



ניהול והנדסת

תשתיות נתוני עתק בענן

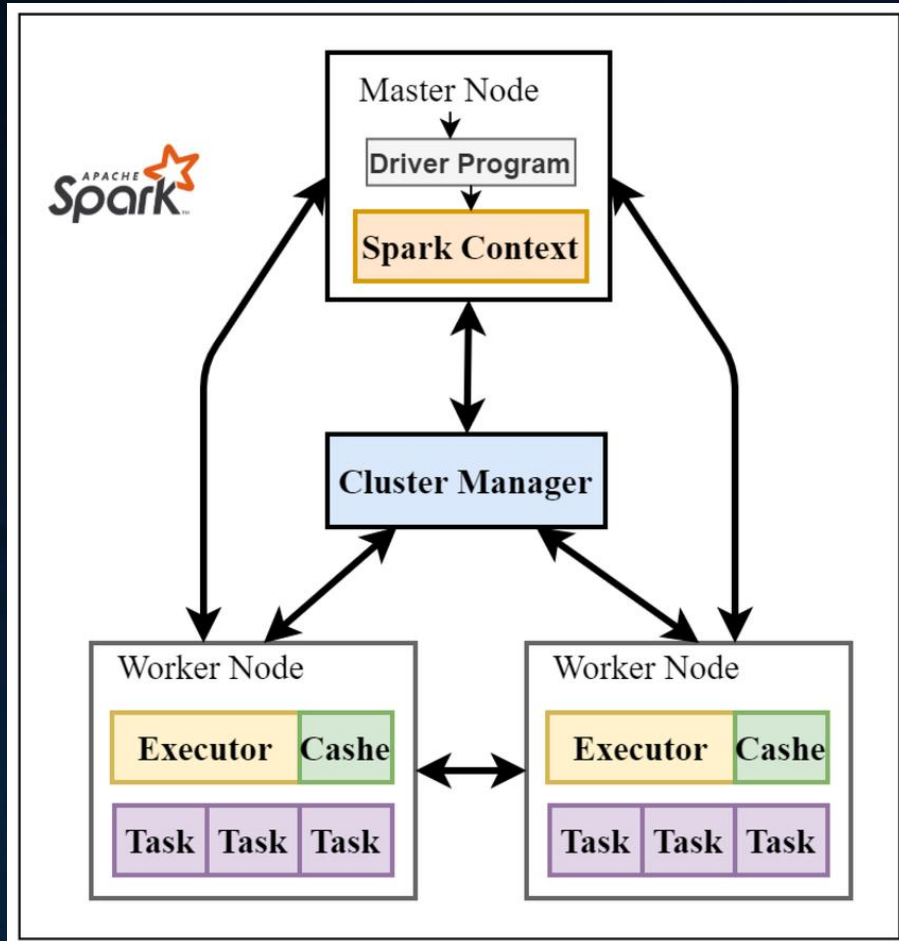
עבודה scikit-learn & PySpark

תשתית Apache Spark / PySpark

- PySpark הוא מנוע אחד (Unified Engine) לעיבוד נתוני עתק (Big Data)
- ממשק PySpark: ה-Python API המאפשר אינטגרציה בין עולם ה-Data Science (Pandas, Scikit-learn) לבין יכולות העיבוד המבוזרות של Spark (הכתוב ב-Scala)
- עקרונות הליבה:
 - In-Memory Computing: טעינת הנתונים לזיכרון
 - Partitioning: חלוקת סט הנתונים (Dataset) לתתי-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 שחוסך זמן ומשאבים יקרים

PySpark – ארכיטקטורת העיבוד

- עיבוד מבוזר על גבי אשכול שרתים (Distributed Cluster)



- Master Node (Driver): ה"מוח" המנהל מתכנן את המשימות ומחלק אותן
- Cluster Manager: ה"סדרן" מקצה משאבי מחשוב בענן (כמו YARN או Kubernetes)
- Worker Nodes (Executors): ה"פועלים" השרתים, הכוללת משאבי CPU וזיכרון, שמבצעים את החישוב בפועל על הנתונים
- Horizontal Scalability: יכולת התרחבות אופקית לטיפול בנפחי נתונים (TB/PB) שאינם ניתנים לטעינה בזיכרון של מחשב יחיד, באמצעות הוספת Worker Nodes על פי הצורך
- השרתים עובדים במקביל ומאפשרים עיבוד מקבילי של נתונים בנפח עצום, ללא תלות במגבלות של מכונה בודדת

PySpark – יתרונות, אתגרים ושימושים

• יתרונות (Pros):

- Scalability: גמישות והרחבה מהירה
- Speed: עיבוד מהיר בזיכרון (In-Memory)
- Versatility: תמיכה רחבה בשפות
- Cost-efficiency: ניצול משאבים חכם

• אתגרים (Challenges):

- Resource management: ניהול משאבים מורכב
- Learning curve: עקומת למידה תלולה
- Integration effort: מורכבות בחיבור מערכות
- Security: אבטחה והרשאות גישה

• שימושים נפוצים (Use Cases):

- Data Processing: עיבוד נתונים מסיבי.
- ETL: בניית Pipelines
- Machine Learning: אימון מודלים
- Real time analytics: ניתוח הנתונים והצגה ויזואלית בזמן אמת

ההבדל העיקרי: Pandas vs PySpark

מאפיין	Pandas (רגיל Python)	PySpark (תשתיות ענן)
נפח הנתונים	קטן עד בינוני - מוגבל ל-RAM של המחשב (לרוב עד כ 16GB – 32GB)	Big Data - יכול לעבד כמויות גדולות של נתונים (PB)
עיבוד	טורי על Single Node (מחשב אחד)	מקבילי ומבוזר – Cluster / Parallel
ביצועים	מהיר מאוד על נתונים קטנים	יעיל רק על נתונים בהיקף גדול
אופן חישוב	Eager (מיידית)	Lazy (עצל) – נדחה בהתאם לתכנון

PySpark – היישום בפרויקט זה

1. שימוש ב-PySpark בפרויקט

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

2. למה Spark?

- Industry Standard - עבודה בכלים המקובלים בשוק
המותאמים לסביבת עבודה אמיתית (Production-ready, Fault Tolerance)
- Scalability & Future-Proofing - תשתית המוכנה לצמיחה בנפחי הנתונים ללא שינוי קוד

הפרוייקט – הגדרת הבעיה והיעד העיסקי

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

הפרוייקט - השלבים בעיבוד הנתונים



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

הפרוייקט – סקירת ומבנה הנתונים

- מקור הנתונים – קובץ נתוני עובדים (בשלב זה נטענו 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

הפרוייקט – בחינה 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

• בחינת כל אחד מהמאפיינים

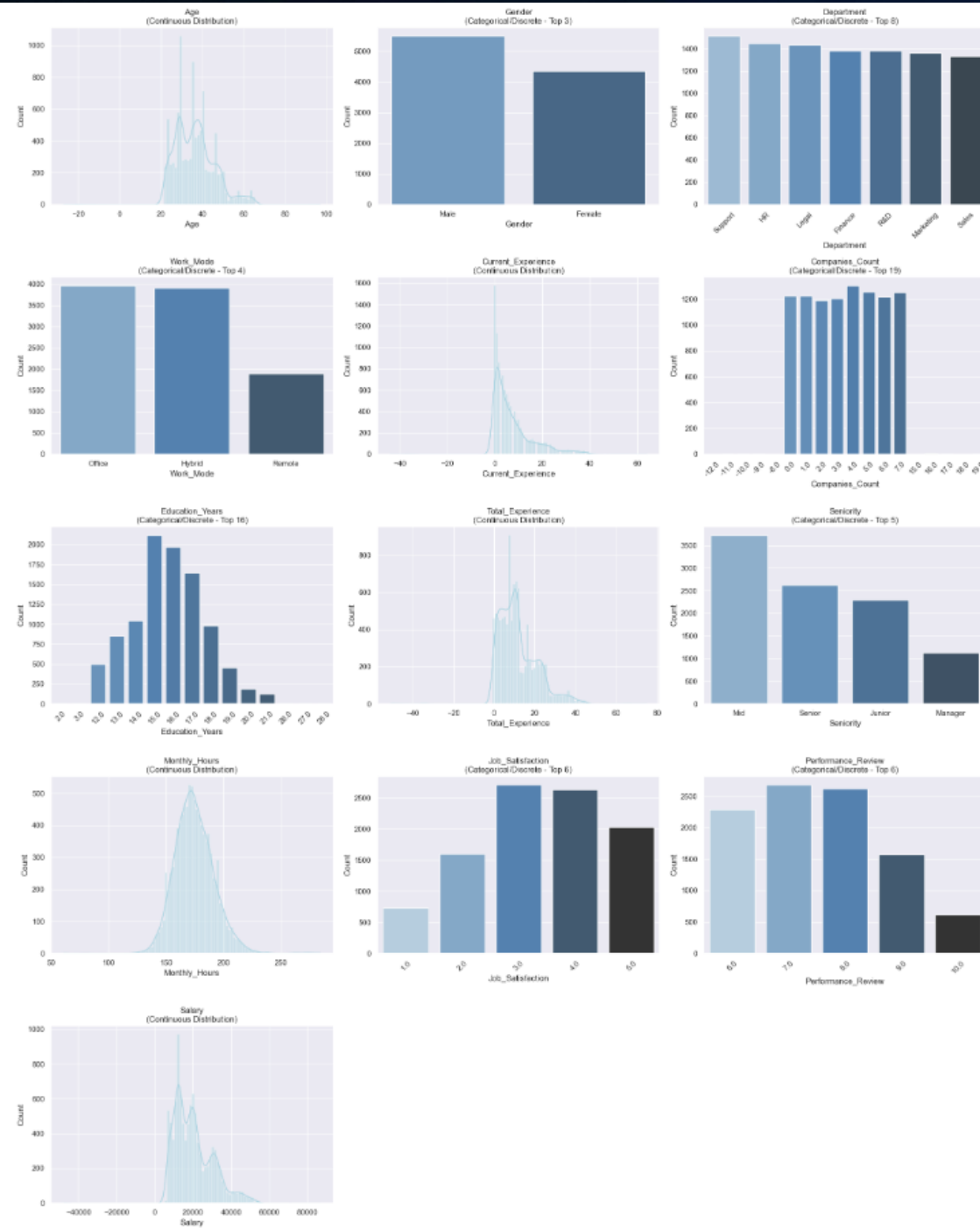
• שיעור הכיסוי (כמות ואחוז הערכים החסרים)

• 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: וידוא אחידות בפורמטים של משתנים קטגוריאליים



הפרוייקט – ניקוי וטיוב הנתונים

- הגדרת כללים לוגיים (נתונים מספריים):

- ניקוי ערכים לא סבירים
ברמת כל שדה

- ניקוי ערכים לא סבירים
בהצלבה בין שדות

ניקוי נתונים – מחיקת ערכים קטנים מ 0 בכל שדה נומרי

```
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 – מגבלת גיל העובד

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

ניקוי נתונים – הגבלת מספר שנות הלימוד בהתאם לגיל

```
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 – הגבלת מספר שנות הניסיון הכולל בהתאם לגיל

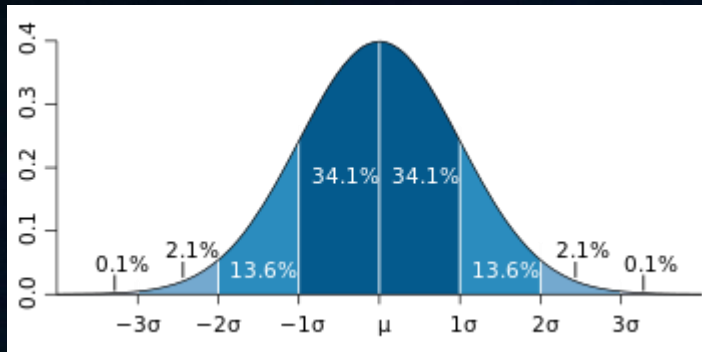
```
dfc = dfc.withColumn("Total_Experience",  
    F.when(F.col("Total_Experience") > (F.col("Age") - 18), F.lit(None))  
    .otherwise(F.col("Total_Experience")))
```

ניקוי נתונים – הגבלת הוותק בתפקיד הנוכחי בהתאם לשנות הניסיון הכולל

```
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 – הגבלת כמות החברות בהתאם לשנות הניסיון הכולל

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



הפרוייקט – ניקוי וטיוב הנתונים

- טיפול בחריגים בנתונים מספריים (Outliers):
זיהוי והסרת ערכי קיצון שעלולים להטות 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
מחדש בחינה של הנתונים

הפרוייקט - טיפול בערכים קטגוריאליים

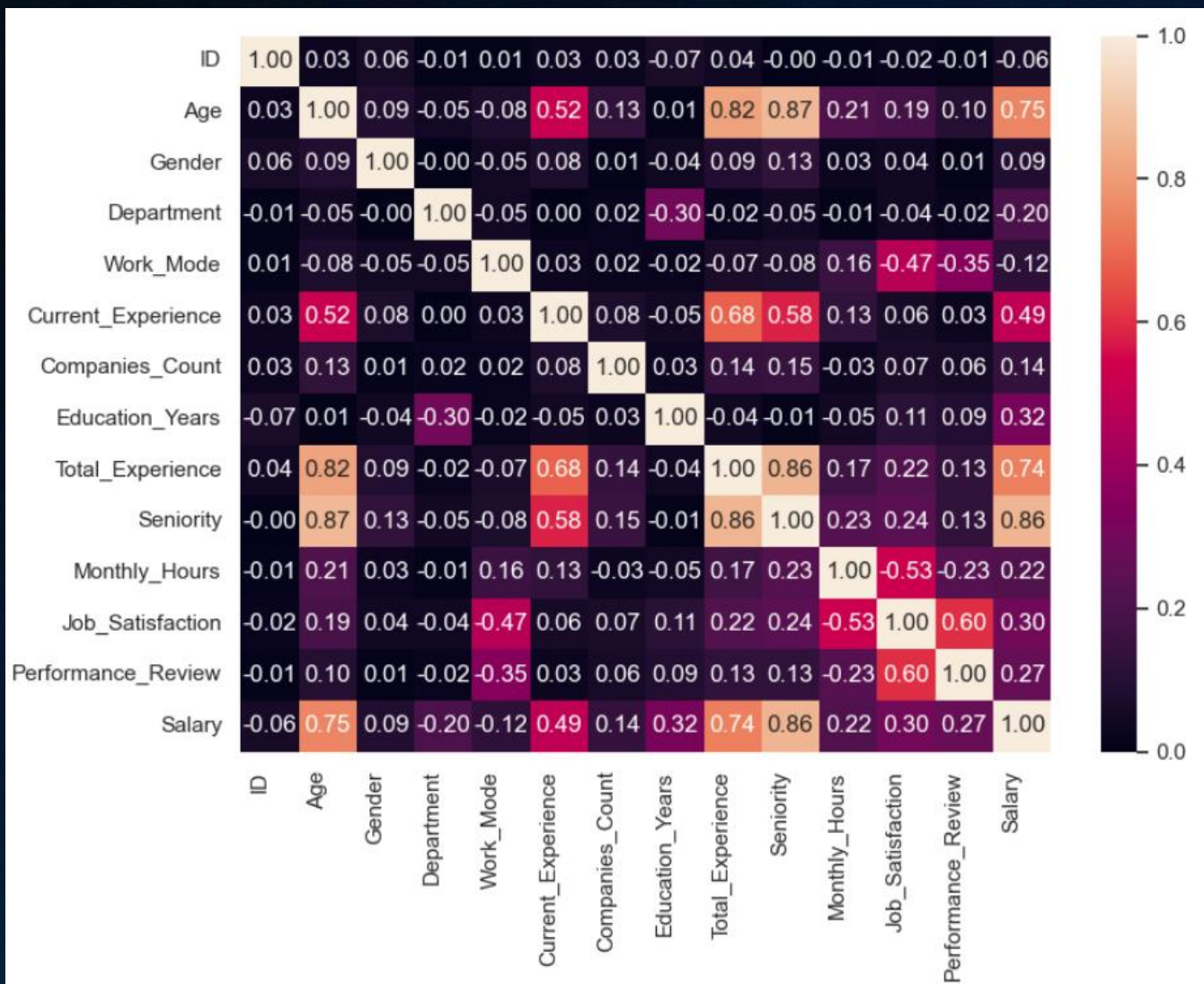
קידוד משתנים קטגוריאליים לערכים נומריים (Label Encoding)

```
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 מחלקה או מגדר)
לערכים נומריים המיצגים אותם
באופן שהמודל יכול לעבד.

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

הפרויקט – בחינת קורלציות



ניתוח הקורלציות:
לטובת ניתוח נוסף של הקשרים בין
העמודות השונות,
השתמשתי במטריצת מתאמים
(Pearson Correlation Heatmap)
המאפשרת לזהות תלויות ליניאריות בין
העמודות

הפרוייקט – השלמת ערכים חסרים

- על מנת שאוכל לבנות מודל חיזוי מספיק חזק, נדרש תחילה להשלים את הנתונים החסרים, שגדלו משמעותית לאחר שלב הניקוי והטיוב
- בהתאם לקורלציות שאותרו 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, עשיתי שימוש במודל רגרסיה
- עבור עמודות מחלקה ושנות לימוד, עמודות משמעותיות עם קורלציה נמוכה לעמודות אחרות, עשיתי שימוש במודל קלסיפיקציה

הפרוייקט – השלמת ערכים חסרים

דוגמא: מימוש לעמודת מודל העבודה

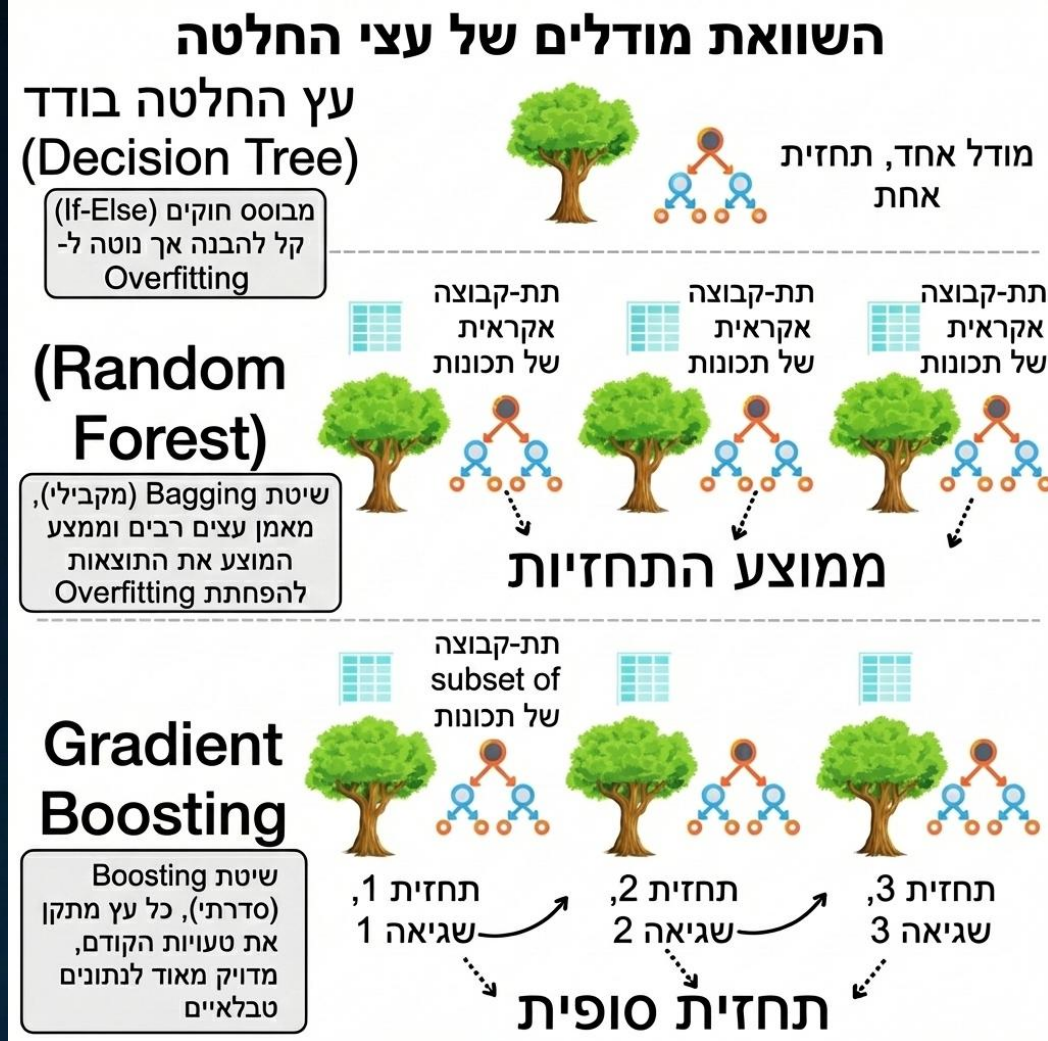
```
dfnum = fill_with_mode(dfnum, [
    ["Education_Years", "Seniority", "Job_Satisfaction", "Department"],
    ["Education_Years", "Seniority", "Department"],
    ["Education_Years", "Department"],
    ["Education_Years"],
    ["Seniority"],
    ["Job_Satisfaction"],
    ["Department"]
], "Work_Mode")

--- 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 ---
```

שיטה 1 - שימוש בקורלציות
עבור עמודות עם קורלציה גבוהה
לעמודות אחרות או עמודות עם
משמעות פחותה
(גיל, מגדר, מודל עבודה, וותק בחברה
נוכחית, מספר חברות, וותק כולל,
בכירות, שעות עבודה חודשיות,
שביעות רצון, חוות דעת) הסתמכתי
על עמודות נוספות עם קורלציה חזקה
לעמודה המבוקשת,
ובהסתמך על עמודות אלה חושב
השכיח לכל קבוצה ונעשה בו שימוש
להשלמת הערכים החסרים

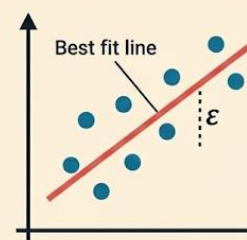
הפרוייקט – השלמת ערכים חסרים

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



השוואת מודלים של רגרסיה

רגרסיה ליניארית (Linear Regression)

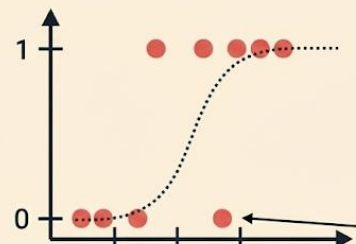


$$y = \beta_0 + \beta_1 x + \varepsilon$$

מנסה למצוא את קו המגמה הטוב ביותר שמתאים לנתונים רציפים. מטרתה לחזות ערך מספרי (כמו מחיר (כמו מחיר דירה).

נתונים רציפים

רגרסיה לוגיסטית (Logistic Regression)

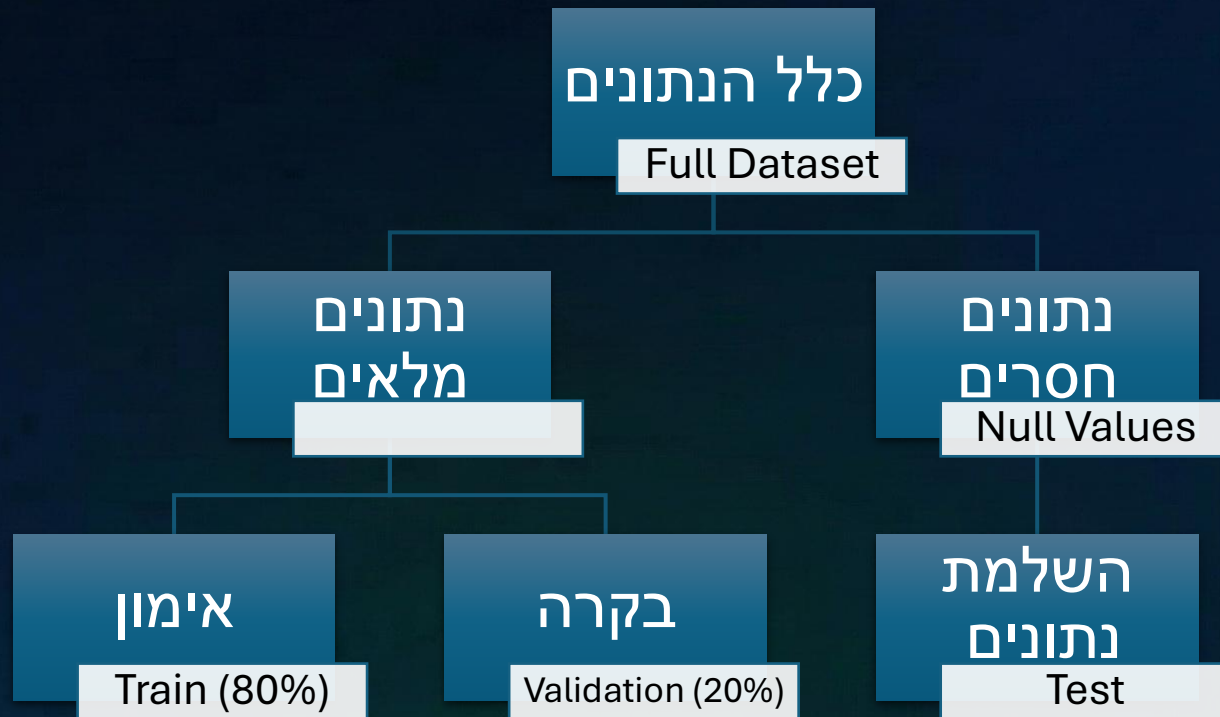


משתמשת בפונקציית סיגמואיד כדי לסווג נתונים לקטגוריות (0 או 1). מטרתה לחזות הסתברות לסיווג (כמו לקוח שינטוש/לא ינטוש).

נתונים קטגוריים/בינאריים

הפרוייקט – השלמת ערכים חסרים

חלוקת הנתונים לטובת המודל



הפרוייקט – השלמת ערכים חסרים

שיטה 2 – מודל רגרסיה (עבור עמודת שכר)

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) בנתונים המודל מצליח להסביר (שואפים ל-1).

- RMSE (Root Mean Squared Error):

מדד שגיאה הרגיש במיוחד לסטיות גדולות ו"מעניש" עליהן, ולכן עדיף כשחשוב להימנע מטעויות קיצוניות.

- 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 ערכים חסרים

Confusion Matrix (מטריצת הבלבול):

שיטה 3 – מודל קלסיפיקציה

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

עבור חיזוי ערך קטגוריאלי, נשתמש במודל קלסיפיקציה – מודל למידת מכונה שתפקידו לחזות לאיזו קטגוריה (מחלקה) שייך נתון חדש, על סמך דפוסים שלמד מנתוני העבר.

הפרמטרים לבחינת כל אחד מהמודלים:

- **Accuracy (דיוק כללי):** אחוז התחזיות הנכונות מתוך סך כל התחזיות – מדד טוב כשהנתונים מאוזנים, אך מטעה כשיש חוסר איזון (Imbalanced Data).
- **Precision (דיוק החיוביים):** מתוך כל המקרים שהמודל חזה כ"חיוביים", כמה באמת היו כאלה (מדד ה"אמינות" – חשוב כדי להימנע מ"אזעקות שוא").
- **Recall (רגישות):** מתוך כל המקרים שהם באמת "חיוביים" במציאות, כמה המודל הצליח לגלות (מדד ה"כיסוי" – חשוב כשאסור לפספס מקרים).
- **F1 Score:** הממוצע ההרמוני המשלב בין Precision ל-Recall למספר אחד, המייצג את האיזון ביניהם (המדד הקובע במקרה של נתונים לא מאוזנים).

הפרוייקט – השלמת ערכים חסרים

שיטה 3 – קלסיפיקציה

הרצתי את הנתונים באמצעות 3 המודלים (Logistic Regression, Decision Tree, Random Forest) עבור עמודות מחלקה ושנות לימוד בסיום ההרצות קיבלתי 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 לימוד –
עבור אף אחד
מהמודלים לא
התקבלה תוצאה
מספקת ולכן נחזור
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	1745	
weighted avg		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
ביותר התקבלה
עבור מודל
Logistic
Regression

הפרויקט – ניתוח הנתונים

למידה לא מונחית וניתוח אשכולות
(Unsupervised Learning)

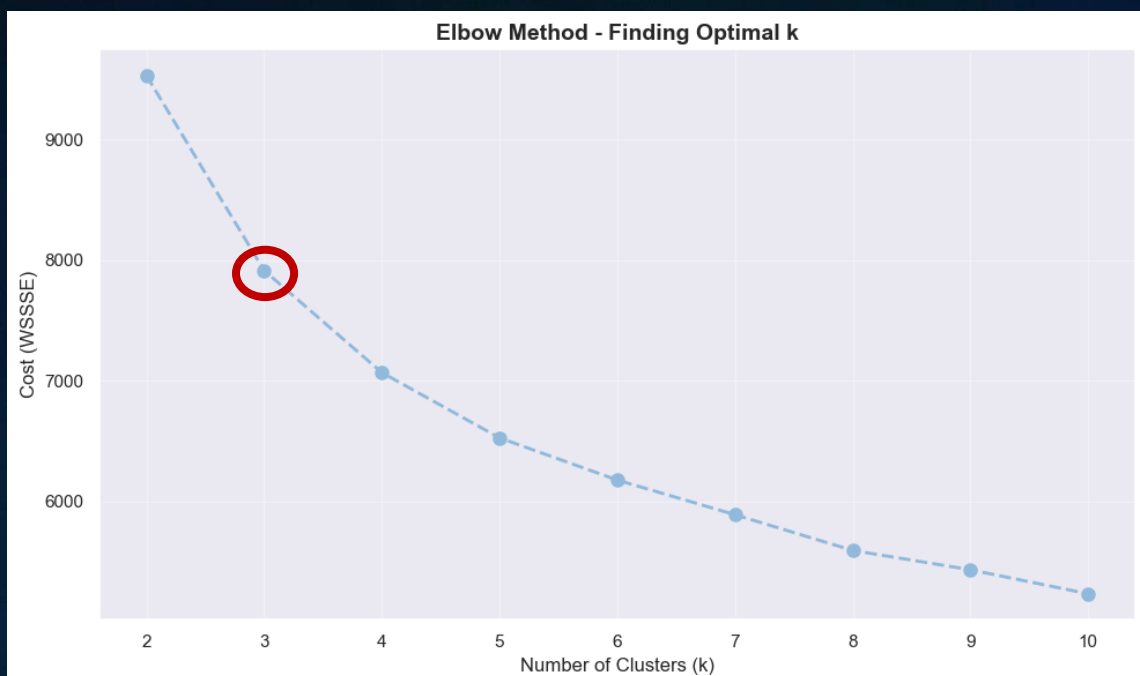
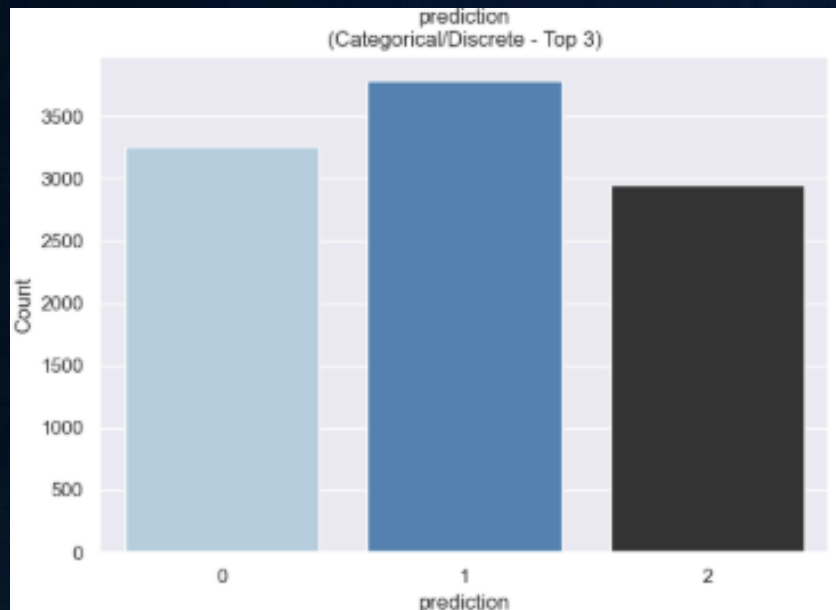
• K-Means

אלגוריתם המחלק את הנתונים ל K קבוצות
נפרדות על בסיס דמיון וקרבה מרחבית.

• Elbow Method (שיטת המרפק)
כלי גרפי לבחירת מספר האשכולות האופטימלי
ע"י זיהוי נקודת האיזון בין דיוק ליעילות.

• Silhouette Score
ציון איכות (בין 1- ל-1) הבוחן עד כמה האשכולות
מופרדים היטב זה מזה ומלוכדים בתוכם.

	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



הפרוייקט – ניתוח הנתונים

- אשכול 0

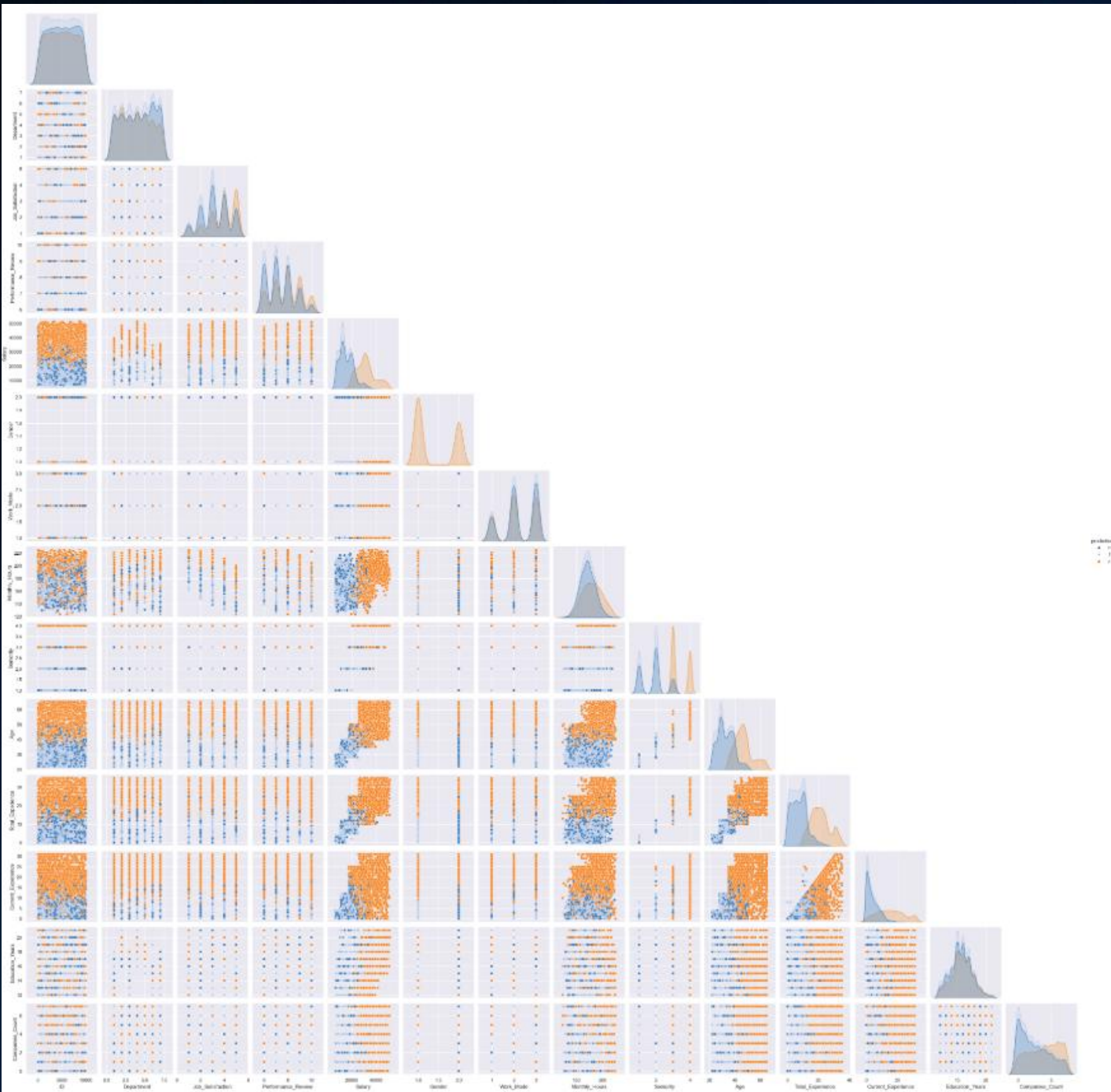
נשים בתחילת ואמצע הקריירה
(דרג ביניים).

- אשכול 1

גברים בתחילת ואמצע הקריירה
(דרג ביניים).

- אשכול 2

הבכירים וההנהלה – העובדים המנוסים
והמתוגמלים ביותר (מכל המגדרים)



הפרוייקט – ניתוח הנתונים

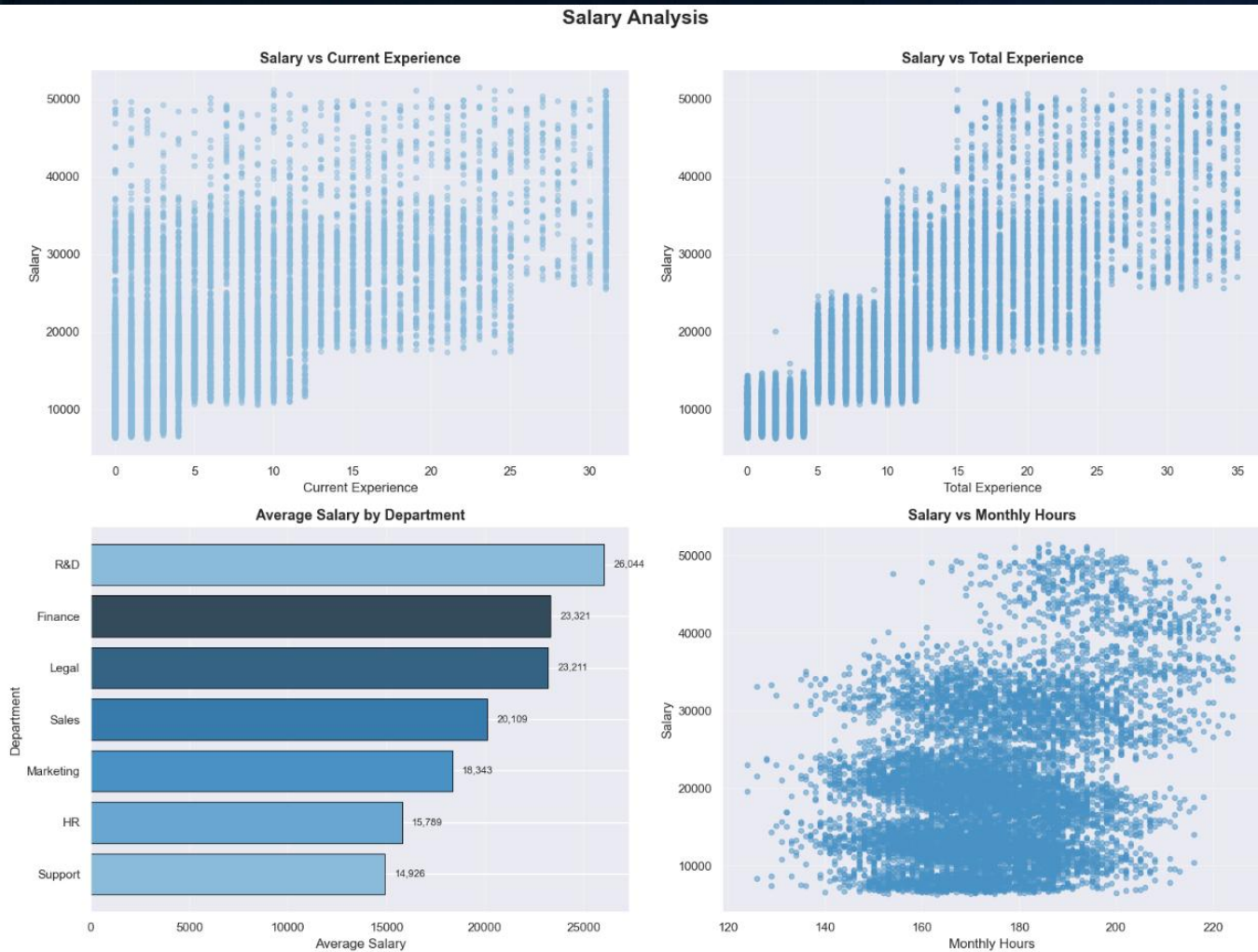
מתאמי תכונות (Salary Analysis)

- Total Experience (וּותק כללי): המנבא החזק ביותר. ניתן לראות מבנה "מדרגות" המייצג את המעבר בין דרגות בכירות (Junior/Mid/Senior).

- Current Experience (וּותק בחברה נוכחית): מציג מתאם חיובי לשכר, אך עם "רעש" (Variance) גבוה יותר בהשוואה לוותק הכללי.

- Department (מחלקה): משתנה מפריד משמעותי. פערי שכר ברורים בין מחלקות הליבה (R&D, Finance) למחלקות התפעול (Support).

- Monthly Hours (שעות עבודה): פיזור רחב ("ענן נקודות") וקורלציה נמוכה. היקף השעות אינו מהווה אינדיקטור אמין לגובה השכר.



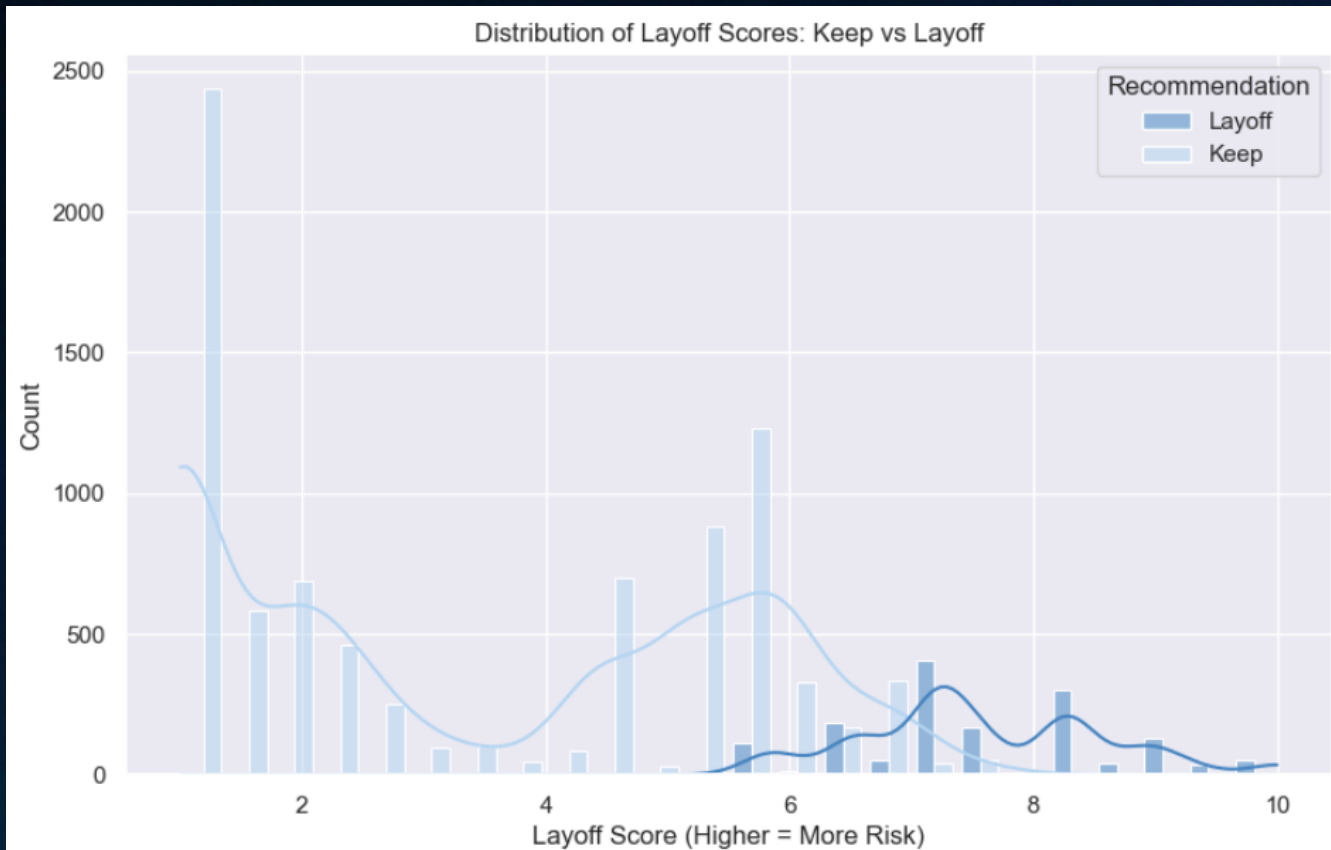
– Business Use Case

קבלת החלטות מבוססות נתונים בנוגע לכוח אדם

- המטרה - בניית מודל להמלצה על פיטורי עובדים על בסיס פרמטרים מוגדרים מראש היעד הינו בניית מאתר של 15% מכלל כוח האדם
- לוגיקת הדירוג - מודל מבוסס LLM המעניק לכל עובד ציון כדאיות פיטורים (1-10), כאשר 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 בדרג ה-Junior
- אילוש קשיח - תהליך הצמצום חייב לשמור על התפלגות זהה של כוח האדם בין המחלקות כדי לא לפגוע בפעילות הליבה של מחלקה ספציפית

– Business Use Case

קבלת החלטות מבוססות נתונים בנוגע לכוח אדם



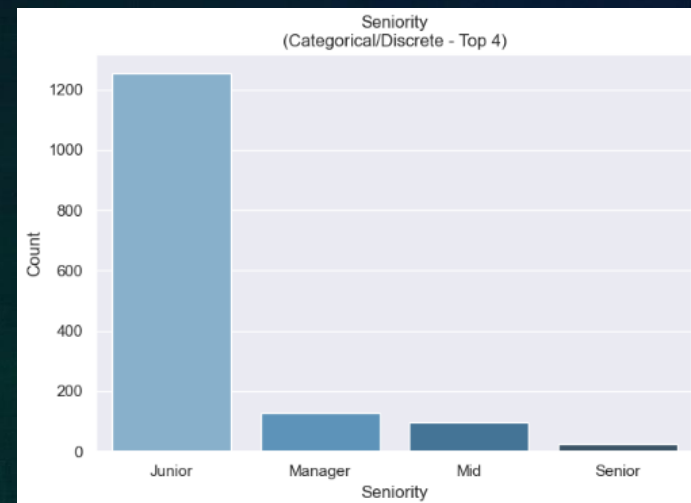
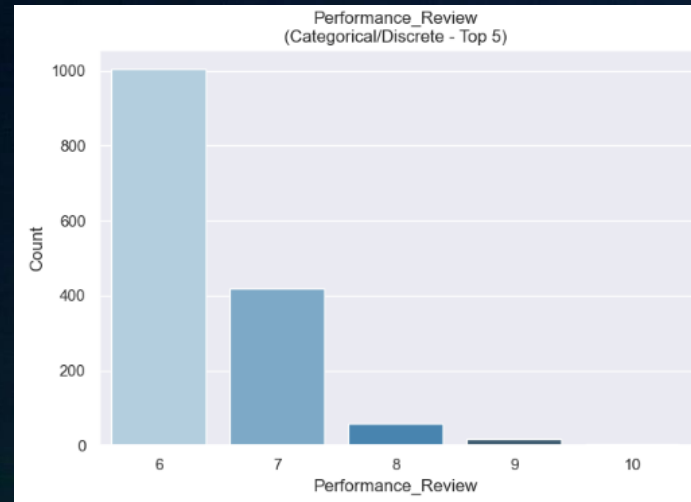
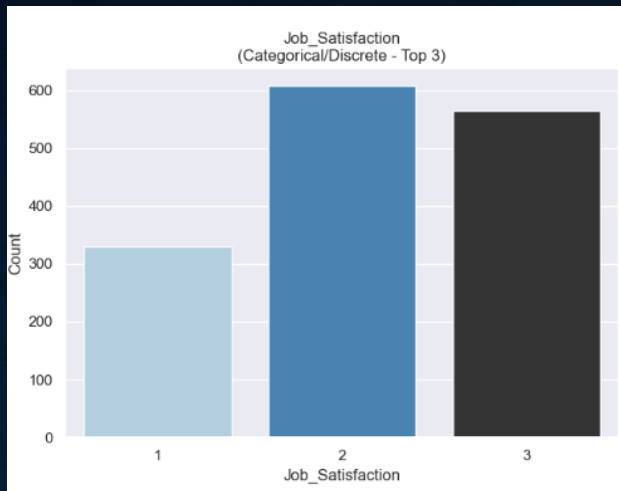
בגרף ניתן לראות את התפלגות הערך של ציון כדאיות הפיטורים

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

זאת למרות שיש כאלו עם ציון גבוהה יותר (עד 7.58), בשל האילוצ של שמירה על התפלגות כ"א בין המחלקות

– Business Use Case

קבלת החלטות מבוססות נתונים בנוגע לכוח אדם

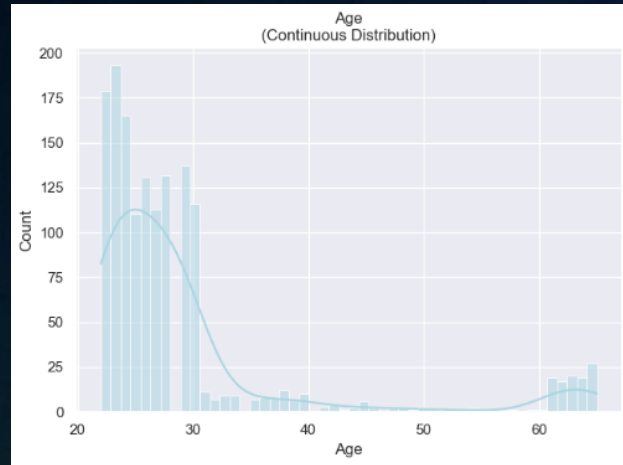
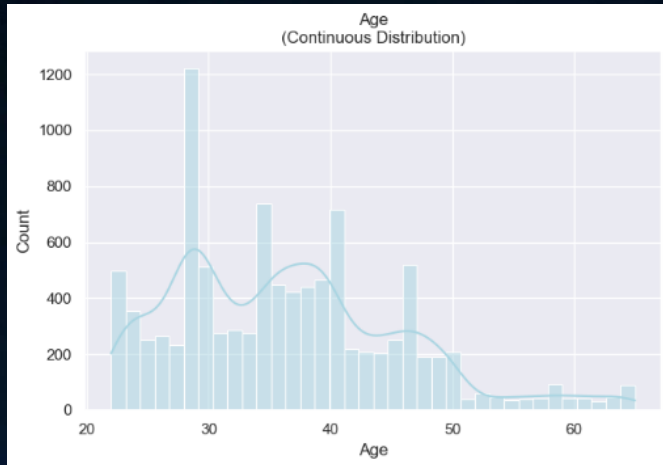


מניתוח האוכלוסייה במאתר המומלצים לפיטורים ניתן לראות כי יש עמידה מלאה בקריטריונים שהוגדרו:

- חוות דעת (ימין למעלה) הרוב המכריע (93%) הינו בעל ציון קטן מ 8
- שביעות רצון (שמאל) כלל המאתר הינו בעל ציון נמוך מ 4
- בכירות (ימין למטה) הרוב (80%) הינם Junior

– Business Use Case

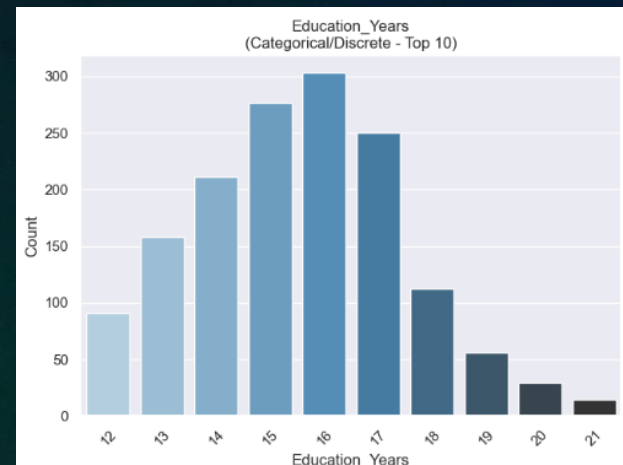
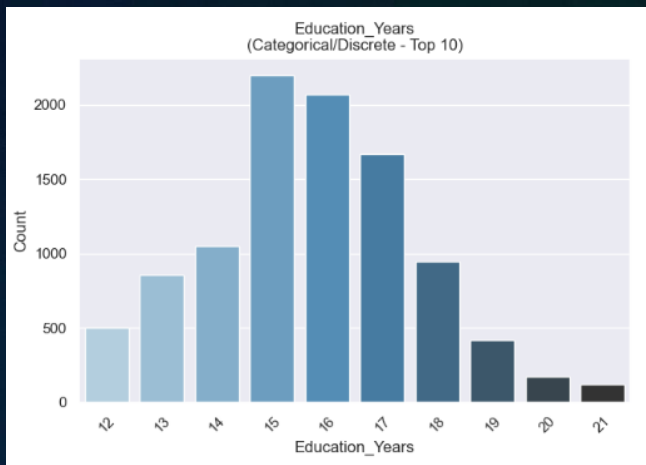
קבלת החלטות מבוססות נתונים בנוגע לכוח אדם



גם במקרים בהם לכאורה היתה חריגה מהקריטריונים, התפלגות נתוני כלל האוכלוסייה יכול להסביר זאת:

- גיל (ימין למעלה)
מרבית המאתר (68%) אינו בטווח המבוקש
(עד 24 או מעל 60)

- שנות לימוד (ימין למטה)
מרבית המאתר (69%) אינו בטווח המבוקש
(פחות מ 15 שנות לימוד)



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



ד!i

תודה על ההקשבה

שאלות ?