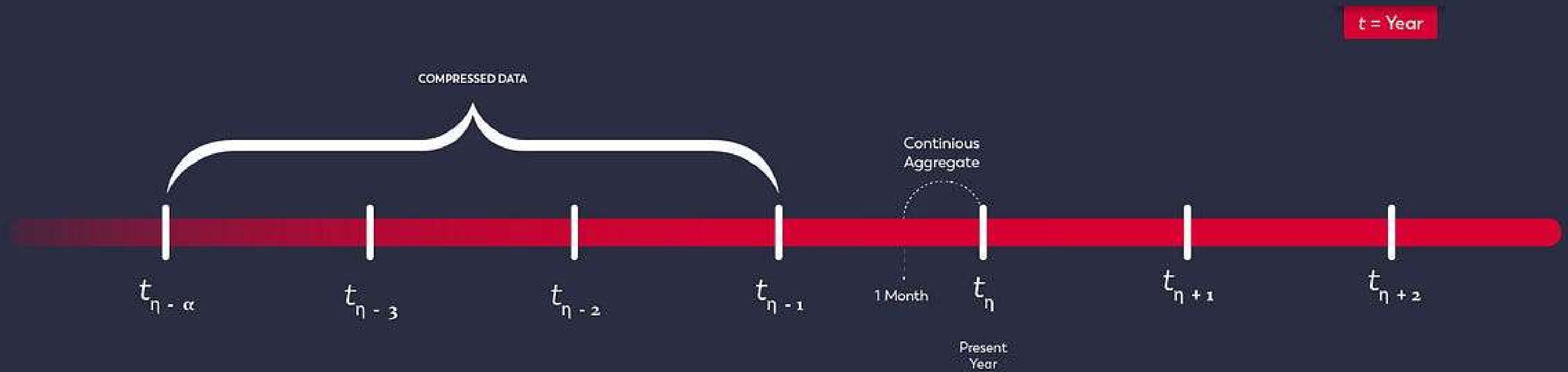
	transaction_year	A-Z coin_name	123 avg_open_value	123 avg_close_value	123 avg_value_high_low
165	2020-06-13 00:00:00.000	BTC/USDT	9,423.67	9,423.7	9,423.65
166	2020-06-14 00:00:00.000	BTC/USDT	9,404.97	9,404.89	9,404.83
167	2020-06-15 00:00:00.000	BTC/USDT	9,244.65	9,244.71	9,244.55
168	2020-06-16 00:00:00.000	BTC/USDT	9,490.08	9,490.15	9,490
169	2020-06-17 00:00:00.000	BTC/USDT	9,444.88	9,444.85	9,444.67
170	2020-06-18 00:00:00.000	BTC/USDT	9,408.79	9,408.73	9,408.65
171	2020-06-19 00:00:00.000	BTC/USDT	9,331.31	9,331.27	9,331.25
172	2020-06-20 00:00:00.000	BTC/USDT	9,303.64	9,303.68	9,303.66
173	2020-06-21 00:00:00.000	BTC/USDT	9,350.16	9,350.12	9,350.02
174	2020-06-22 00:00:00.000	BTC/USDT	9,480.76	9,481.03	9,480.96

TimescaleDB'de Compression ve Continuous Aggregate

```
ALTER TABLE coin_data SET (  
    timescaledb.compress,  
    timescaledb.compress_segmentby = 'symbol'  
);  
  
SELECT add_compression_policy('coin_data', INTERVAL '1 year');
```

```
CREATE MATERIALIZED VIEW tmp_schema.view_coin_data  
WITH (timescaledb.continuous) as  
select time_bucket('1 minute',transaction_time),transaction_time,symbol,  
open_value, high_value, low_value, close_value, volume_coin, volume_usdt,  
trade_count  
from tmp_schema.coin_data  
order by 1,2;  
  
SELECT add_continuous_aggregate_policy('tmp_schema.view_coin_data',  
    start_offset => interval '1 month', ---Verinin en güncel tarihinden kaç gün geriye gideceği tanımlanır.  
    end_offset => null ---Materialize edilecek son tarih belirlenir.  
    schedule_interval => INTERVAL '1 Day' ---Bu policy'in çalışma sıklığı belirlenir.
```

Summary



Airflow ile ETL Uygulamaları



Airflow bir iş akışı (workflow) yönetim sistemidir. Bu uygulama üzerinden Python görevlerinin takibi, yönetimi vb. işler kolayca yapılabilir.

DagsRunsTask Instances

Q Search Dags

Advanced SearchCtrl+K

AllFailedRunningSuccess

AllFilter by tag

105 Dags2

Sort by Latest Run Start Date...

01_A01_S0003_hisse_senedi_fiyat_bilgileri_gunluk

Schedule

30 00 * * *

Latest Run

2025-10-20, 05:03:04

✓

Next Run

2025-10-21, 03:30:00

🔌▶

01_A01_S0002_hisse_senedi_bilgileri_guncelleme

Schedule

30 00 * * *

Latest Run

2025-10-20, 05:00:26

✓

Next Run

2025-10-21, 03:30:00

🔌▶

01_A01_S0001_yts_staging

Schedule

30 00 * * *

Latest Run

2025-10-20, 03:30:00

✓

Next Run

2025-10-21, 03:30:00

🔌▶

99_Z99_S9999_log_clean

Schedule

0 15 1 * *

Latest Run

2025-10-01, 18:00:00

✓

Next Run

2025-11-01, 18:00:00

🔌▶

01_A01_S0004_twitter_staging

Schedule

0 0 * * 1

Latest Run

2025-06-15, 22:17:00

✗

Next Run

2025-06-16, 03:00:00

🔌▶

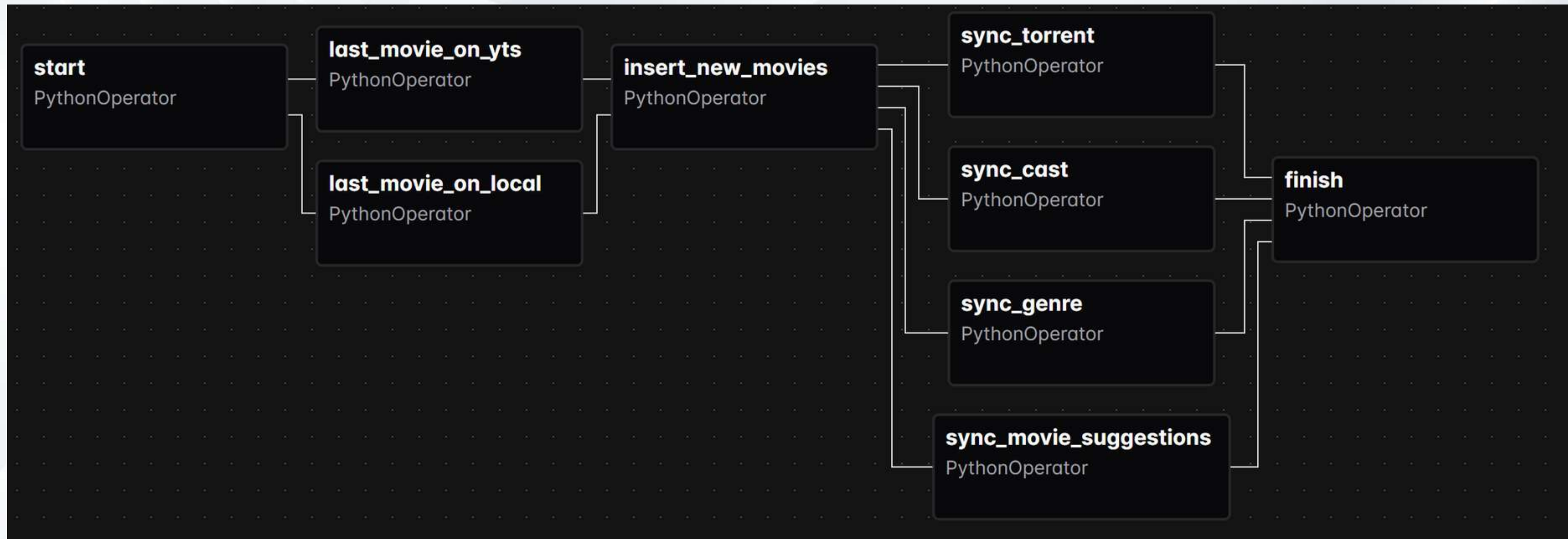
tutorial_taskflow_templatesexample

Schedule

Latest Run

Next Run

🔌▶



As is Aktarım

```
def update():
    ms = db.ms_con
    ts = db.ts_con

    ms_cursor = ms.cursor()
    ts_cursor = ts.cursor()

    yesterday = date.today() - timedelta(days=1)

    ts_cursor.execute(''' DELETE FROM fused_data.fused_data WHERE fused_date >= now()::date -1 ''')
    ts.commit()

    ms_cursor.execute(''' SELECT vSegID, vSegDir, fusedDate, fusedSpeed
                           FROM TDMS_Fused.dbo.FusedData_Archive
                           where fusedDate >= cast(cast(GETDATE() as date) as smalldatetime)-1 ''')
    execution = ms_cursor.fetchall()
    print("Data Executed...")

    ts_cursor = ts.cursor()
    cols = ('vseg_id', 'vseg_dir', 'fused_date', 'fused_speed')
    mgr = CopyManager(ts, 'fused_data.fused_data', cols)
    mgr.copy(execution)
    ts.commit()
    print("Data Imported...")

    ts_cursor.execute(''' DELETE FROM fused_data.fused_data WHERE fused_date >= now()::date ''')
    ts.commit()
```

Airflow v3 ile Gelen Yeni Özellikler

- Yeni arayüz.
- Git ile sürümleme
- Backfill

Superset ile Veri Görselleřtirme

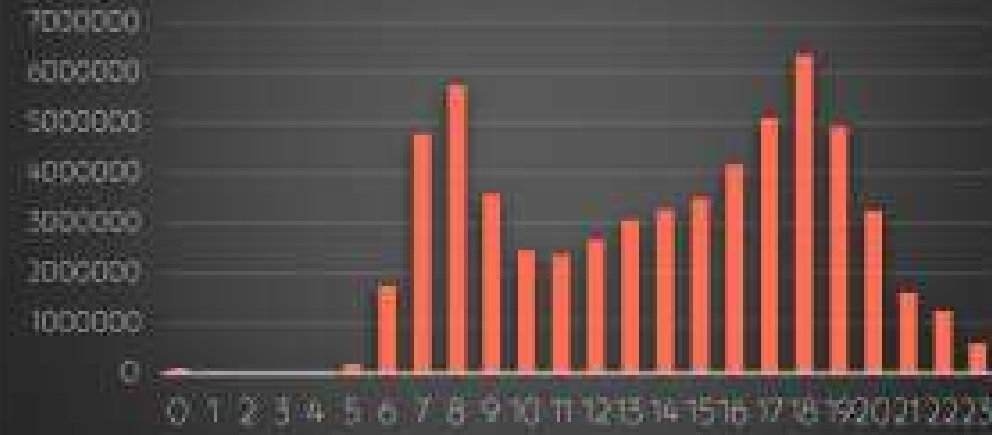


Superset, veriyi analiz etmemizi ve görselleřtirmemizi saęlayan açık kaynaklı bir Business Intelligence (BI) platformudur.

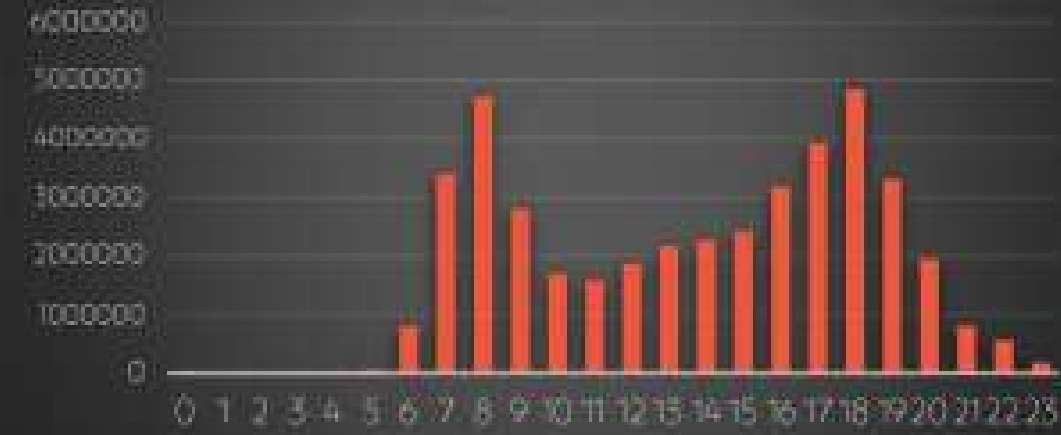




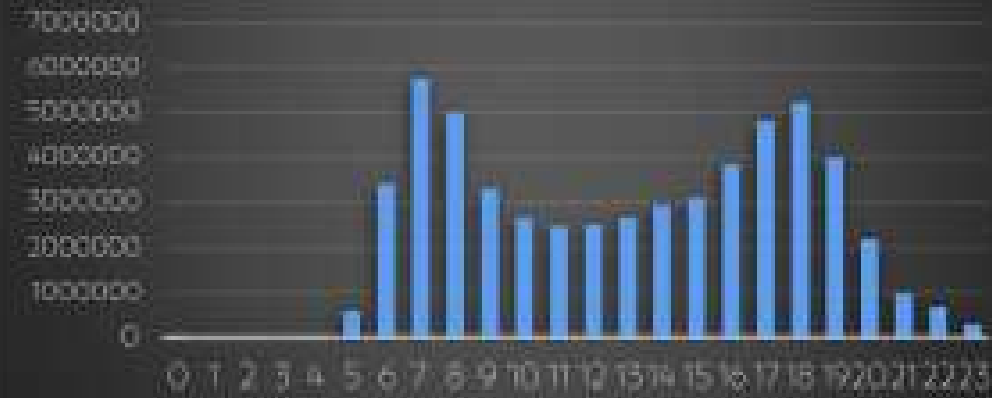
**Bekar olan Erkek Yolcuların
Saatlik Geçiş Sayıları**



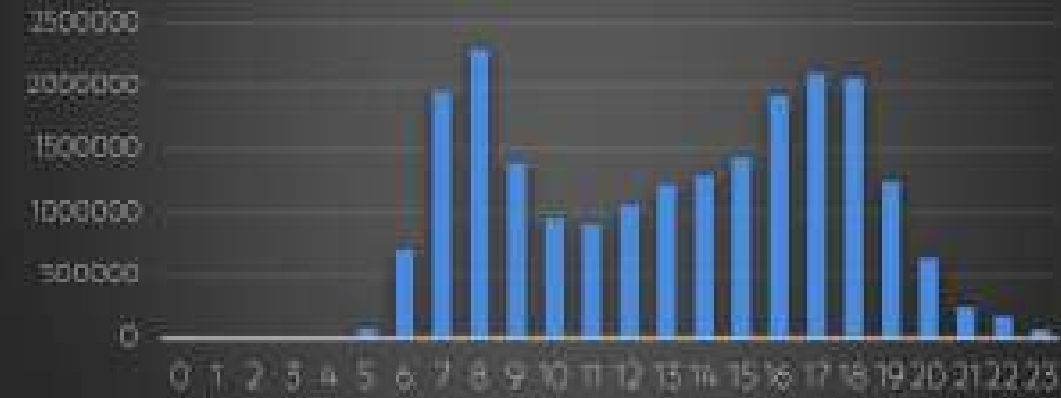
**Bekar Olan Kadın Yolcuların
Saatlik Geçiş Sayıları**



**Evli Olan Erkek Yolcuların
Saatlik Geçiş Sayıları**



**Evli Olan Kadınların Saatlik
Geçiş Sayıları**



Ortamlarda Satılacak Üç Analiz :)

1. Conjoint Analizi

Conjoint Analizi, insanların bir ürünü tercih ederken hangi özelliklere ne kadar değer verdiğini anlamamızı sağlar. Kısacası, “karar verirken hangi faktör en çok etkiliyor?” sorusuna yanıt verir. Pazarlama, ürün fiyatlandırma ve kamu hizmetlerinde vatandaş memnuniyetinin hangi unsurlardan beslendiğini ölçmek için kullanılır. Genellikle regresyon ya da logit modelleriyle tercih verileri analiz edilerek, kullanıcı davranışlarının ardındaki motivasyonlar ortaya çıkar. Yani veri bize yalnızca ne istediklerini değil, neden istediklerini söyler.

2. Gri İlişkisel Analiz

Gri İlişkisel Analiz ise belirsizlik altında alternatifleri değerlendirmeyi mümkün kılar. Tam bilgiye sahip olunmadığında bile, gözlemler arasındaki benzerlik derecelerini hesaplayarak en uygun seçeneği belirler. Özellikle enerji verimliliği, mühendislik, sürdürülebilirlik ve performans ölçümü alanlarında kullanılır. Verinin eksik, gürültülü veya normalize edilmemiş olduğu durumlarda bile güvenilir sonuçlar üretir. Klasik makine öğrenmesi büyük veriye ihtiyaç duyarken, gri analiz az veriyle bile anlam üretmeyi başarır.

Kısaca Gri İlişkisel Analiz elma ile armutu aynı kefedede analiz etmenize yarar

3. Veri Zarflama Analizi

Veri Zarflama Analizi, aynı kaynaklara sahip birimlerin etkinliğini ölçmeye yarayan güçlü bir optimizasyon yöntemidir. Çeşitli sektörlerde faaliyet gösteren şirketlerin şubelerinin, hastaneler, bankalar vb. kurumların verimliliğini karşılaştırmak için kullanılır. Lineer programlama temelli bu yaklaşım, çok boyutlu performans göstergelerini tek bir “etkinlik skoru”na indirger. Böylece sadece sayılar değil, kaynakların ne kadar etkili kullanıldığı da görünür hale gelir. Veri zarflama analizi, veriyi nicelikten çıkarıp verimlilik düzeyinde anlamlandırır.



İstanbul Üniversitesi Teknokent / Entertech
Üniversite Mh. Sarıgül Sk. No: 37/1 İç Kapı No: 28 Avcılar/İSTANBUL



www.thalesanalytics.io



info@thalesanalytics.io