Matemàtica Computacional i Analítica de Dades

Aprenentatge Computacional

Pràctica 1: Regressió

Autors (NIU):

Àlex Correa Orri (1564967)

Júlia Pumares Benaiges (1566252)

Curs 2021-2022, 1r semestre

Índex

1 Descripció del dataset

1.1 Llista d'Atributs

- **EmployeeName (Text)**: Nom complert del treballador
- **EmpID (Text)**: Número únic d'identificació per a cada treballador
- **MarriedID (Binary)**: Està la persona casada? (1 o 0 per sí o no, respectivament)
- **MaritalStatusID (Integer)**: Codi d'estat marital que coincideix amb el text de 'MaritalDesc'
- **GenderID (Binary)**: Gènere (1 per home i 0 per dona)
- **EmpStatusID (Integer)**: Codi d'estat laboral que coincideix amb el text de 'EmploymentStatus'
- **DeptID (Integer)**: Codi d'identificació de departament que coincideix amb el departament en el que treballa
- **PerfScoreID (Integer)**: Codi de qualificació del rendiment que coincideix amb la qualificació més recent del seu rendiment.
- **FromDiversityJobFairID (Binary)**: Prové l'empleat de la fira de l'empleat justa? 1 o 0 per sí o no, respectivament.
- **Salary (Float)**: El salari anual de la persona en dòlars estatunidencs.
- **Termd (Binary)**: Ha estat aquest treballador fet fora? 1 o 0 per sí o no, respectivament.
- **PositionID (Integer)**: Un enter que indica la posició de la persona.
- **Position (Text)**: El nom/títol de la posició que té la persona
- **State (Text)**: L'estat en el que viu la persona.
- **Zip (Text)**: El codi postal de l'empleat
- **DOB (Date)**: La data de naixament del treballador.
- **Sex (Text)**: Sexe - M o F
- **MaritalDesc (Text)**: L'estat marital de la persona ("Divorced" -divorciat-, "Single" -solter-, "Widowed" -vidu-, "Separated" -separat-, etc.)
- **CitizenDesc (Text)**: Etiqueta per la qual una persona és un Ciutadà o No Ciutadà
- **HispanicLatino (Text)**: Camp de "Yes" (Sí) o "No" per identificar si el treballador és hispanic/llatí.
- **RaceDesc (Text)**: Descripció/text sobre la raça amb la que s'identifica la persona

- **DateofHire (Date)**: Data en la que la persona va ser contractada.
- **DateofTermination (Date)**: Data en que la persona va ser acomiadada, només si 'Termd' = 1
- **TermReason (Text)**: Raó o descripció de perquè la persona va ser acomadiada.
- **EmploymentStatus (Text)**: Categoria de la situació laboral del treballador. Si treballa a temps complet: "Active" -actiu-.
- **Department (Text)**: Nom del departamnet on treballa la persona.
- **ManagerName (Text)**: El nom del gerent del treballador.
- **ManagerID (Integer)**: Un identificador unic per cada gerent.
- **RecruitmentSource (Text)**: El nom de la font de reclutamnet del treballador.
- **PerformanceScore (Text)**: Qualificació de rendiment. Hi ha 4 categories: "Fully Meets" -Compleix completamnet-, "Partially Meets" -compleix parcialment, "PIP" -pla de millora del rendiment-, "Exceeds" -supera les espectavites-.
- **EngagementSurvey (Float)**: Resultats de l'enquesta de compromís, realitzada per un soci extern.
- **EmpSatisfaction (Integer)**: Una qualificació (de l'1 al 5) feta pel treballador en una enquesta de satisfacció.
- **SpecialProjectsCount**: El nombre de projectes especials en que ha treballat l'empleat durant els últims 6 mesos.
- **LastPerformanceReviewDate (Date)**: La data més recent de l'avaluació sobre el rendiment del treballador.
- **DaysLateLast30 (Integer)**: El nombre de vegades que el treballador va arribar tard durant els últims 30 dies.
- **Absences (Integer)**: El nombre de vegades que el treballador va estar absent a la feina.

1.2 Preparació de les dades: Canvis al Dataset

A partir de la base de dades donada no podiem treballar ja que hi havia molts atributs catègorics i altres atributs redundants i per això hem eliminat o substituït atributs.

Atributs que hem eliminat:

- **EmployeeName**: El nom del treballador no ens aporta informació per fer l'estudi ni per fer una regressió.

- **MarriedID**: L'atribut 'MaritalDesc' ja conté aquesta informació.
- **DeptID**: Aquest atribut coincideix amb l'atribut 'Department'.
- **ZIP**: Ja tenim l'atribut 'State' que és bastant similar.
- **ManagerID**: Concideix amb l'atribut 'ManagerName'.
- **MaritalStatusID**: Coincideix amb 'MaritalDesc'.
- **EmpStatusID**: Coincideix amb 'EmploymentStatus'.
- **PositionID**: Coincideix amb 'Position'.
- **Sex**: Coincideix amb 'GenderID'.
- **PerformanceScore**: Coincideix amb el 'PerfScoreID'.
- **LastPerformanceReviewDate**: L'hem eliminat perquè és una data i no hem considerat que sigui rellevant.

Atributs que hem passat de categòrics a binaris: **MaritalDesc**, **CitizenDesc**, **RaceDesc**, **TermReason**, **EmploymentStatus**, **Department**, **RecruitmentSource**, **Position**, **State**, **ManagerName**.

També hi ha atributs que tenien un format de data i els hem passat a nombre de dies.

- **DOB**: Hem restat la data de naixament respecte la data de referència 01/01/2020 i així hem calculat l'edat en dies creant el nou atribut 'Age' de tipus integer.
- **DateofHire** i **DateofTermination**: Hem restat aquestes dues dates per obtenir el nombre de dies treballats. En cas que no hi haguès una 'DateofTermination' hem fet servir la data 01/01/2020. Hem obtingut el nou atribut 'WorkedDays' de tipus integer.

Per a obtenir informació més rellevant sobre el dataset i els seus atributs, treiem els 'outliers' del 'Salary', 'WorkedDays' i 'EngagementSurvey'. Posteriorment (veurem en els següents apartats) treurem els atributs menys correlacionats i ens quedarem amb aquells que determinen (en major mesura) el salari. En algunes de les proves treurem tots els atributs binaris ja que obtindrem millors resultats.

2 Apartat C: Analitzant Dades

2.1 Normalitat dels atributs

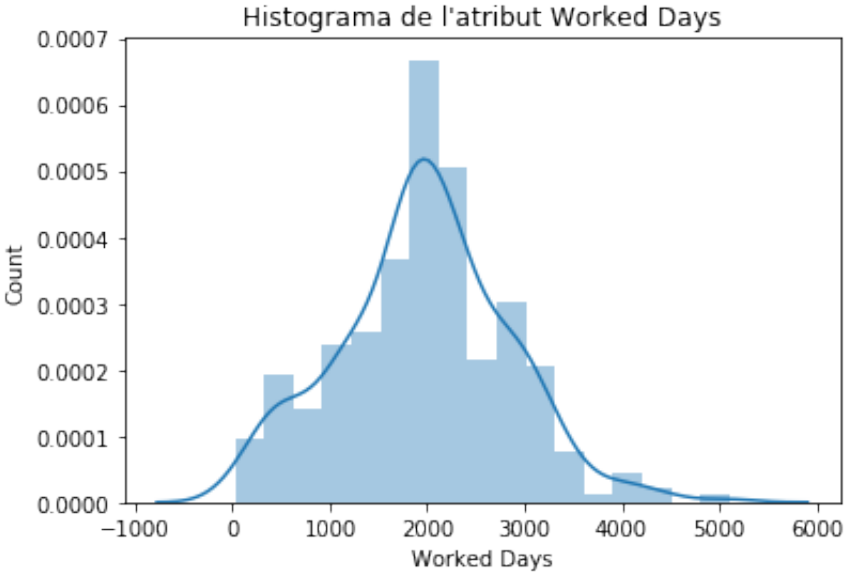
Tenim alguns atributs binàris i ja sabem que no tindràn una distribució gaussiana. Per la resta d'atributs hem fet servir la funció *shapiro* i també hem graficat la distribució de cada atribut.

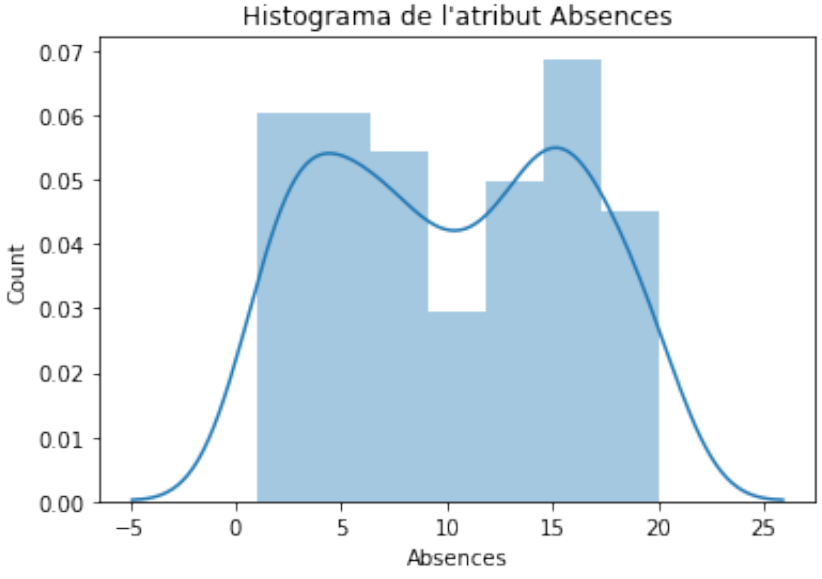
2.1.1 Test de Shapiro

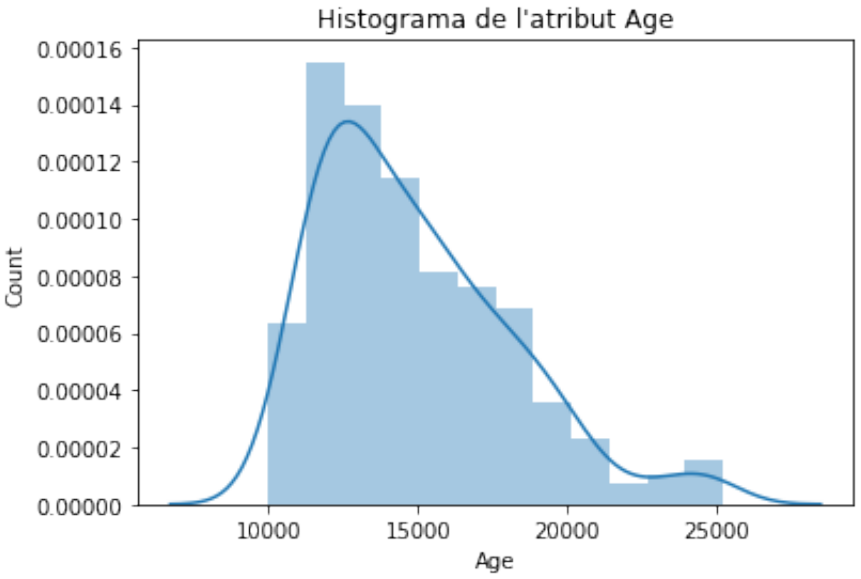
En el test de Shaìro plantejem la H_0 (hipòtesi nul.la) que és que les dades estàn distribuides normalment. Amb la funció shapiro obtenim un *p-value* (que va de 0 a 1), si aquest és molt petit rebutjem la H_0. En tots els atributs ens ha sortit un *p-value* inferior a 0.05. I per tant, les nostres dades no seguixen una distrbució normal. La majoria dels valors ens han sortit al voltant de 10^{-35} o 10^{-25}. La que ens ha sortit més alt ha sigut 'WorkedDays' amb *p-value* 0.0004966543638147414. El següent valor més alt ja era de l'ordre de 10^{-8}.

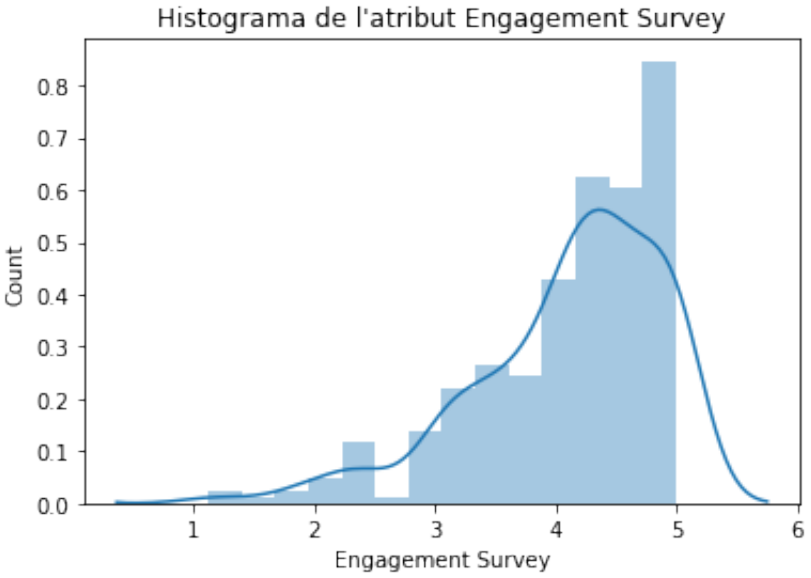
2.1.2 Histogrames dels atributs

També hem fet un gràfic dels atributs per veure la seva distribució. El gràfic de 'WorkedDays' que és l'altribut que més s'acosta a una gaussiana de tots els que tenim, tot i que no té una distribució gaussiana:





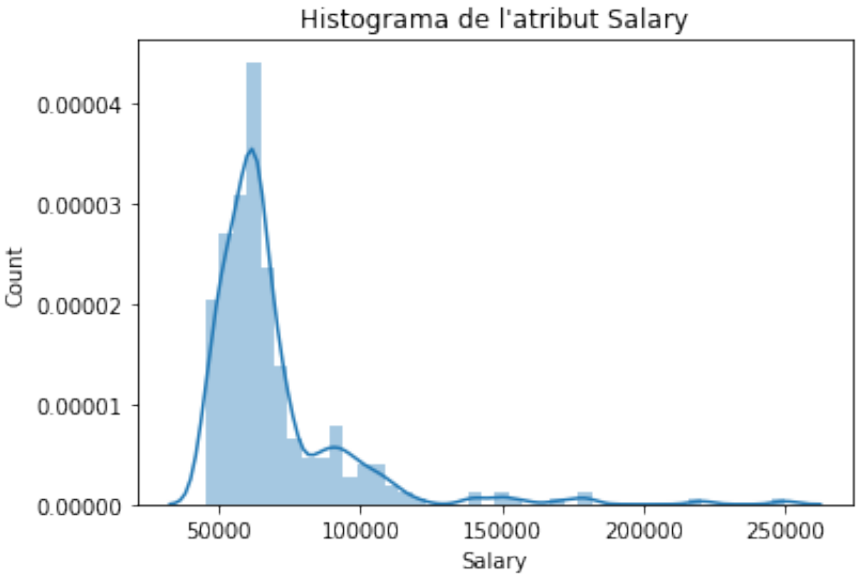




Aquests són els atributs que ens han sortit amb un *p-value* més gran, però com veiem en els gràfics no tenen una distribució gaussiana.

2.2 Atribut Objectiu

El dataset que tenim recull les dades dels treballadors d'una empresa. Per això ens ha semblat interessant intentar predir el salari d'un treballador a partir de les dades donades, com ara el departament en el que treballa, el temps que porta a l'empresa, l'edat, el sexe, la raça, el resultat de les enquestes de satisfació, l'avalució del seu rendiment. Això també ens pot servir per veure si factors com el sexe o la raça influeixen en el salari obtingut o si aquest es basa en el rediment del treballador i el tipus de feina que fa.

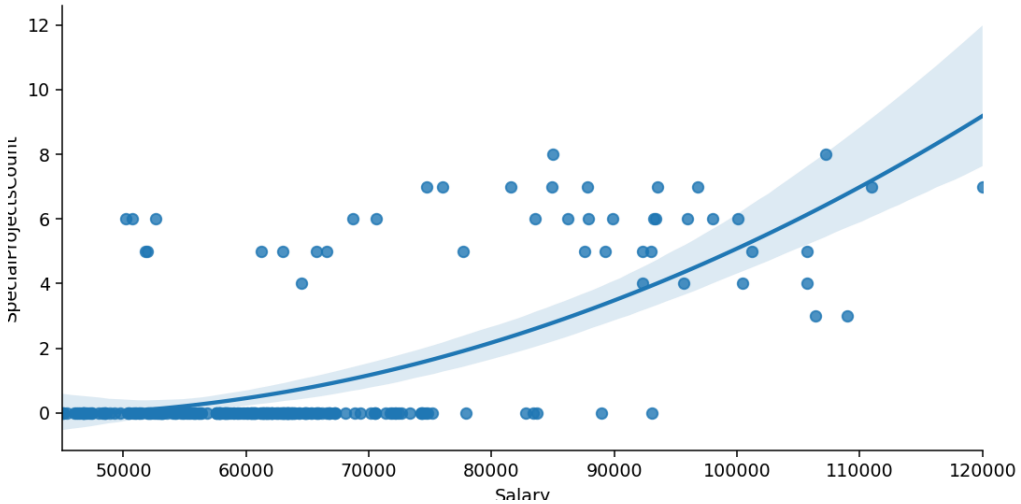


En l'histograma del 'Salary' veiem que hi ha molta gent que cobra poc i després menys gent que cobra molt, que deuen ser els que tenen algún càrrec alt dins la jerarquia de l'empresa.

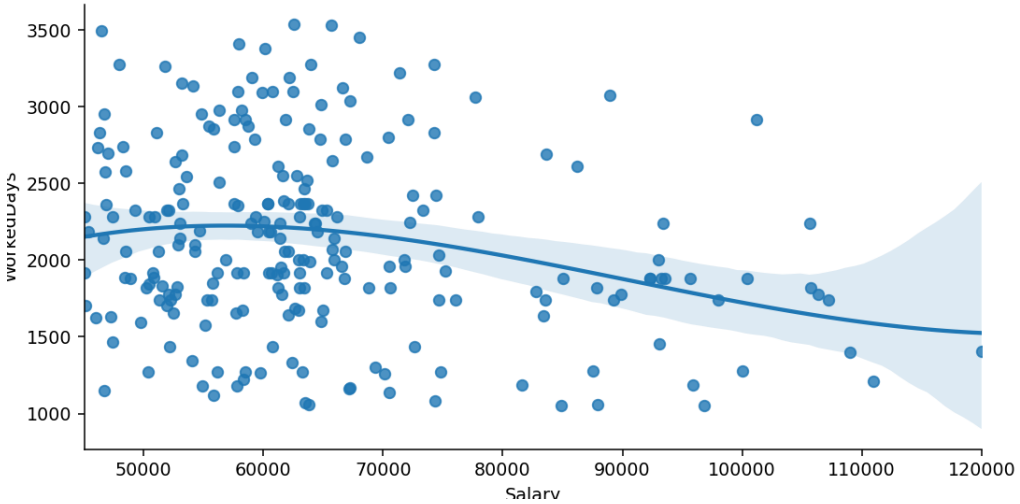
2.3 Relacions entre els atributs numèrics de la BBDD

A continuació veurem algunes relacions entre diferents atributs i el Salari, que és el nostre atribut objectiu:

- 'SpecialProjectsCount': Veiem que la majoria de treballadors no han realitzat projectes especials, però aquells que n'han fet, majoritàriament surten afavorits en quant al salari. Gràfic (ordre 2):

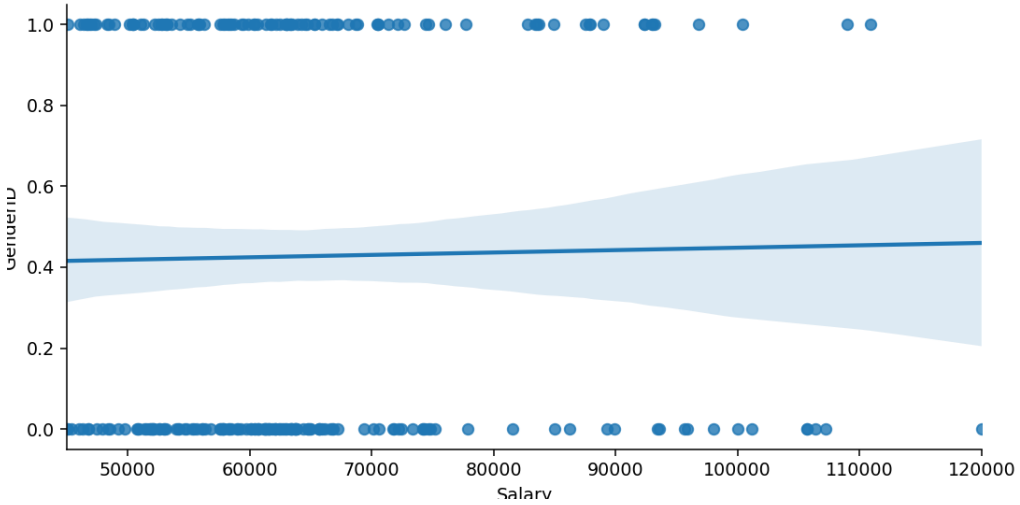


- 'WorkedDays': Veiem que no hi ha una tendència clara i sembla que casi hi ha un núvol distribuit uniformement. Tot i això hi ha una lleugera tendència que els treballadors amb menys dies treballats a l'empresa tenen major salari. Entenem que al tractar-se de casos més aviat puntuals es refereix a caps de l'empresa i no la mitjana de treballadors. Gràfic (ordre 3):



Per tant, sembla que el salari vindrà més determinat pel càrrec o el tipus de feina que s'exerceix que no per l'antiguitat adins l'empresa.

- 'GenderID': Veiem que el salari no ve gaire determinat pel gènere dels treballadors encara que s'aprecia un lleuger pendent positiu en la recta, que mostra un petit increment del salari pel gènere masculí. (1 home, 0 dona). Gràfic (ordre 1):



3 Apartat B: Primeres regressions

3.1 Correlació entre els atributs

Per veure la correlació entre els diferents atributs hem fet un *heatmap*. Per veure la correlació de cada atribut amb el 'Salary' que és el nostre atribut objectiu hem fet servir la funció *spearmanr* que calcula el coeficient de correlació i un *p-value* que indica si la correlació és significativa o no. Ens interessa que

ens surti un valor proper a 1 en valor absolut i que el *p-value* ens surti molt petit, inferior a 0.05. A continuació es mostra una taula dels atributs i les seves correlacions amb l'atribut objectiu:

```
Atribut 0 GenderID
SpearmanrResult(correlation=0.016369541120544954, pvalue=0.8032976754305855)
--------------------------------------------------------------------------------
Atribut 1 PerfScoreID
SpearmanrResult(correlation=0.05943515384289806, pvalue=0.3654046015775446)
--------------------------------------------------------------------------------
Atribut 2 Salary
SpearmanrResult(correlation=1.0, pvalue=0.0)
--------------------------------------------------------------------------------
Atribut 3 EngagementSurvey
SpearmanrResult(correlation=-0.04916642226496004, pvalue=0.45414238818473784)
--------------------------------------------------------------------------------
Atribut 4 EmpSatisfaction
SpearmanrResult(correlation=0.022303598279226046, pvalue=0.7343124560673149)
--------------------------------------------------------------------------------
Atribut 5 SpecialProjectsCount
SpearmanrResult(correlation=0.49881349735630093, pvalue=3.999473828559406e-16)
--------------------------------------------------------------------------------
Atribut 6 DaysLateLast30
SpearmanrResult(correlation=-0.06679918035027281, pvalue=0.30891756232964573)
--------------------------------------------------------------------------------
Atribut 7 Absences
SpearmanrResult(correlation=0.030628359352945132, pvalue=0.6411265897347012)
--------------------------------------------------------------------------------
Atribut 8 WorkedDays
SpearmanrResult(correlation=-0.16018934318325848, pvalue=0.014160491591920595)
--------------------------------------------------------------------------------
Atribut 9 Age
SpearmanrResult(correlation=-0.02110194465504348, pvalue=0.7481309395379521)
--------------------------------------------------------------------------------
```

Veiem que l'únic atribut que ens dona una correlació que pot ser determinant amb un *p-value* molt baix (menor a 0.05) és el recompte de projectes especials. El nombre de dies treballats sembla que també serà important, però no és una correlació elevada.

3.2 Atributs per la regressió

Per a començar fem servir els 10 atributs que hem estat utilitzant fins ara: 'GenderID', 'PerfScoreID', 'Salary', 'EngagementSurvey', 'EmpSatisfaction', 'SpecialProjectsCount', 'DaysLateLast30', 'Absences', 'WorkedDays' i 'Age'

3.3 Normalització

Després de fer algunes proves amb regressions veiem que l'error és molt elevat (ordre 10^8) i, per tant, decidim normalitzar les dades per a reduir l'error (fins a entre 0.2 i 0.5).

3.4 Atributs Significants

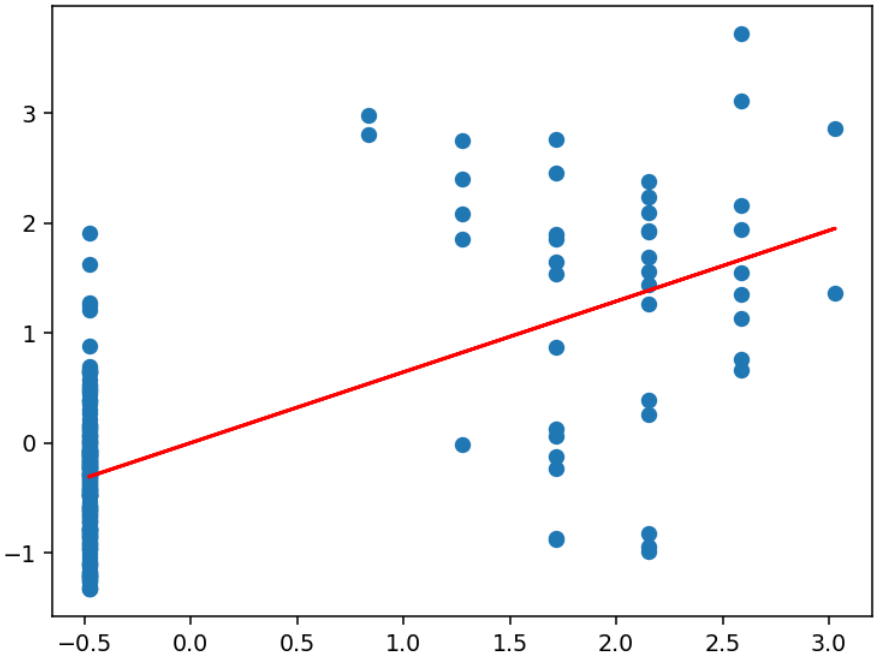
3.4.1 Amb tots els atributs

Per començar hem predit el 'Salary' utilitzant cadascún dels atributs, en total 133. I després hem agaft només els atributs que tinguessin una correlació amb el 'Salary' que fos superior a 0.1. En cada execució

que feiem ens surtien errors molt diferents per cada atribut, per exemple en alguna execució en que per cap atribut s'arribava a un error que baixés de 1 i en altres hi havia atributs en els quals l'error s'acostava a 0.2. Atribuïm aquesta diferència de resultats a l'aletorietat al crear les dades de Test i les de Valadició, el nostre conjunt de dades és massa petit (tenim 311 files al dataset). Donada aquesta diferència en els resultats hem calculat quins són els atributs que sortien més vegades amb el mínim error. Hem trobat que amb l'atribut que s'assoleix un mse menor és el de 'Department Production'. És el 2n que estava més correlacionat amb 'Salary'. Altres atributs que també han donat un mse baix són: 'SpecialProjectsCount', 'ManagerNameJennifer Zamora', 'DepartmentIT/IS', 'PositionProduction Technician I'.

3.4.2 Amb només els 10 atributs més correlacionats i no binaris

Ara provarem de fer les regressions amb els 10 atributs mencionats anteriormnent, els 10 atributs més correlacionats no binaris (creats a partir de categòrics). Quan provem de fer regressions bàsiques només trobem 1 atribut amb un error notablement menor a 1, el 'SpecialProjectsCount', amb un MSE = 0.585, i un 'R2 score' = 0.415. Tot i això segueix sense ser gaire bo, com podem veure al següent gràfic:



4 Apartat A: El descens del gradient

Per a implementar el mètode del descens del gradient definim una classe 'Regressor' a partir de la qual cridarem les funcions per a entrenar i predir principalment:

```python
class Regressor(object):
    def __init__(self, w0, w1, alpha):
        # Inicialitzem w0 i w1 (per ser ampliat amb altres w's)
        self.w0 = w0
        self.w1 = w1
        self.alpha = alpha

    def predict(self, x):
        # implementar aqui la funció de prediccio: f[i] = w0 + w1 * x[i]
        hy = []
        for xx in (x):
            hy.append(self.w0 + self.w1*xx)
        return hy

    def __update(self, hy, y, x):
        # actualitzar aqui els pesos donada la prediccio (hy) i la y real.
        # Calculem les derivades de J respecte w0 i w1
        m=len(y)
        d_w0 = 0
        d_w1 = 0
        for i in range(m):
            d_w0 = d_w0 + hy[i] - y[i]
            d_w1 = d_w1 + (hy[i] - y[i])*x[i]

        l = 0.1
        d_w0 = d_w0/m
        d_w1 = (d_w1 - l*self.w1)/m

        #Calculem les noves w0 i w1
        self.w1 = self.w1 - self.alpha * d_w1
        self.w0 = self.w0 - self.alpha * d_w0
```

```python
    def train(self, max_iter, epsilon, x, y):
        # Entrenar durant max_iter iteracions o fins que la millora sigui inf
        y_pred = self.predict(x)
        i = 0
        J = 1
        m = len(y)
        while (i < max_iter and J > epsilon):
            self.__update(y_pred, y, x)
            y_pred = self.predict(x)
            #Calculem J segons la formula
            for j in range(m):
                J = J + (y_pred[j] - y[j])**2
            l = 0.1
            J = (J + l*(self.w0**2 + self.w1**2))/(2*m)
            i += 1
            print(J)
        print("")
        print("f[i] = " + str(self.w0) + " + " + str(self.w1) + "*x[i]")
        return J, self.w0, self.w1
```

4.1 Polinomi d'ordre 1 amb 1 atribut

Comencem aplicant el mètode amb un polinomi amb 1 sol atribut i d'ordre 1. El que té més sentit, i amb el que també hem obtingut millors resultats (menor error), és utilitzar l'atribut amb una major correlació, el 'SpecialProjectsCount'.

En les primeres iteracions l'error està al voltant de 0.42 i aconseguim reduir-lo fins a 0.29 aproximadament.

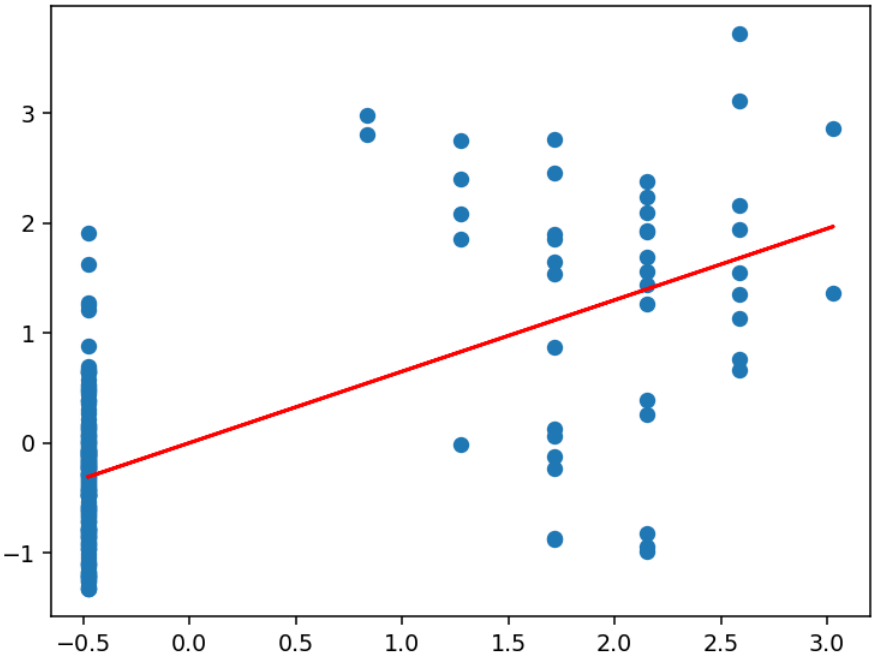
Veiem que si modifiquem el valor d'epsilon (tolerància) no tenim gaire variació en el resultat ja que molt aviat s'estabilitza a 0.29...

Pel que fa el valor d'alpha (en el nostre programa la variable 'l'), no cal que sigui petita ja que ens aproximem molt ràpid a un valor estable. Comencem fent proves amb 0.1 però l'anem incrementant fins a 2.0 on comencem a veure que ja el tercer decimal de l'error varia. Agafem per bo aquest valor d'alpha i obtenim els següents resultats:

- Error inicial: 0.4296351265752844
- Error final: 0.29517816060336777
- Alpha: 2.0
- Epsilon: 0.15
- Polinomi obtingut:

$$f[i] = 2.1585623705526015e-05 + 0.6493913495346555 \times x[i]$$

- Gràfic de la recta amb les dades de salari en funció de l'atribut que hem fet servir:



4.2 Polinomi d'ordre 2 amb 1 atribut

Ara modifiquem lleugerament la classe i les funcions per a obtenir un regressor de la forma:

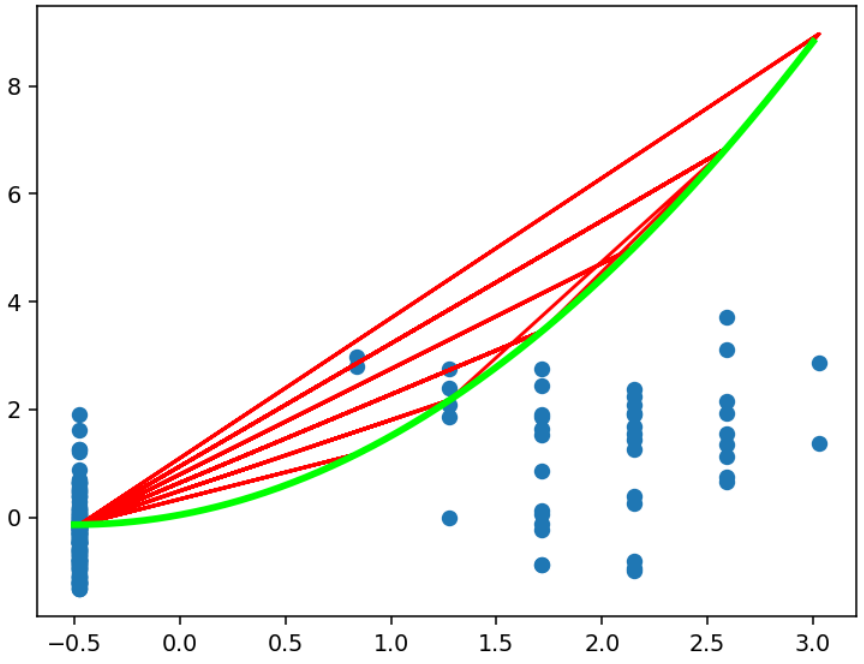
$$f[i] = w0 + w1 \times x[i] + w2 \times x[i]^2$$

En aquest cas obtenim una molt lleugera millora en l'error utilitzant els mateixos paràmetres. No és una regressió amb millores significants:

- Error inicial: 1.044764392947517
- Error final: 0.2938545429341532
- Alpha: 2.0
- Epsilon: 0.15
- Polinomi obtingut:

$$f[i] = 0.04660516443781354 + 0.731228521526849 \times x[i] + 0.731228521526849 \times x[i]^2$$

- Gràfic del polinomi amb les dades de salari en funció de l'atribut *SpecialProjectsCount*:



4.3 Polinomi d'ordre 2 amb 2 atributs diferents

Tornem a modificar la classe i funcions per a obtenir un regressor de la forma:

$$f[i] = w0 + w1 \times x1[i] + w2 \times x2[i]^2$$

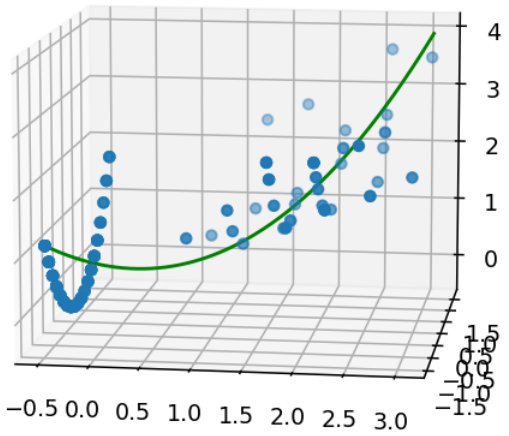
Comencem provant amb els atributs 'SpecialProjectsCount' i 'DaysLateLast30' respectius a x1 i x2, i

veiem un dels casos estudiats a la teoria, l'error tendeix a infinit. Disminuim la aplha i ens surt un error de 0.294. Si invertim l'ordre dels atributs obtenim un error de 0.36 aprox.
Encara que l'atribut 'SpecialProjectsCount' tingui major correlació amb el salari que els altres atributs, aconseguim el millor error quan l'assignem a x1 ja que és la variable d'ordre 1. El que acaba fent el mètode és trobar una w1 relativament gran i una w2 molt petita, i així obtenim una paràbola molt suau i molt més propera a una recta (com la que hem provat a l'inici). D'aquesta manera és bastant indiferent quin atribut agafem per a la variable x2. Així que combinant qualsevol dels atributs amb l'atribut *SpecialProjectsCount* ens surt un error al voltant de 0.29. Això vol dir que els altres atributs són rellevants. Resultats:

- Atribut 1 (respectiu a x1): 'SpecialProjectsCount'
- Atribut 2 (respectiu a x2): 'Absences'
- Error inicial: 0.44381589889778655
- Error final: 0.29499072807842713
- Alpha: 2.0
- Epsilon: 0.15
- Polinomi obtingut:

$$f[i] = 0 + 0.6492496164096203 \times x1[i] + 0.01477970233638578 \times x2[i]^2$$

- Gràfic en 3 dimensions al tenir 2 atributs i el salari



5 Conclusions

Els diferents atributs de la nostra base de dades estaven poc correlacionats. Al fer els gràfics combinant 2 atributs ja veiem que en la majoria de casos no es podia observar una correlació. Hem intentat canviar atributs passant-los a binàris i també treure els *outliers* de l'atribut 'Salary', però tot i així ens seguien sortint pitjors o iguals relacions entre els atributs. En visualitzar els gràfics de les regressions lineals de l'apartat B, veiem que no eren gaire bones regressions. Hem provat amb diferents atributs i hem intentat agafar els que tenien una major correlació. Però provant amb tots els atributs possibles no hem pogut trobar cap en que la regressió fos bastant exacte. El mètode del descens del gradient, l'hem implementat i hem vist que funcionava bé però tampoc ens ha donat els resultats desitjats. En fer servir més d'un atribut en el descens del gradient no hem millorat l'error, hem mantigut el mateix que quan feiem servir un sol atribut. L'atribut que més determina el *salary* és el *SpecialProjectsCount*. Quan feiem regressió amb 2 atributs, ens ha donat el mateix error que si fessim servir només el *SpecialProjectsCount*, de manera que l'altre atribut influia poc.