



ניהול והנדסת

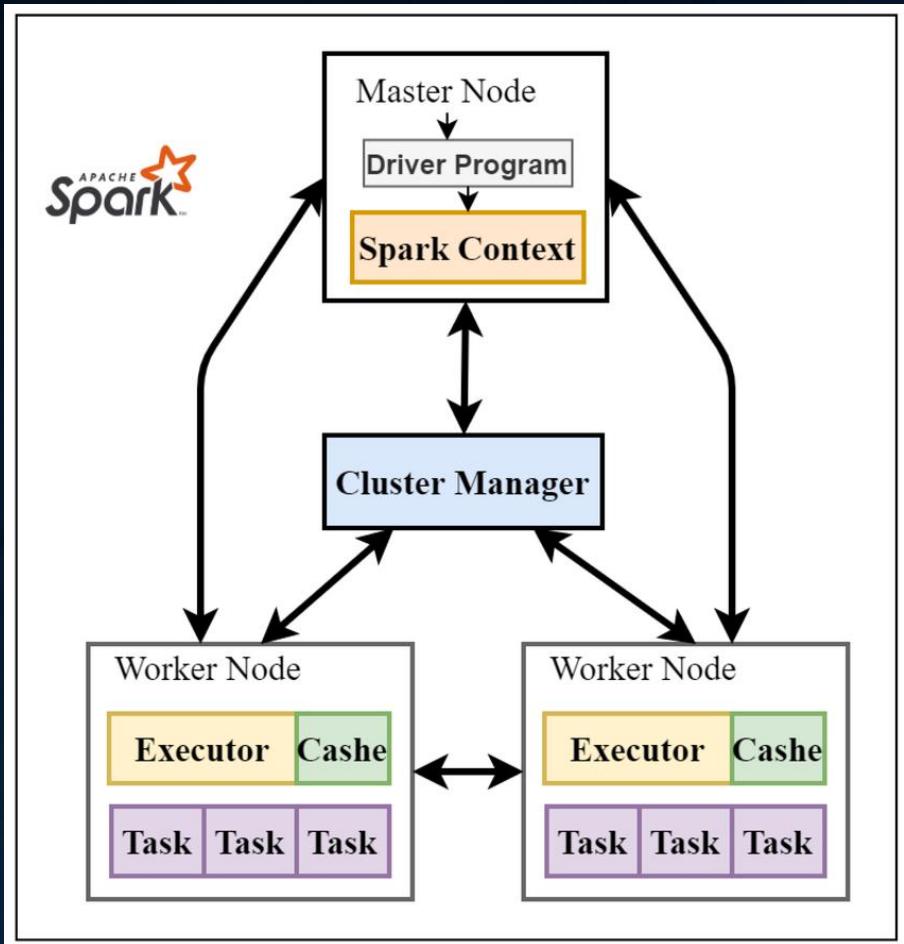
תשתיות נתונים עתק בענן

עבודה scikit-learn & PySpark

תשתיית Apache Spark / PySpark

- PySpark הוא מנוע אחד (Unified Engine) לעיבוד נתונים עתק (Big Data)
- ממשק PySpark: Python API המאפשר אינטגרציה בין עולם ה-Data Science לבין יכולות העיבוד המבוזרות של Spark (Pandas, Scikit-learn)
- עקרונות הליבה:
 - In-Memory Computing: טעינת הנתונים לזיכרון
 - Partitioning: חלוקת סט הנתונים (Dataset) לחתתי-משימות קטנות
 - Parallelism: ביצוע חישובים במקביל על גבי מספר שרתים (Worker Nodes) במקביל
 - Lazy Evaluation: המערכת לא מבצעת חישובים מיד, אלא בונה "תוכנית עבודה" (DAG) ומבצעת אותה רק כשמתבקשת תוצאה סופית – מה שໂחסן זמן ומשאבים יקרים

PySpark – ארכיטקטורת העבודה



- עיבוד מפוזר על גבי אשכול שרתים (Distributed Cluster)
- (Driver) Master Node: ה"מוח" המנהל מתכון את המשימות ומחולק אותן
- מקהה משאבי מחשב בענן (כמו YARN או Kubernetes) Cluster Manager
- (Executors) Worker Nodes: ה"פועלים" השרתים, הכוללת משאבי CPU וזיכרון, שמבצעים את החישוב בפועל על הנתונים
- Horizontal Scalability: יכולת התרחבות אופקית לטיפול בנפח נתונים (TB/PB) שאינם ניתנים לטעינה בזכרון של מחשב יחיד, באמצעות הוספה של Worker Nodes על פי הצורך
- השירותים עובדים במקביל ומאפשרים עיבוד מקבילי של נתונים בנפח עצום, ללא תלות ב מגבלות של מכונה בודדת

– יתרונות, אתגרים ו שימושים PySpark

- **אתגרים (Challenges):**
 - ניהול משאבים מורכב: Resource management
 - עקומת למידה תלולה: Learning curve
 - מרכיבות בחיבור מערכות: Integration effort
 - אבטחה והרשאות גישה: Security

- **יתרונות (Pros):**
 - גמישות והרחבה מהירה: Scalability
 - עיבוד מהיר בזיכרון (In-Memory): Speed
 - תמייה רחבה בשפות: Versatility
 - ניצול משאבים חכמים: Cost-efficiency

- **שימושים נפוצים (Use Cases):**
 - עיבוד נתונים מסיבי: Data Processing
 - בניית ETL Pipelines
 - אימון מודלים: Machine Learning
 - ניתוח בזמן אמת והציג ייזואלית בזמן אמת: Real time analytics

הבדל העיקרי: Pandas vs PySpark

מאפיין	Pandas (rangle Python)	PySpark (תשתית ענן)
נפח הנתונים	קטן עד בינוני - מוגבל ל- RAM של המחשב (לרוב עד כ 16GB – 32GB)	Big Data - יכול לעבוד כמוניות גדולות של נתונים (PB)
עיבוד	טורי על Single Node (מחשב אחד)	מקבילי וمبוזר – Cluster / Parallel
ביצועים	מהיר מאוד על נתונים קטנים בהיקף גדול	יעיל רק על נתונים בהיקף גדול
אופן חישוב	Eager (מיידי)	Lazy (עציל) – נדחה בהתאם לתוכנו

– הימוש בפרויקט זה – PySpark

1. שימוש ב- PySpark בפרויקט

- Feature Engineering - בניית Pipeline של טרנספורמציות להבנת הנתונים למידול
- Analysis - חקירת נתונים ועיבוד סטטיסטי
- MLlib - למידת מכונה - אימון מודלים וביצוע אופטימיזציה וקיצור זמני ריצה

2. למה ?Spark

- Industry Standard - עבודה בכלים המוכנבים בשוק המותאמים לסביבת עבודה אמיתי (Fault Tolerance, Production-ready)
- Scalability & Future-Proofing - תשתיית המוכנה לצמיחה בנפח הנתונים ללא שינוי קוד

הפרוייקט – הגדרת הבעה והיעד העסקי

- הנושא - ניהול הון אנושי בארגון (כולל יכולת גידלה ל Large-Scale)
- המטרה - בניית מודל המלצה לפיתורי עובדים על בסיס פרמטרים מוגדרים מראש
- הערך המוסף - מעבר מניהול מבוסס אינטואיציה לניהול מבוסס נתונים (Data-Driven Decision Making)

הפרויקט - השלבים בעיבוד הנתונים



*** כל השלבים בוצעו במחברת Jupyter תוך שימוש ב PySpark

הפרויקט – סקירה ומבנה הנתונים

- מקור הנתונים – קובץ נתוני עובדים (בשלב זה נטען 10,000 רשומות)
- מאפיינים עיקריים:
 - דמוגרפיה (גיל, מגדר, שנות לימוד)
 - תעסוקה (מחלקה, שכר, וותק)
 - הערכה (חוות דעת ושביעות רצון)

ID	Age	Gender	Department	Work_Mode	Current_Experience	Companies_Count	Education_Years	Total_Experience	Seniority	Monthly_Hours	Job_Satisfaction	Performance_Review	Salary
0	1	28.0	Male	Legal	Office	0.0	0.0	17.0	4.0	Junior	169.0	4.0	7.0 11500.0
1	2	50.0	Male	Support	Remote	35.0	0.0	15.0	35.0	Manager	191.0	5.0	10.0 31700.0
2	3	36.0	Female	Finance	Office	2.0	5.0	15.0	8.0	Mid	171.0	4.0	6.0 17800.0
3	4	34.0	Male	Finance	Hybrid	11.0	7.0	16.0	12.0	Mid	171.0	3.0	8.0 18800.0
4	5	29.0	Male	Legal	Hybrid	5.0	0.0	17.0	6.0	Mid	159.0	5.0	9.0 20500.0

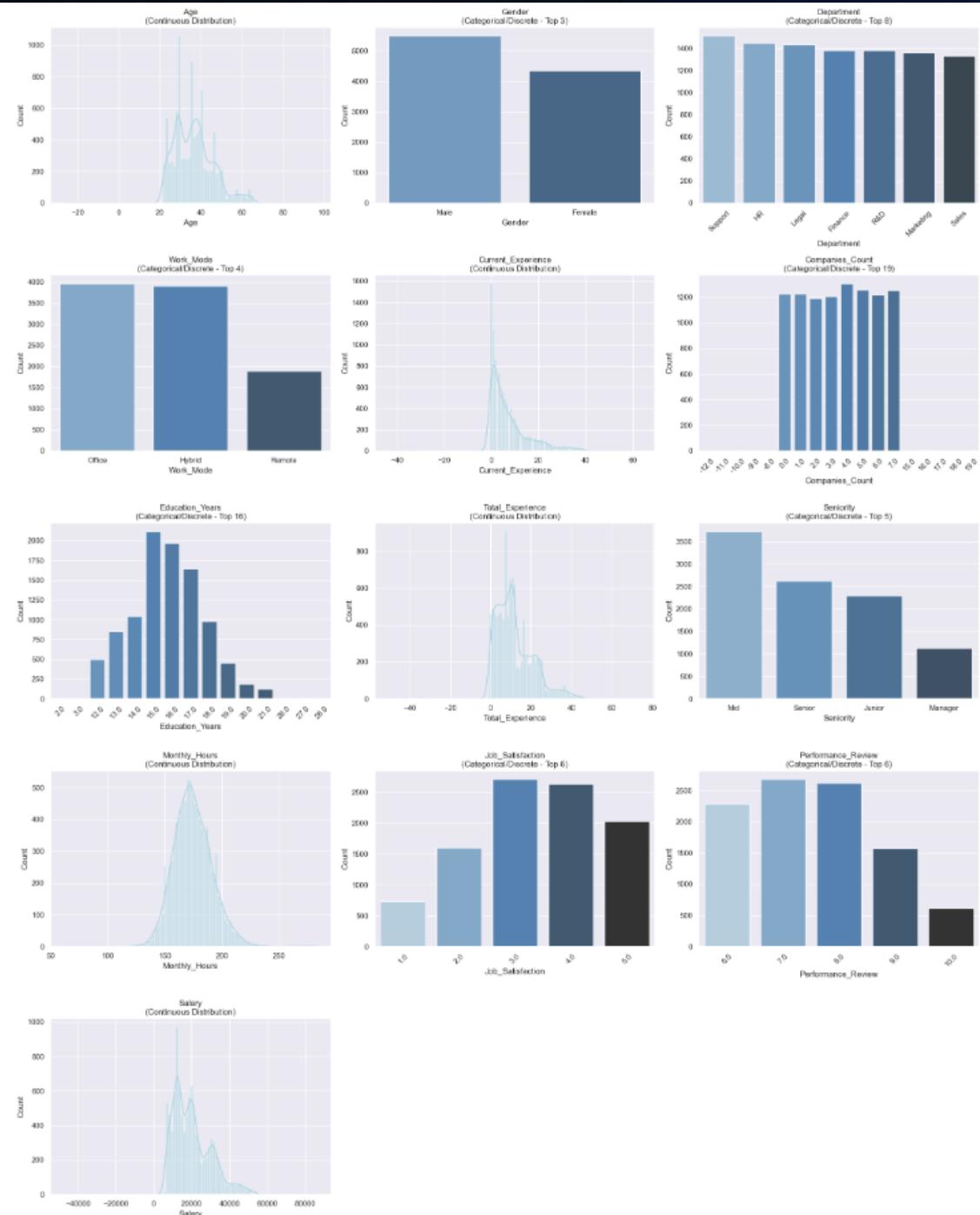
הפרויקט – בוחינה ראשונית של הנתונים (EDA)

Dataset Shape: (10000, 14) Total rows with at least one NULL: 2209 (22.09%)					
Column	Dtype	Non-Null	Null	Null %	Unique
ID	int	10000	0	0.0	10000
Age	double	9722	278	2.78	56
Gender	string	9845	155	1.55	2
Department	string	9856	144	1.44	7
Work_Mode	string	9753	247	2.47	3
Current_Experience	double	9816	184	1.84	60
Companies_Count	double	9895	105	1.05	18
Education_Years	double	9883	117	1.17	15
Total_Experience	double	9857	143	1.43	59
Seniority	string	9765	235	2.35	4
Monthly_Hours	double	9799	201	2.01	131
Job_Satisfaction	double	9695	305	3.05	5
Performance_Review	double	9773	227	2.27	5
Salary	double	9895	105	1.05	477

- בוחינת כל אחד מהמאפיינים
- שיעור היחסוי (במota ואחוז הערכים החסרים)
- ניתוח סטטיסטי

summary	count	mean	stddev	min	25%	50%	75%	max	mode	mode_freq%
ID	10000.0	5000.50	2886.90	1.0	2499.0	4999.0	7499.0	10000.0	8.0	0.01
Age	9722.0	36.47	9.49	-27.0	29.0	36.0	42.0	97.0	28.0	5.76
Current_Experience	9816.0	7.30	8.49	-44.0	1.0	4.0	10.0	64.0	0.0	15.76
Companies_Count	9895.0	3.53	2.37	-12.0	2.0	4.0	6.0	19.0	4.0	13.04
Education_Years	9883.0	15.76	1.98	2.0	15.0	16.0	17.0	28.0	15.0	21.17
Total_Experience	9857.0	11.75	9.37	-51.0	5.0	10.0	17.0	75.0	11.0	6.58
Monthly_Hours	9799.0	174.55	17.03	61.0	163.0	173.0	185.0	284.0	171.0	2.76
Job_Satisfaction	9695.0	3.37	1.20	1.0	3.0	3.0	4.0	5.0	3.0	27.09
Performance_Review	9773.0	7.54	1.19	6.0	7.0	7.0	8.0	10.0	7.0	26.80
Salary	9895.0	20159.81	10031.27	-47700.0	12400.0	18500.0	26400.0	86700.0	11900.0	0.90

הפרוייקט – בחינה ראשונית של הנתונים



- **Distribution:** התפלגות כל אחד מהמאפיינים
- **Data Consistency:** ידוא אחידות בפורמטים של משתנים קטגוריאליים

הפרוייקט –

ניקוי וטיפול הנתונים

- הגדרת כללים לוגיים (נתונים מספריים):

```
numeric_cols = [c for c in df.columns if isinstance(df.schema[c].dataType, NumericType) and c not in ['ID']]  
  
for c in numeric_cols:  
    dfc = dfc.withColumn(c, F.when(F.col(c) >= 0, F.col(c)).otherwise(F.lit(None)))
```

ניקוי נתונים – מחיקת ערכים קטנים מ 0 בכל שדה נומרי

```
dfc = dfc.withColumns({  
    "Age": F.when((F.col("Age") < 18) | (F.col("Age") > 70), F.lit(None))  
        .otherwise(F.col("Age"))})
```

ניקוי נתונים – מגבלת גיל העובד

```
dfc = dfc.withColumn("Education_Years",  
    F.when(F.col("Age") >= (F.col("Education_Years") + 6), F.col("Education_Years"))  
        .otherwise(F.lit(None)))
```

ניקוי נתונים – מגבלת מספר שנות הלימוד בהתאם לגיל

```
dfc = dfc.withColumn("Total_Experience",  
    F.when(F.col("Total_Experience") > (F.col("Age") - 18), F.lit(None))  
        .otherwise(F.col("Total_Experience")))
```

ניקוי נתונים – מגבלת הותק בתפקיד הנוכחי בהתאם לשנות הניסיון הכלול

```
dfc = dfc.withColumn("Current_Experience",  
    F.when(F.col("Current_Experience") > F.col("Total_Experience"), F.lit(None))  
        .otherwise(F.col("Current_Experience")))
```

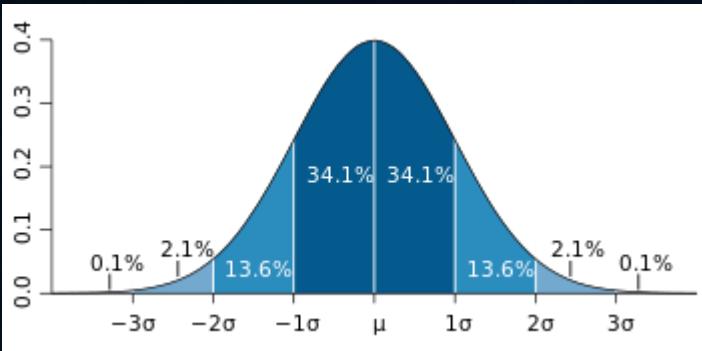
ניקוי נתונים – מגבלת כמות חברות בהתאם לשנות הניסיון הכלול

```
dfc = dfc.withColumn("Companies_Count",  
    F.when(F.col("Companies_Count") > F.col("Total_Experience"), F.lit(None))  
        .otherwise(F.col("Companies_Count")))
```

- ניקוי ערכים לא סבירים בرمת כל שדה

- ניקוי ערכים לא סבירים בהצלבה בין שדות

הפרוייקט – ניקוי וטיזוב הנתונים



- טיפול בחירגים נתונים מספריים (Outliers):
זיהוי והסרת ערכי קיצון שעלולים להטעות את המודל
באמצעות הגדרת טווח סטטיסטי

ניקוי נתונים – הסרת ערכי קיצון החורגים מ 3 סטיות תקן מהממוצע

```
selected_cols = ['Current_Experience', 'Companies_Count', 'Education_Years', 'Total_Experience', 'Monthly_Hours', 'Salary']

stats = dfc.select(
    *[F.mean(c).alias(c + '_mean') for c in numeric_cols],
    *[F.stddev(c).alias(c + '_std') for c in numeric_cols]
).first()

for c in selected_cols:
    mean_val = stats[c + '_mean']
    std_val = stats[c + '_std']

    if std_val is None: std_val = 0

    lower_bound = mean_val - (3 * std_val)
    upper_bound = mean_val + (3 * std_val)

    dfc = dfc.withColumn(c, F.when((F.col(c) >= lower_bound) & (F.col(c) <= upper_bound), F.col(c))
                           .otherwise(F.lit(None)))
```

** בסיום שלב זה מבוצעת
מחדר בchnerה של הנתונים

הפרויקט - טיפול בערכים קטגוריאליים

קידוד משתנים קטגוריאליים לערכים נומריים (Label Encoding)

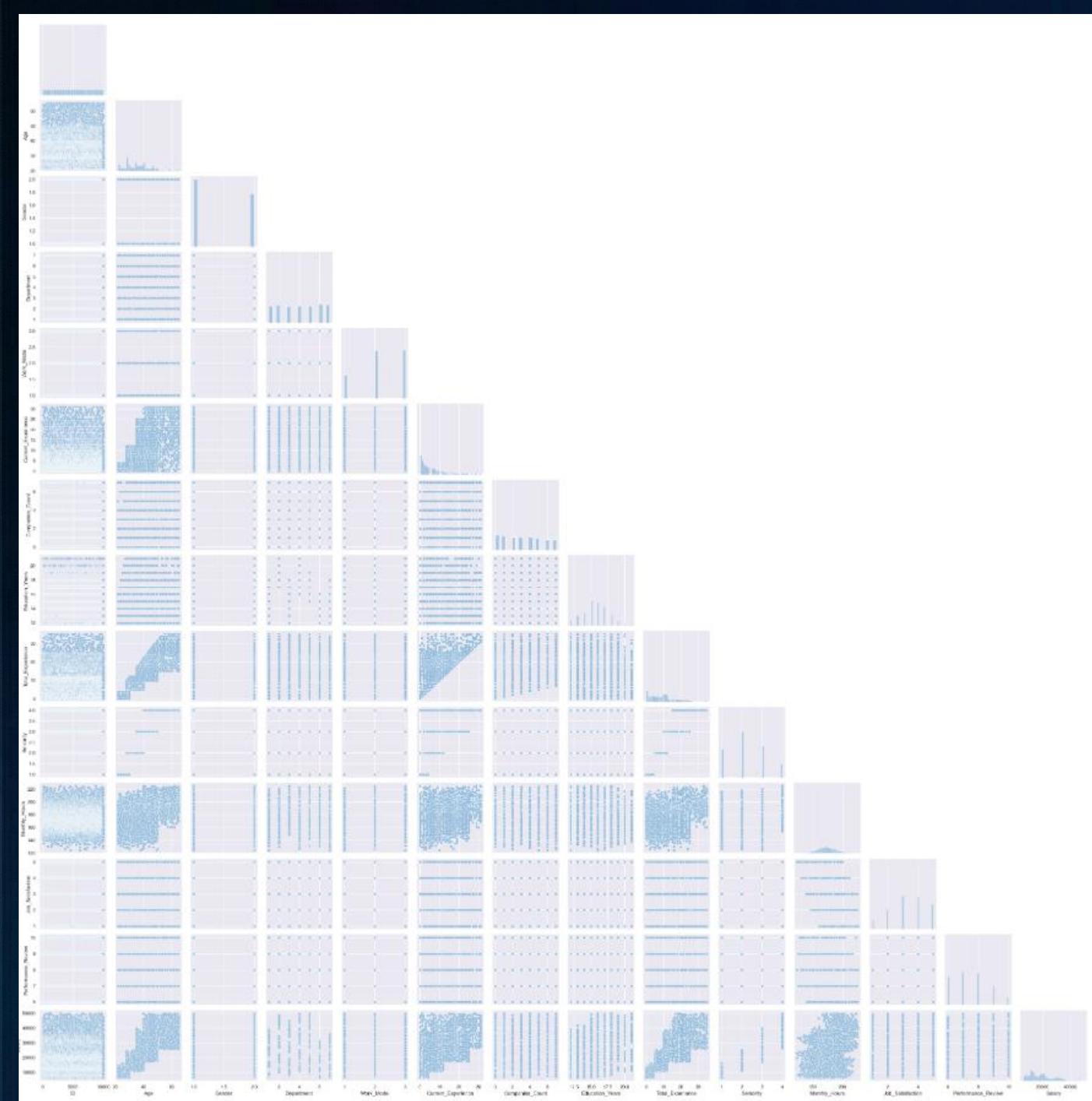
```
dfnum = dfc.withColumns({  
    "Gender": F.when(F.col("Gender") == "Male", 1)  
        .when(F.col("Gender") == "Female", 2)  
        .otherwise(None),  
  
    "Seniority": F.when(F.col("Seniority") == "Junior", 1)  
        .when(F.col("Seniority") == "Mid", 2)  
        .when(F.col("Seniority") == "Senior", 3)  
        .when(F.col("Seniority") == "Manager", 4)  
        .otherwise(None),  
  
    "Work_Mode": F.when(F.col("Work_Mode") == "Remote", 1)  
        .when(F.col("Work_Mode") == "Hybrid", 2)  
        .when(F.col("Work_Mode") == "Office", 3)  
        .otherwise(None),  
  
    "Department": F.when(F.col("Department") == "Marketing", 1)  
        .when(F.col("Department") == "Legal", 2)  
        .when(F.col("Department") == "Sales", 3)  
        .when(F.col("Department") == "R&D", 4)  
        .when(F.col("Department") == "Finance", 5)  
        .when(F.col("Department") == "Support", 6)  
        .when(F.col("Department") == "HR", 7)  
        .otherwise(None)  
})
```

:Label Encoding
המרת משתנים טקסטואליים
(כמו מחלוקת או מגדר)
לערכים נומריים המציגים אותן
באופן שהמודול יוכל לעבוד.

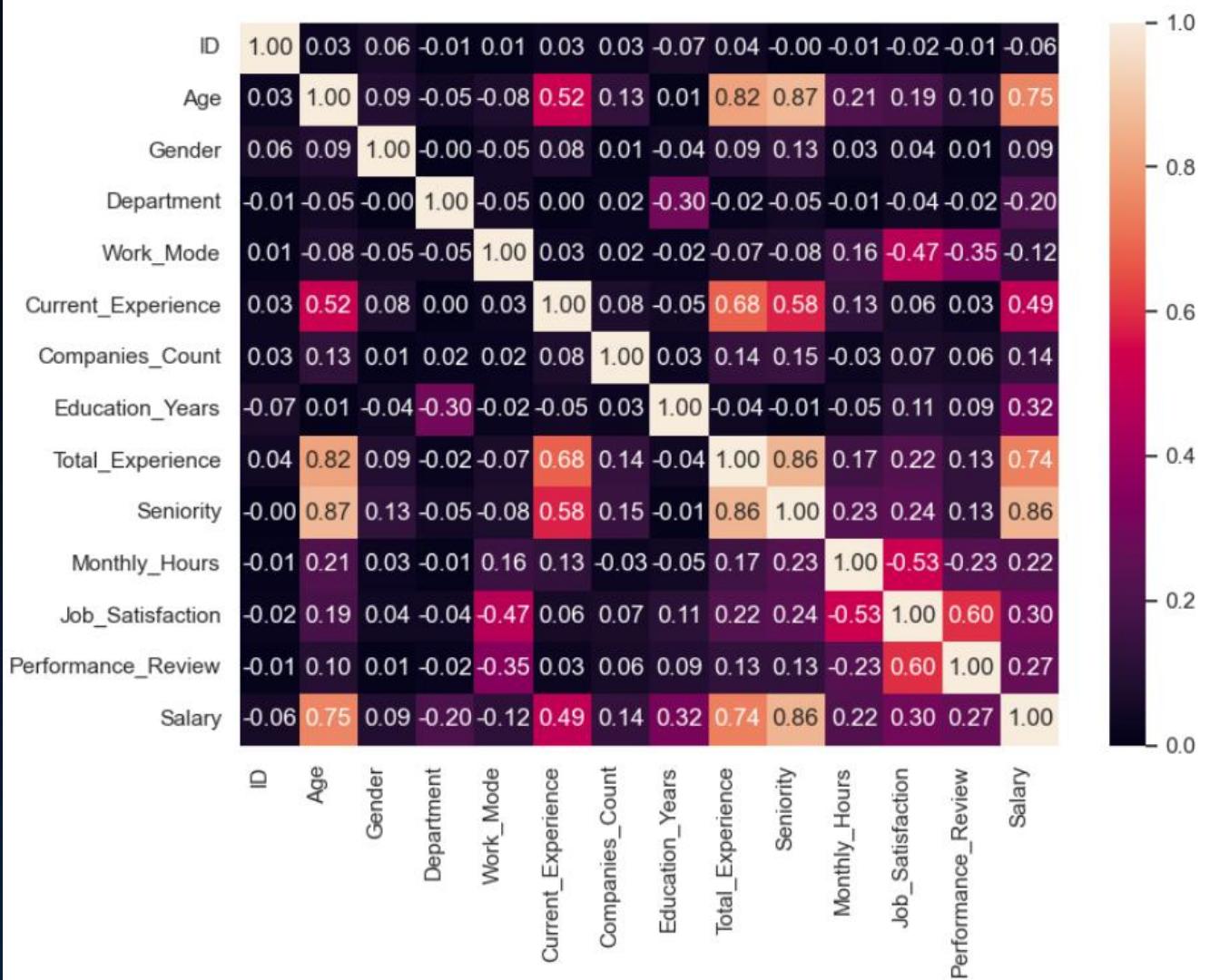
*** מצב זה הינו זמני לטובת הפעלת מודלים.
בסיום התהיליך יוחזרו הערכים המקוריים

הפרויקט – בחינת קורלציות

- לפני השלב הבא בטיפול בנתונים,
וכහשלה לבחינת הנתונים
הראשונית, בוחנתי את הקורלציות
בין העמודות (המשתנים) השונות
בDataset
- זיהוי ויזואלי של קורלציות:
השתמשתי בספרית Seaborn
ליצירת pairplot המפה ויזואלית
את ההצלבות בין כל זוג עמודות.
תצוגה זו מאפשרת לזהות במהט
מהיר מתחמים (Correlations),
דפוסים וקשריםلينיארים בין
המשתנים השונים



הפרוייקט – בחינת קורלציות



ניתוח הקורלציות:
לטובהניתו ניתוח נוסף של הקשרים בין
העמדות השונות,
השתמשתי במטריצת מתאימים
(Pearson Correlation Heatmap)
המאפשרת לזהות תלויות LINEAR בין
העמדות

הפרוייקט – השלמת ערכים חסרים

- על מנת שאוכל לבנות מודל חייזי מספיק חזק, נדרש תחילת להשלים את הנתונים החסרים, שגדלו משמעותית לאחר שלב הניקוי והטיבוב
- בהתאם לkorלציות שאיתרנו בשלב הקודם, בחרתי ב – 3 שיטות להשלמת הערכים החסרים:
 - עבור עמודות עם קורלציה גבוהה לעמודות אחרות או עמודות עם משמעות פחותה הסתמכתי על עמודות אלו להשלמת הערכים החסרים
 - עבור עמודת השכר, שהינה משמעותית, עשית שימוש במודל גרסיה
 - עבור עמודות מחלוקת ושות לימוד, עמודות משמעותיות עם קורלציה נמוכה לעמודות אחרות, עשית שימוש במודל קלסיפיקציה

הפרויקט – השלמת ערכים חסרים

דוגמה: שימוש לעמודות מודל העבודה

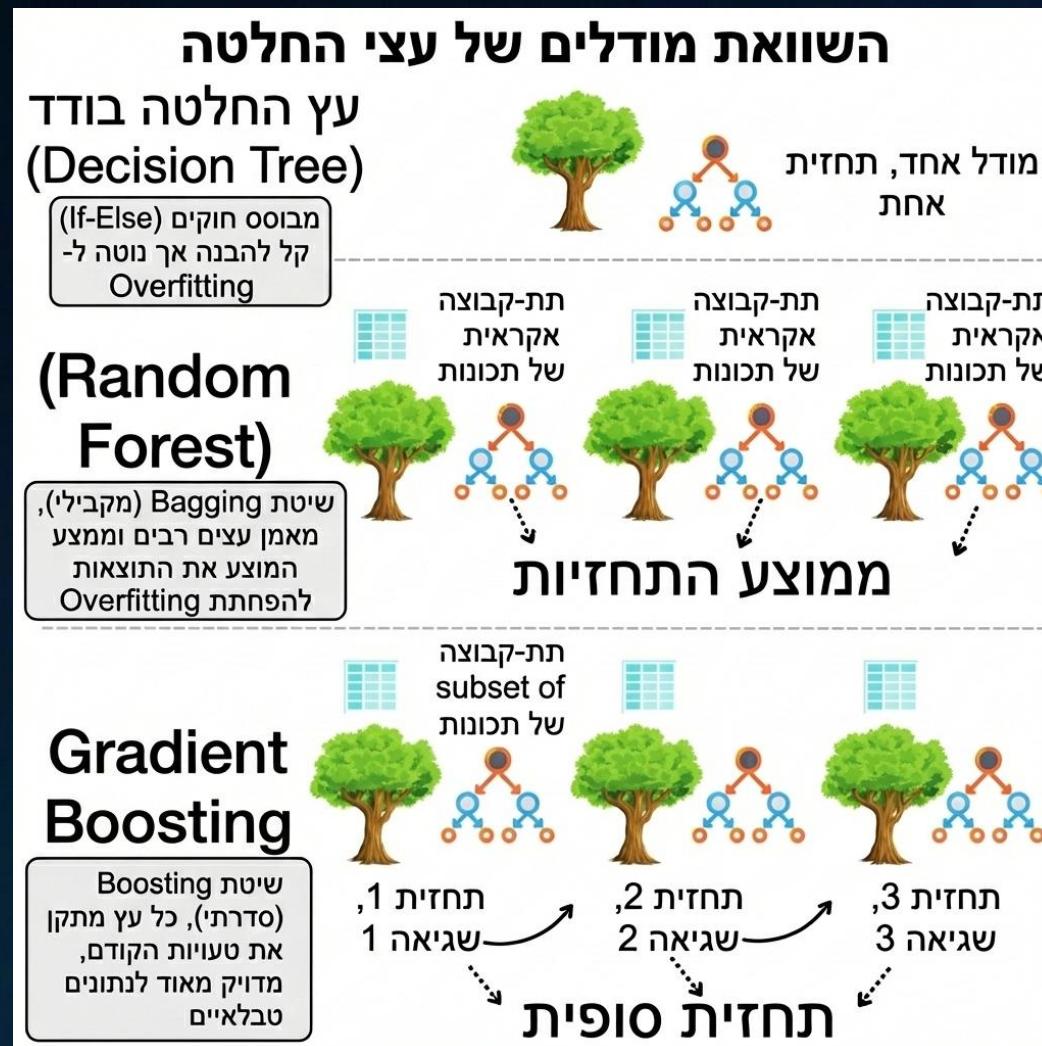
```
dfnum = fill_with_mode(dfnum, [
    ["Education_Years", "Seniority", "Job_Satisfaction", "Department"],
    ["Education_Years", "Seniority", "Department"],
    ["Education_Years", "Department"],
    ["Education_Years"],
    ["Seniority"],
    ["Job_Satisfaction"],
    ["Department"]
], "Work_Mode")

--- Starting Pyramid for Work_Mode ---
Level 1: Current NULL count: 247
Level 1: Processing group ['Education_Years', 'Seniority', 'Job_Satisfaction', 'Department']
Level 2: Current NULL count: 24
Level 2: Processing group ['Education_Years', 'Seniority', 'Department']
Level 3: Current NULL count: 18
Level 3: Processing group ['Education_Years', 'Department']
Level 4: Current NULL count: 15
Level 4: Processing group ['Education_Years']
Level 5: Current NULL count: 11
Level 5: Processing group ['Seniority']
Level 6: Current NULL count: 0
Work_Mode is already full. Skipping remaining levels.
--- Finished Work_Mode: 0 NULLs remaining ---
```

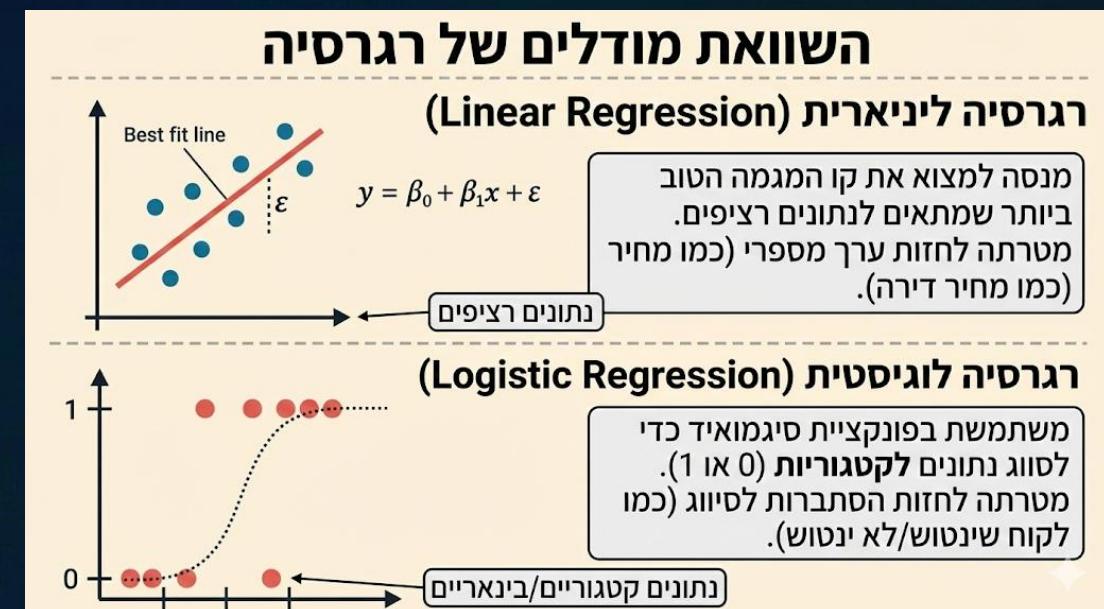
שיטת 1 - שימוש בקורסיות

עבור עמודות עם קורסיה גבוהה
לעמודות אחרות או עמודות עם
משמעות פחותה
(גיל, מגדר, מודל עבודה, וותק בחברה
נכחית, מספר חברות, וותק כולל,
בכירות, שעות עבודה חודשיות,
шибיאות רצון, חוות דעת) הסטמכתו
על עמודות נוספות עם קורסיה חזקה
לעמודה המבוקשת,
ובסתמך על עמודות אלה חושב
השכיח לכל קבוצה ונעשה בו שימוש
השלמת הערכים החסרים

הפרוייקט – השלמת ערכי חסרים

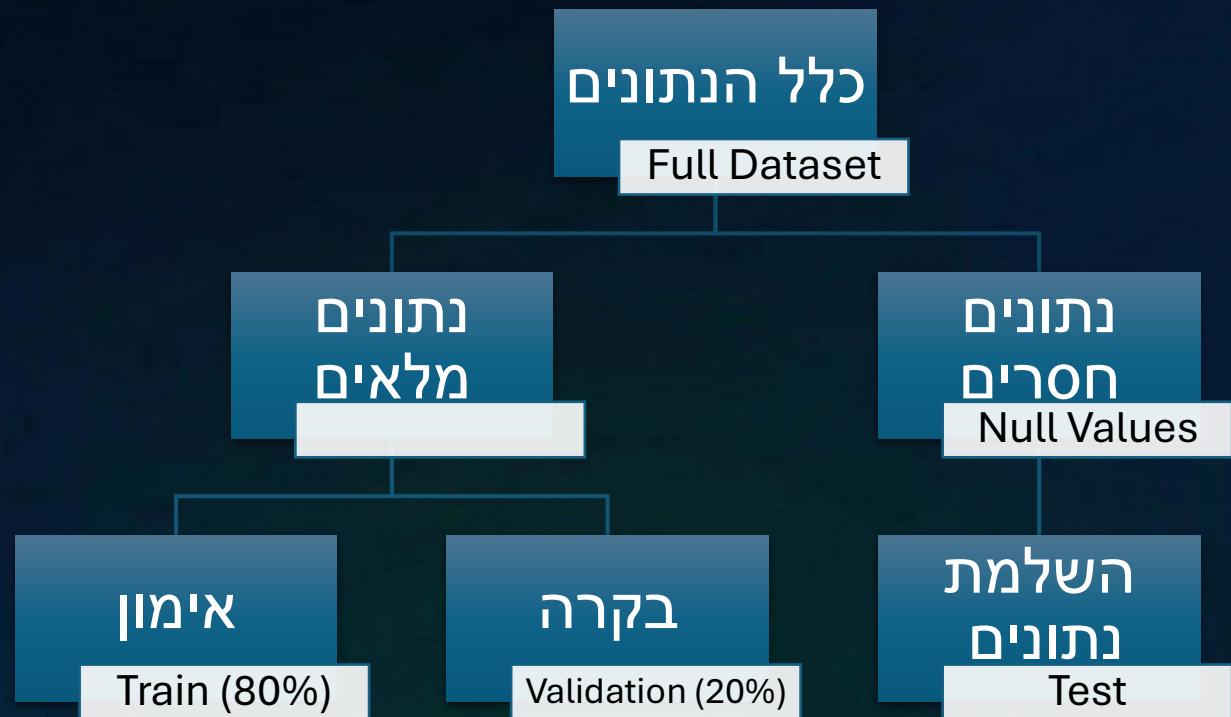


הסבר על מודלים של למידה מונחית (Supervised Learning) יש צורך לבחור את המודל המתאים ביותר לצורכי הספציפי



הפרויקט – השלמת ערכים חסרים

חלוקת הנתונים לטובת המודל



הפרויקט – השלמת ערכים חסרים

שיטת 2 – מודל גראסיה (עבור עמודות שבר)

הרצתי את הנתונים על 4 המודלים בסיום ההרצה קיבלתי את 3 הפרמטרים הבאים על כל אחד מהמודלים:

	Model	R2	RMSE	MAE
3	Gradient Boosting	0.986139	1109.065469	641.356324
1	Decision Tree	0.966760	1717.460940	1122.301628
2	Random Forest	0.941887	2270.880568	1567.210097
0	Linear Regression	0.889611	3129.829979	2449.469371

- R2 (R-Squared): מדד המציג איזה אחוז מהשונות (Variance) בתוצאות המודל מצליח להסביר (שוואפים ל-1).
- RMSE (Root Mean Squared Error): מדד שגיאה הרגילה במיוחד לסטיות גדולות ו"מעונייש" עליו, ולכן עדיף בשחזרה להימנע מטעויות קיצונית.
- MAE (Mean Absolute Error): השגיאה הממוצעת המוחלטת – מציג את הסטייה הממוצעת של המודל ביחידות המקוריות, בצורה האינטואיטיבית ביותר.

בחרתי במודל Gradient Boosting בגלל שבכל אחד מהפרמטרים הוא הכי מתאים באופן מובהק



הפרוייקט – השלמת ערבים חסרים

שיטת 3 – מודל קלאסיפיקציה Confusion Matrix (מטריצת הבלבול):

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP) Type II Error	False Negative (FN) Recall (Sensitivity) $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN) Specificity $\frac{TN}{(TN + FP)}$
	Precision	Negative Predictive Value (NPV) $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$
	F1-Score		$2 * \frac{Precision * Recall}{Precision + Recall}$

שיטת 3 – מודל קלאסיפיקציה

עבור חיזוי ערך קטגוריאלי, משתמש במודל קלאסיפיקציה – מודל למידת מכונה שתפקידו לחזות לאיזו קטgorיה (מחלקה) שייר נתון חדש על סמך דפוסים שלמד מנתוני ה עבר.

הפרמטרים לבחינת כל אחד מהמודלים:

- **Accuracy (דיק בילי)**: אחוז התוצאות הנכונות מתוך סך כל התוצאות – מدد טוב בשנתונים מאוזנים, אך מטעה בשיש חוסר איזון (Imbalanced Data).
- **Precision (דיק חיובים)**: מתוך כל המקרים שהמודל חזה כ"חיוביים", כמה באמת היו כאלה (מדד ה"אמינות" – חשוב כדי להימנע מ"ازעקות שוא").
- **Recall (רגישות)**: מתוך כל המקרים שהם באמת "חיוביים" במצבות, כמה המודל הצליח לגלוות (מדד ה"כיסוי" – חשוב בשאSOR לפפס מקרים).

- **F1 Score**: המוצע ההרמוני המשלב בין Precision ל-Recall למספר אחד, המציג את האיזון ביניהם (מדד הקובע במקרים של נתונים לא מאוזנים).

הפרוייקט – השלמת ערבים חסרים

שיטת 3 – קלסיפיקציה

הרצתי את הנתונים באמצעות 3 המודלים (Logistic Regression, Decision Tree, Random Forest) עבור עמודות מחלוקת ושנות למידה בסיסים הרצות קיבלו את הפרמטרים הבאים:

--- MODEL: Random Forest ---									
	precision	recall	f1-score	support					
12.0	0.00	0.00	0.00	90					
13.0	0.32	0.26	0.29	151					
14.0	0.26	0.18	0.21	201					
15.0	0.35	0.66	0.45	389					
16.0	0.41	0.52	0.46	359					
17.0	0.35	0.37	0.36	269					
18.0	0.00	0.00	0.00	154					
19.0	0.00	0.00	0.00	67					
20.0	0.00	0.00	0.00	40					
21.0	0.00	0.00	0.00	25					
accuracy			0.35	1745					
macro avg	0.17	0.20	0.18	1745					
weighted avg	0.27	0.35	0.30	1745					
Confusion Matrix:									
[[0 26 25 36 2 1 0 0 0 0]	[0 40 38 73 0 0 0 0 0 0]	[0 40 36 122 2 1 0 0 0 0]	[0 17 34 255 80 3 0 0 0 0]	[3 2 6 145 188 15 0 0 0 0]	[0 0 0 67 102 100 0 0 0 0]	[0 0 0 32 45 77 0 0 0 0]	[0 0 0 1 24 42 0 0 0 0]	[0 0 0 2 6 32 0 0 0 0]	[0 0 0 0 10 15 0 0 0 0]]

שנות למידה –
עבור אפ אחד
מהמודלים לא
התקבלה תוצאה
מספקת ולכון נחזר
להשלים את
הנתונים באמצעות
השיטה הראשונה

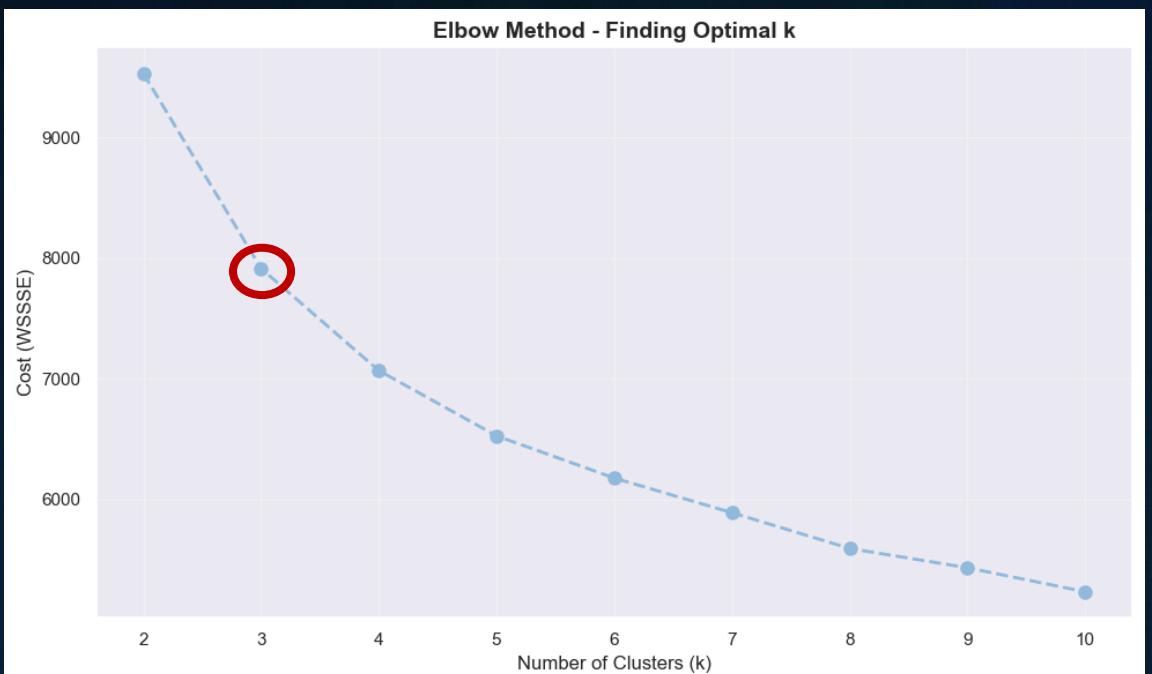
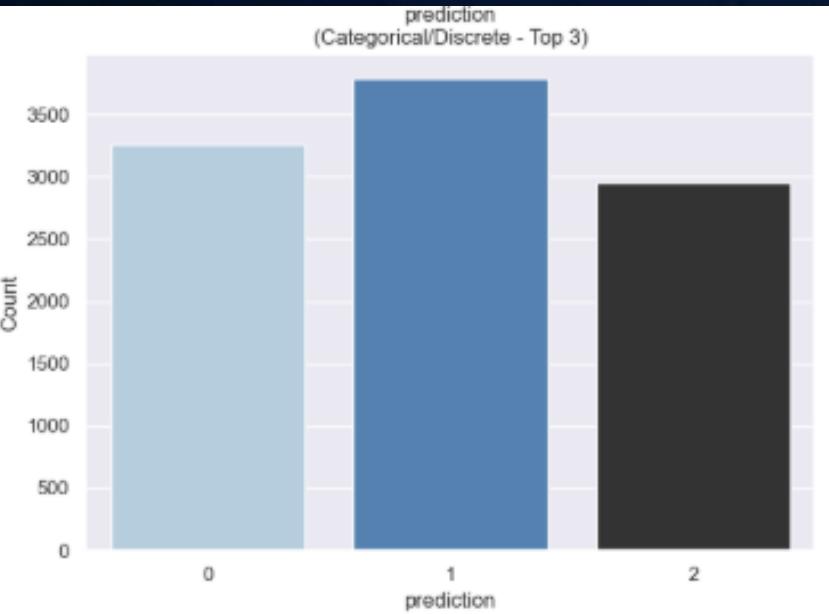
--- MODEL: Logistic Regression ---						
	precision	recall	f1-score	support		
1	0.57	0.58	0.57	224		
2	0.67	0.74	0.71	239		
3	0.65	0.63	0.64	230		
4	0.74	0.68	0.71	235		
5	0.65	0.64	0.64	252		
6	0.71	0.76	0.74	296		
7	0.56	0.52	0.54	269		
accuracy			0.65	1745		
macro avg	0.65	0.65	0.65	1745		
weighted avg	0.65	0.65	0.65	1745		
Confusion Matrix:						
[[129 5 30 1 4 13 42]	[4 178 3 38 16 0 0]	[38 5 144 4 30 4 5]	[0 37 0 160 38 0 0]	[3 39 35 14 161 0 0]	[7 0 2 0 0 226 61]	[46 0 9 0 0 74 140]]

מחלקה –
התוצאה הטובה
bijouter התקבלה
עבור מודל
Logistic
Regression

הפרוייקט – ניתוח הנתונים

למידה לא מוחשית וניתוח אשכלהות
(Unsupervised Learning)

	k	cost	slope	percent
0	2	9533.271	NaN	NaN
1	3	7915.676	-1617.595	-16.968
2	4	7068.707	-846.969	-10.700
3	5	6527.128	-541.579	-7.662
4	6	6177.455	-349.672	-5.357
5	7	5889.850	-287.605	-4.656
6	8	5593.519	-296.331	-5.031
7	9	5433.371	-160.149	-2.863
8	10	5234.343	-199.028	-3.663

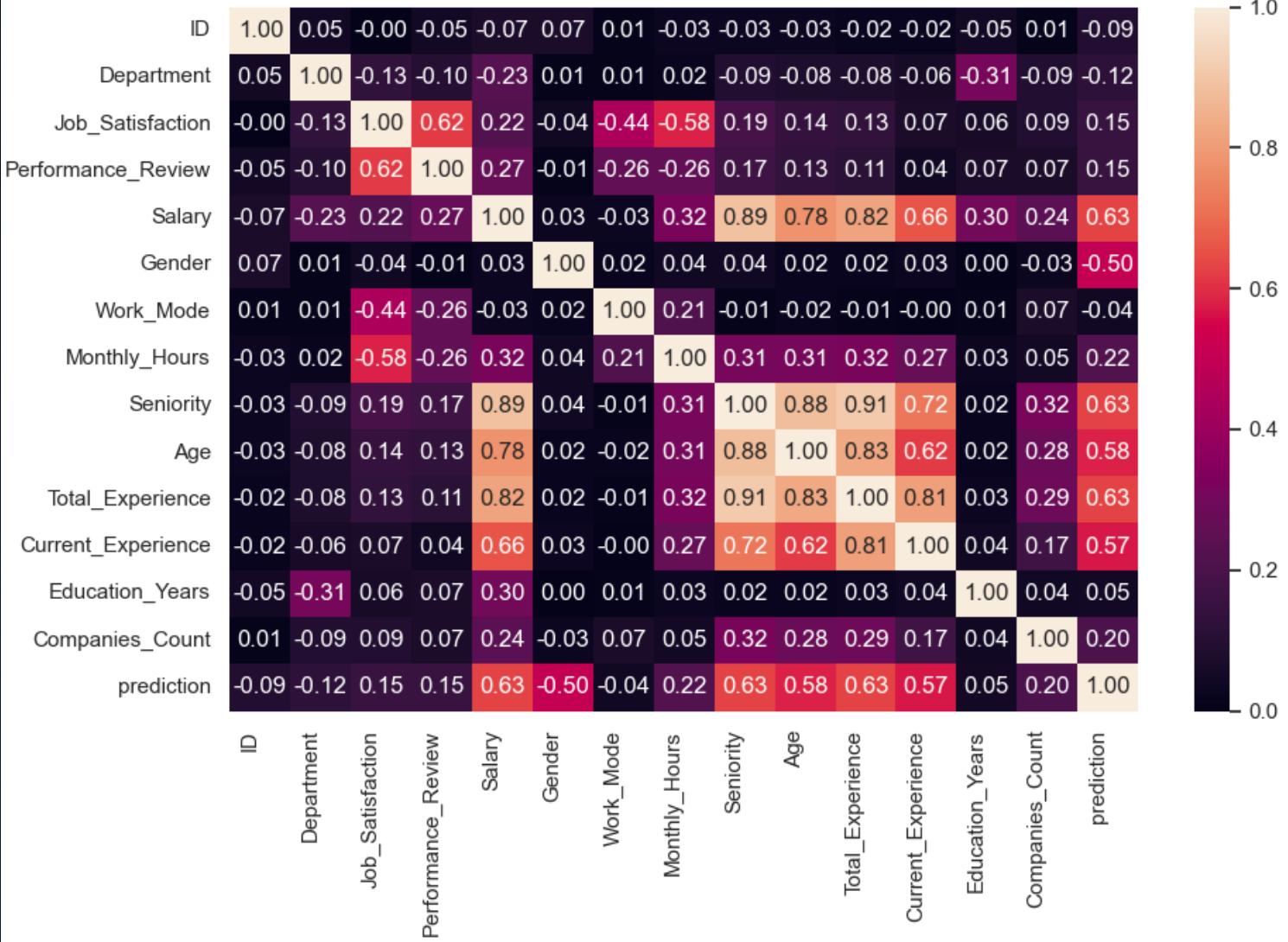


- אלגוריתם המחלק את הנתונים ל K קבוצות נפרדות על בסיס דמיון וקרבה מרחבית.
- Method Elbow (שיטת המפרק)
כלי גרפי לבחירת מספר האשכלהות האופטימלי ע"י זיהוי נקודת האיזון בין דיקן ליעילות.
- Silhouette Score
ציון איזות (בין 1- ל-1) הבודק עד כמה האשכלהות מופרדים היטב זה מזה ומוליכים בתוכם.

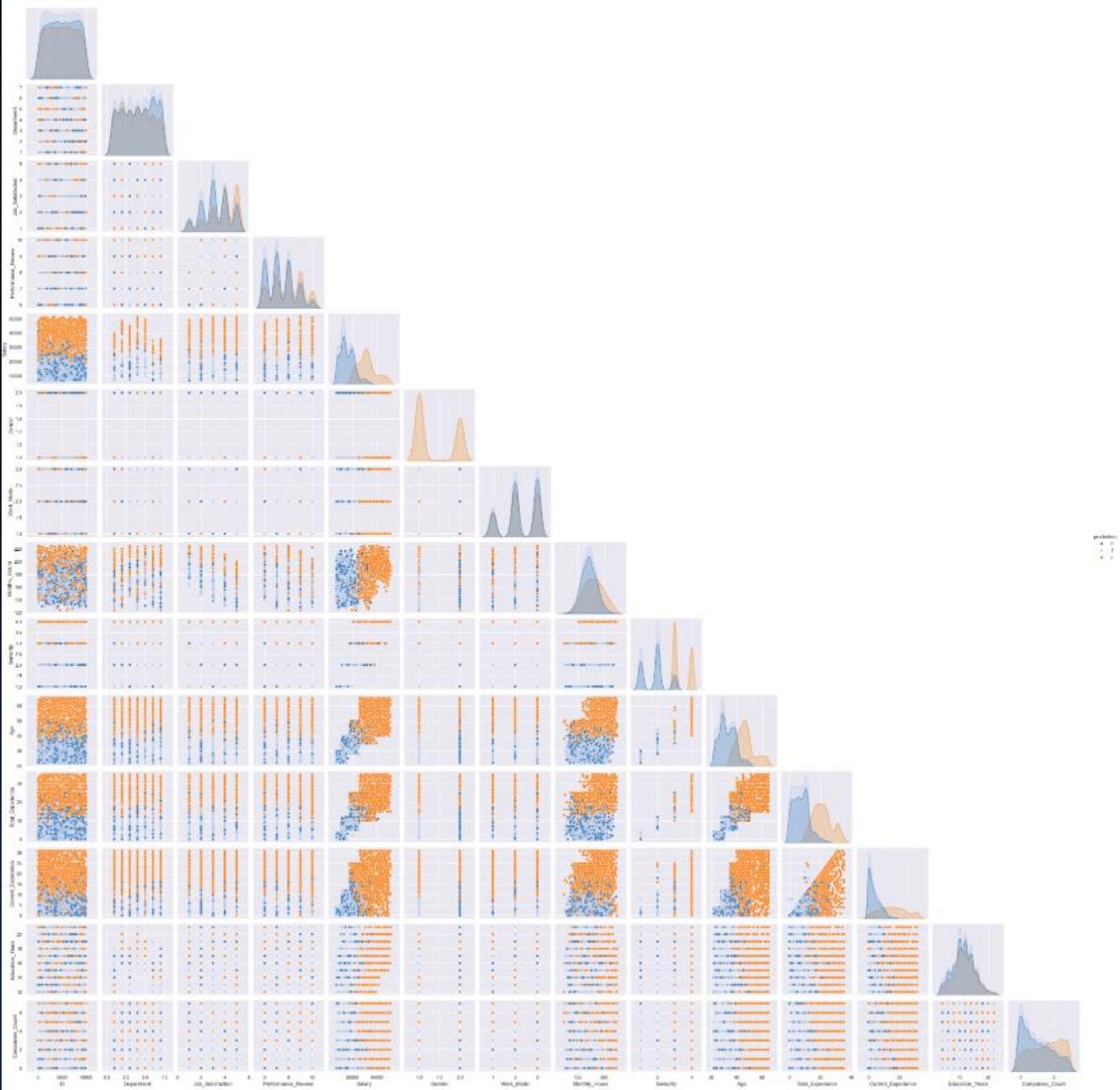
הפרוייקט –

ניתוח הנתונים

- בגרף ה-Elbow Method ניתן לראות בבירור שנקודת ה"ברך" (השבירה החדה ביותר) מתרחשת ב-K=3
- בנוסף ניתן לראות קשרים מאד חזקים עם רוב העמודות האחרות ביחס לעמודת החיזוי.
- הקשרים החזקים ביותר קשורים לוותק כולל, בכירות, משכורת, גיל, וותק בחברה וכחית ומדרג

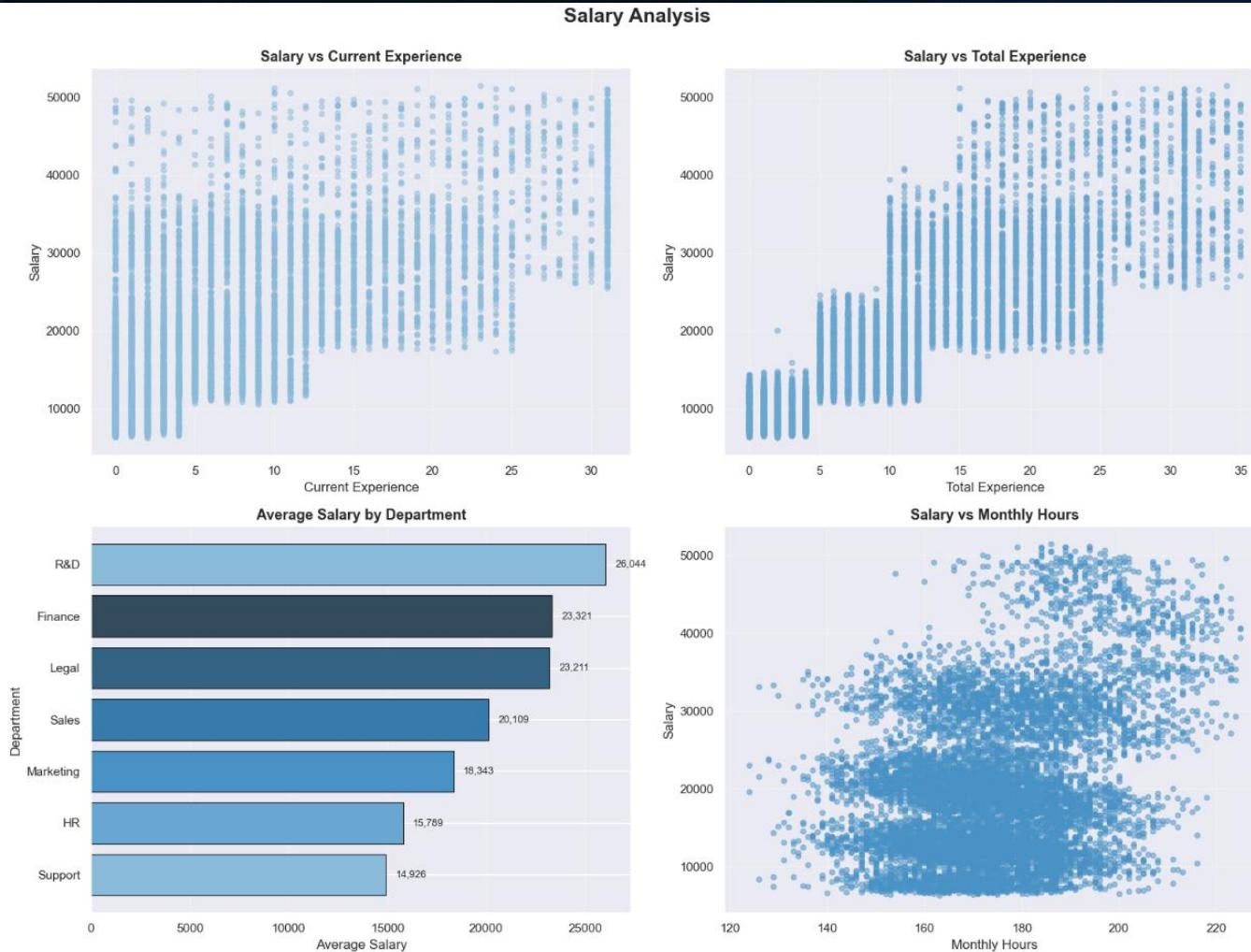


הפרויקט – ניתוח הנתונים



- אשכול 0
נשים בתחילת ואמצע הקריירה
(דרג בינוניים).
- אשכול 1
גברים בתחילת ואמצע הקריירה
(דרג בינוניים).
- אשכול 2
ה בכירים והנהלה – העובדים המנוסים
ומתוגמלים ביותר (מכל המגדלים)

הפרויקט – ניתוח הנתונים



מתאמים תכונות (Salary Analysis)

- **Total Experience** (וותק כללי): המבאה חזק ביותר. ניתן לראות מבנה "מדרגות" המייצג את המעבר בין דרגות בכירות (Junior/Mid/Senior).
- **Current Experience** (וותק בחברה הנוכחי): מציג מתאם חיובי לשבר, אך עם "רעש" (Variance) גבוהה יותר בהשוואה לוותק הכללי.
- **Department** (מחלקה): משתנה מפיד משמעותי. פער שבר ברורים בין מחלקות הליבה (R&D, Finance) למחלקות התפעול (Support).
- **Monthly Hours** (שעות עבודה): פיזור רחב ("ענן נקודות") וקורלציה נמוכה. היקף השעות אינו מהו אינדיקטור אמיתי לגובה השכר.

– Business Use Case

קבלת החלטות מבוססות נתונים בנוגע לכוח אדם

- המטרה - בניית מודל להמלצה על פיטורי עובדים על בסיס פרמטרים מוגדרים מראש
היעד הינו בניית אתר של 15% מכלל כוח האדם
- לוגיקת הדירוג - מודל מבוסס MLM המעניק לכל עובד ציון כדאיות פיטורים (10-1),
באשר ציון 10 מייצג המלצה גבוהה לפיטורים.
הציון משקלל את הפרמטרים הבאים:
 - קריטריונים לשימור
 - חוות דעת - ציון הערכת עובד 8 ומעלה
 - שנות לימוד – תואר ראשון לפחות (15+ שנות לימוד ומעלה)
 - שביעות רצון - דירוג 4 ומעלה
 - קריטריונים לפיטורים
- גיל - עובדים מעל גיל 60 או מתחת לגיל 24
- בכירות - התמקדות בדרגת-Junior
- אילוץ קשיich - תהלייך הצטנום חייב לשמר על התפלגות זהה של כוח האדם בין המחלקות
כדי לא לפגוע בפעולות הליבה של מחלקה ספציפית

– Business Use Case קבלת החלטות מבוססות נתוניים בנוגע לכוח אדם



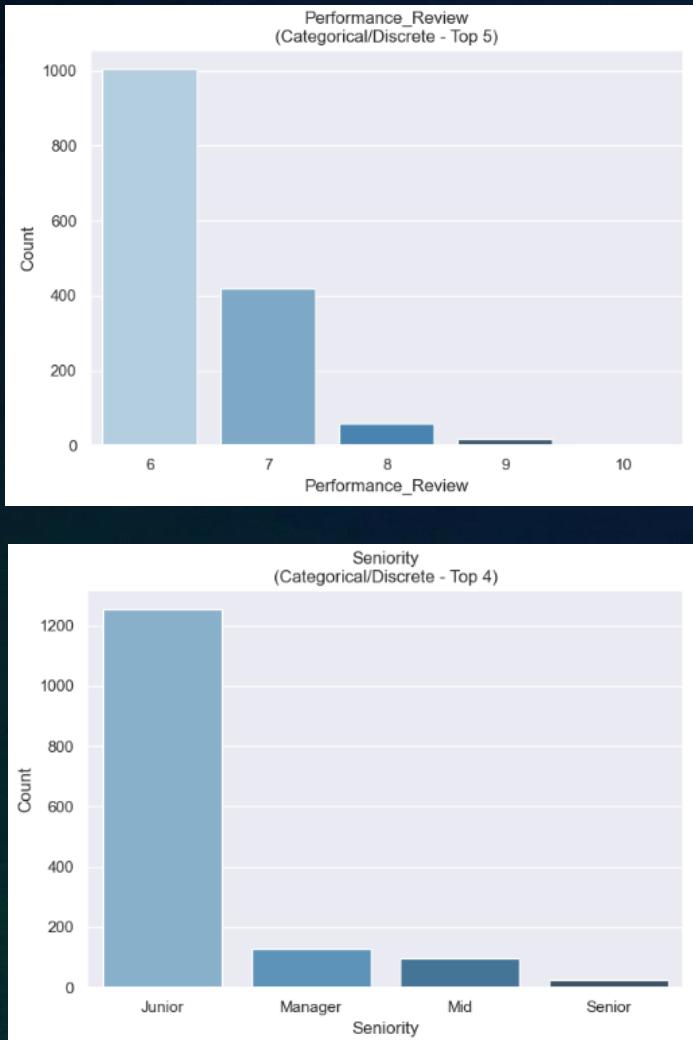
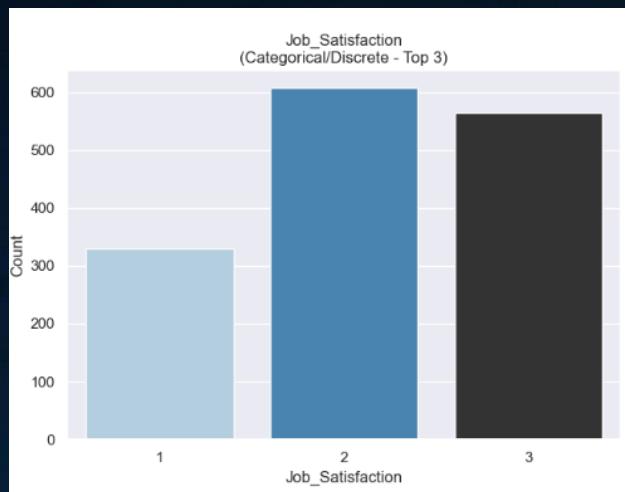
בגרף ניתן לראות את התפלגות הערך של ציון כדיות הפיטורים

על מנת לעמוד ביעד של 15% מכלל האוכלוסייה, נבחרו ערכי ציונים החל מ 5.85

זאת לעומת שישה כלו עם ציון גובהה יותר (עד 7.58), בשל האילוץ של שמירה על התפלגות כ"א בין המחלקות

– Business Use Case

קבלת החלטות מבוססות נתונים בנוגע לכוח אדם

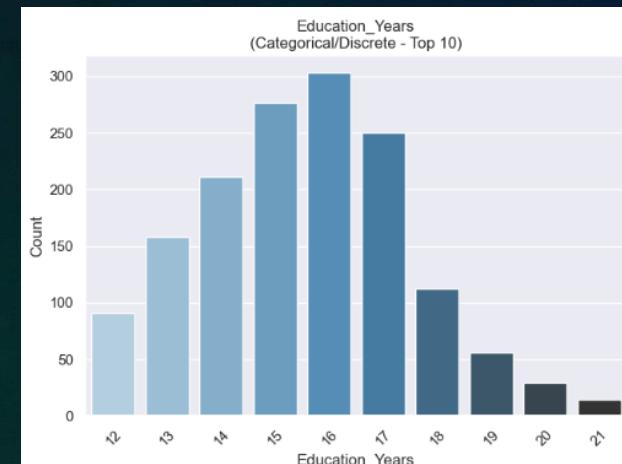
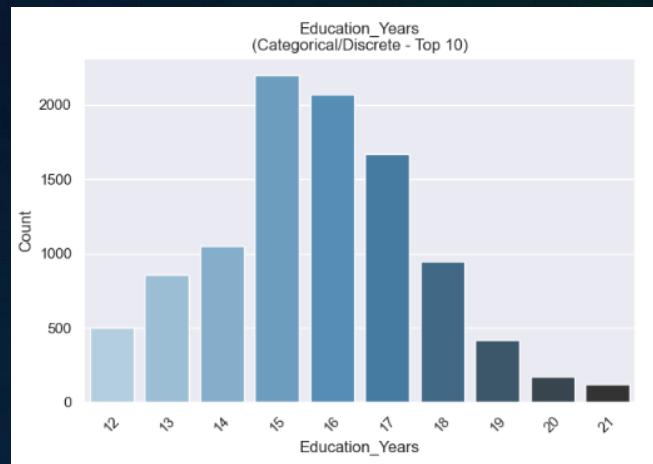
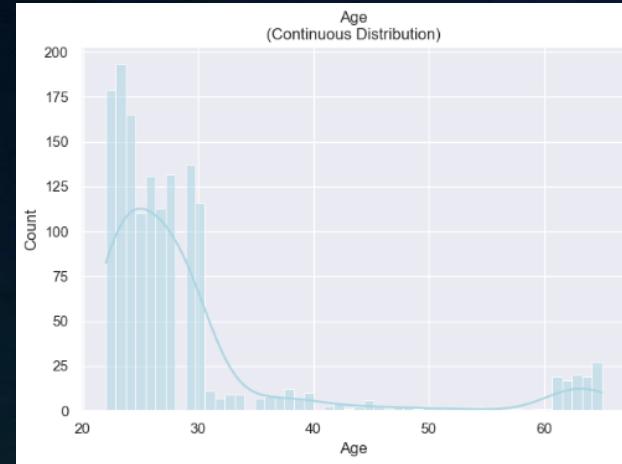
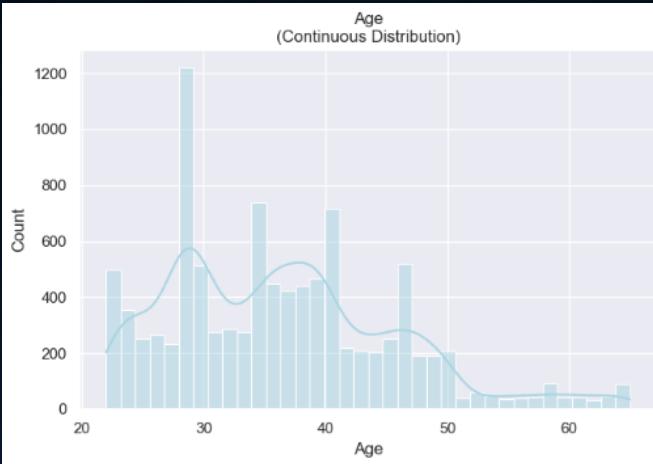


מניתוח האוכלוסייה במאתר המומלצים לפיטורים ניתן לראות כי יש עמידה מלאה בקריטריונים שהוגדרו:

- חווית דעת (ימין למטה) הרוב המכריע (93%) היו בעלי ציון קטן מ 8
- שביעות רצון (שמאל) ככל המאתר היו בעלי ציון נמוך מ 4
- בכירות (ימין למעלה) הרוב (80%) היו Juniors

– Business Use Case

קבלת החלטות מבוססות נתונים בנוגע לכוח אדם



גם במקרים בהם לכוארה הייתה חריגה מהקריטריונים,
התפלגות נתוני כלל האוכלוסייה יכול להסביר זאת:

- גיל (ימין למעלה)
 מרבית המאטר (68%) אינם בתחום המבוקש
(עד 24 או מעל 60)

- שנות לימוד (ימין למטה)
 מרבית המאטר (69%) אינם בתחום המבוקש
(פחות מ 15 שנות לימוד)

מצפיה בהתפלגות כלל האוכלוסייה
(בגרפים המקבילים שמשמאלי)
ניתן לראות כי עבור עמודות אלו, אין מספיק עובדים
 בתחום המבוקש שעומדים גם ביתר הקריטריונים

HiT

תודה על ההקשבה

שאלות ?