PREDICTION SALARY

Salary prediction for data scientist

Cristobal Yob (100% code/100% report)
Python For engineering

Abstract

Salary prediction of Data scientists

Context

The world is full of data of different types of information and it is known that information is key to being able to predict different things and analyze and companies are realizing more that jobs are needed to help them manage, for this they are jobs in the area of data science and like every market this being so early people do not know enough how much they would have to be paid and where to look for the job that they like the most and has their best expectations

Aims

In this project we will work to give as much information as possible to new data scientists who want to know what jobs to look for, what skills are necessary, where it is safest to get the job and the sectors that they like the most and, most importantly, the salary that would be expected. from this

methods

For this work, a mixture of descriptive and inferential analysis will be made using different machine learning libraries to generate models so that people can obtain their salary and do a interface for the facility of the user

Results

What is expected in the models and the analysis is that it will be a good tool for the next data science workers to be able to choose the best jobs for them

Contents

Introduction

Context

In the world of data that we live the new jobs that are rise in the data analytics field as actuarial scientists, analysts, mathematicians, business analytic professionals, and software programming analysts serve in this fields the named data scientists are assigned with various names for the different functions and specializations these professionals focus on working with digital data with descriptive and inferential statistical studies coupled with mathematical studies and more importantly with computer awareness with which they generate models, analyze information and various other things.

Aims

For this new field, our project focuses on helping people who work or want to work in this field in the United States so that they can predict how much salary they should earn with market information. This is useful for them to have the information and as scientists or future scientists. of data occupy it and convert it into power so that they know the value of their functions and can ask for what they are worth in the market.

Method

We are gona do an analysis of data science salary data. This analysis will first have a basic descriptive analysis of the data to know how the market is behaving in terms of number of jobs, salaries, and skills for the types of works this helping for the next part that is the part of the code by us that is the part of inference in which a study and analysis will be generated in terms of linear relationships of the dependent variable which is the salary with the independent and also to be able to create and analyze normal, ridge and lasso linear regressions to be able to know well the relationship of these variables with the salary and to be able to predict the salary that you should have.

As the best way to help with data science salary prediction, the same data science functions will be used, therefore we will work with a database obtained from kaggle that gives us salary information in the United States and then with python A data cleaning will be generated, then these will be analyzed, and different prediction models will be made to obtain the salary.

What is expected from this project is to be able to give you the most information and the specific salary data in an interface for a user.

Analytical Method

In this project, data analysis methods will be used with descriptive statistics with the creation of graphs and averages in Python as well as the grouping of jobs and salaries to be able to know well where the jobs and salaries are focused, as well as the necessary skills after This begins with the inference with the assumptions to create the linear regression in the case of collinearity of the independent var, since the linear regression is multiple and the linearity that the independent var has with the dependent one, which in this case will not be treated normally due to the number and handling of variables, in addition to the creation of normal multiple linear regressions, then ridge to reduce the coefficients and be able to see the most important coefficients without having to lose variables and Lasso regression to eliminate variables and see only the most important ones.

Foreign code

In the third-party code of Kaggle, the libraries for handling vector data, graphics, machine learning and linear models are first used, such as matplotlib.pyplot for numpy graphics for vector handling and panda for handling data frames, in this you have the graphics and all this is already the author's code RUCHEIITR on the kaggle page (code link in bibliography)

In this code panda is used to read the file and inspect it to analyze it and be able to clean it in this you can see the elimination of the index that does not generate help in this case check if there is null data know the names of variables and how many options, they have each one if they are scalar or nominal in addition to generating a summary of the scalar data of the data frame

Histograms and boxplots are generated to know outlier values in which it can be seen in rating -1 that it is not an option, therefore it is a typing error, most likely the same is seen with age with values -1 these are changed according to their respective mean.

After this, a data exploration of the nominal variables is carried out, counting the jobs and for that, the number of jobs is seen by state, by sector, by industry, in this the first 10 with the most jobs are shown in a special case, 3 maxima of each this helps to know

states: California, Miami and New York.

Sectors: information tech, biotech & pharmaceuticals and business

Industry: biotech & pharmaceuticals, insurance, Computer software & hardware.

Then the top 10 with the highest average wages are shown and the average wage is shown, the highest and lowest wage for each State, Sector and Industry the top 3 are:

States: California, Illinois, Miami

Sector: aerospace & defense, biotech & pharmaceuticals and business

Industry: advertising, aerospace & defense and biotech & pharmaceuticals

Then the 3 most Popular types of data scientists that exist are job Data scientist, other scientist and data engineer

And those with the highest average salary would be Machine learning engineer, data scientists and data engineer

Then to see the best skills for each type of job like Python for data scientist and machine learning engineer and excel for data engineering

In addition to this, for the work and the necessary data frame that is dealt with at the end, the ratings are grouped into 6 categories in addition to occupying only the 5 states with the largest number of jobs.

In contrast to the total data from the beginning, already with the previous analysis, it was reduced to make it a more manageable and interpretable data frame for that, the estimated salary was transformed into average salary, then variables such as the headquarters are not necessary because we already have the ,the size when it was founded it was not taken care of because a similar variable that is earnings would be more important in addition to the sector in which the company works , what type of ownership it is and the most important skills were those that passed to the final data frame In addition, dummy variables were created for the after mentioned nominal variables.

User interface code

the other code that we use for guide is the user interface code of geeksforgeeks.org called MCQ Quiz Game, this code use Tkinter to create a interface to ask questions and display results of this questions first you have the data of questions, answers and options since this is a hit or miss quiz, then for the quiz the entire tkinter library is imported with * and then the creation of a quiz with different functions within which are used to display results information and show the correct and the wrong answers, the review of whether the question is wrong or not, the creation of the buttons such as next and quit as well as the dimensions and the type of the interface.

Own Code and Python Methods

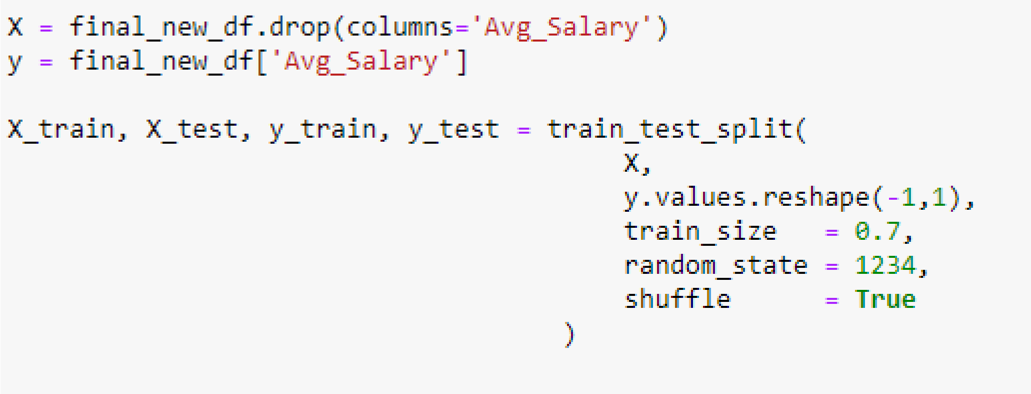
At the beginning of the code, the libraries of the other code and the functions are used to create the data frame, fix the outliers, group variables such as the rating in which the rating var is transformed into a nominal var of 6 intervals, also the grouping for the locations in the largest number of jobs, this applies to states and sectors and then we are going to take only the first 5 for the final data frame and the types of data scientists that were seen previously and the skills are more used.

In the specific methods of Python for machine learning, it was necessary to learn different libraries review of the pandas, numpy and matplot.lib libraries for the kaggle code in these libraries functions such as .drop were learned, which is used to eliminate a column the function shape that shows the number of columns and rows, the options of the graphs were also learned, it was also learned to use the seaborn library where the boxplot function was learned, in addition to the groupby that works to group nominal variables.

Now for our code we worked more with the scikit-learn machine learning library

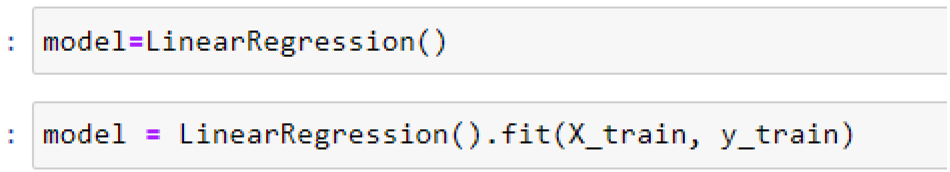
In this, many functions were imported for linear regression analysis and to be able to see them, in addition to continuing to occupy the previously named functions for things such as the linear correlation that panda gives us with. corr() or the heatmap used to generate a graph that better displays the pearson correlation from seborn than is obtained with sns.heatmap(correlation).

Already for the generation of the models, we learned to use train_test_split that is imported from sklearn.model_selection is separating the data frame in a training sector for the creation of the models 70% of the data to train models and 30% to be able to test and see if the models were good or not



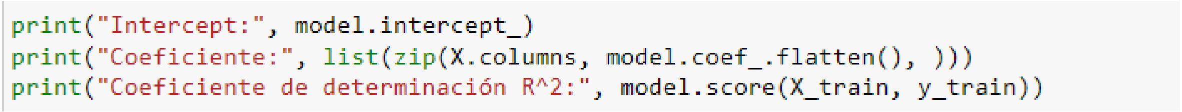
```
X = final_new_df.drop(columns='Avg_Salary')
y = final_new_df['Avg_Salary']

X_train, X_test, y_train, y_test = train_test_split(
                                        X,
                                        y.values.reshape(-1,1),
                                        train_size   = 0.7,
                                        random_state = 1234,
                                        shuffle      = True
                                    )
```

Then with these data and with the function of sklearn for LinearRegression, the model could be created, which is as easy as creating it and then generating the fit with the training data for the creation of the basic linear model.



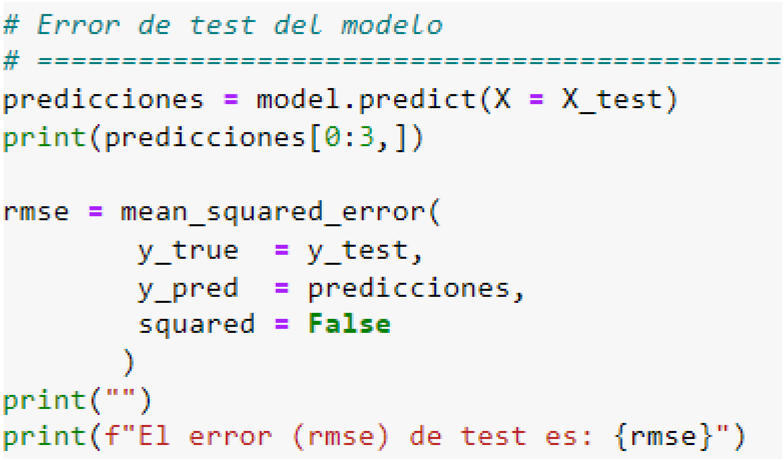
```
: model=LinearRegression()

: model = LinearRegression().fit(X_train, y_train)
```

That is created with this code where you can see the coefficients and the adjustment



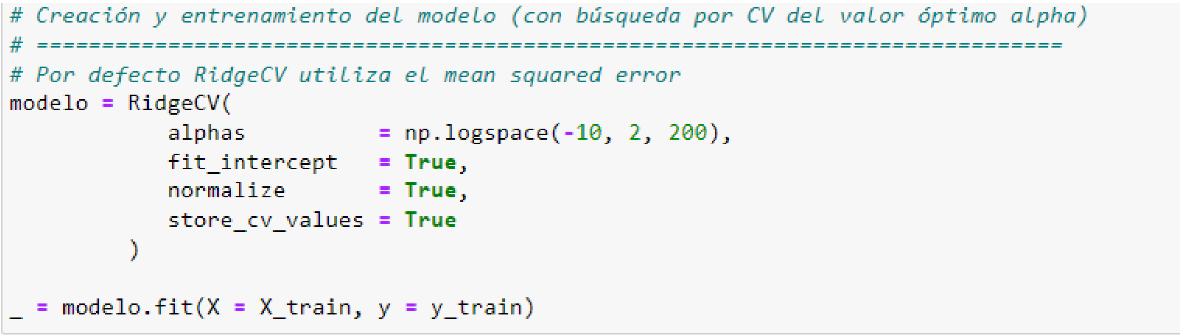
```
print("Intercept:", model.intercept_)
print("Coeficiente:", list(zip(X.columns, model.coef_.flatten(), )))
print("Coeficiente de determinación R^2:", model.score(X_train, y_train))
```

Then we do te model predict and the mean squared_error that are functions for the coming from sckit learn



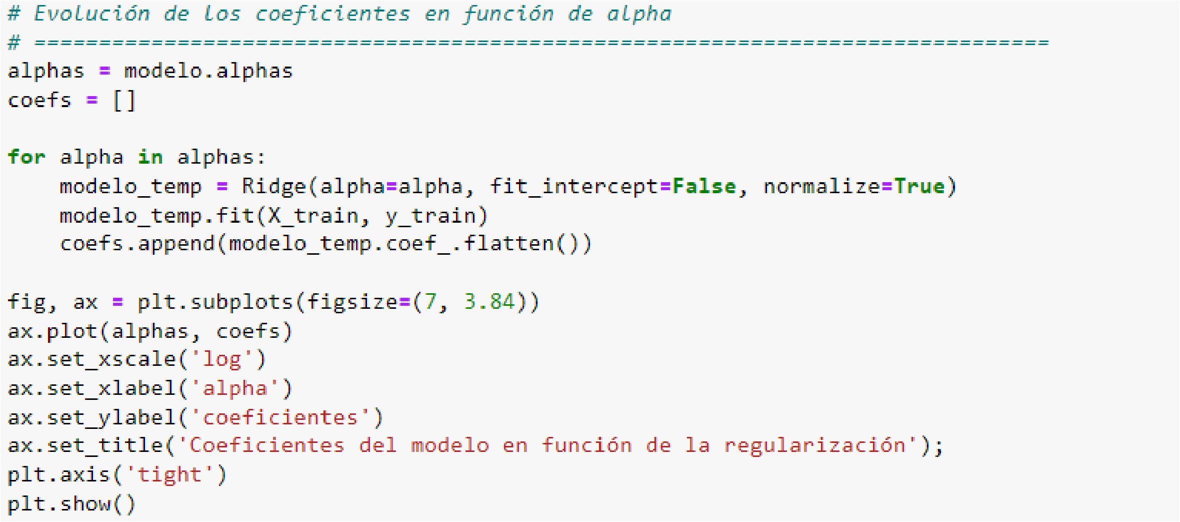
```
# Error de test del modelo
# ==============================================
predicciones = model.predict(X = X_test)
print(predicciones[0:3,])

rmse = mean_squared_error(
        y_true  = y_test,
        y_pred  = predicciones,
        squared = False
       )
print("")
print(f"El error (rmse) de test es: {rmse}")
```

And later we generate de RidgeCv that is the lineal ridge model



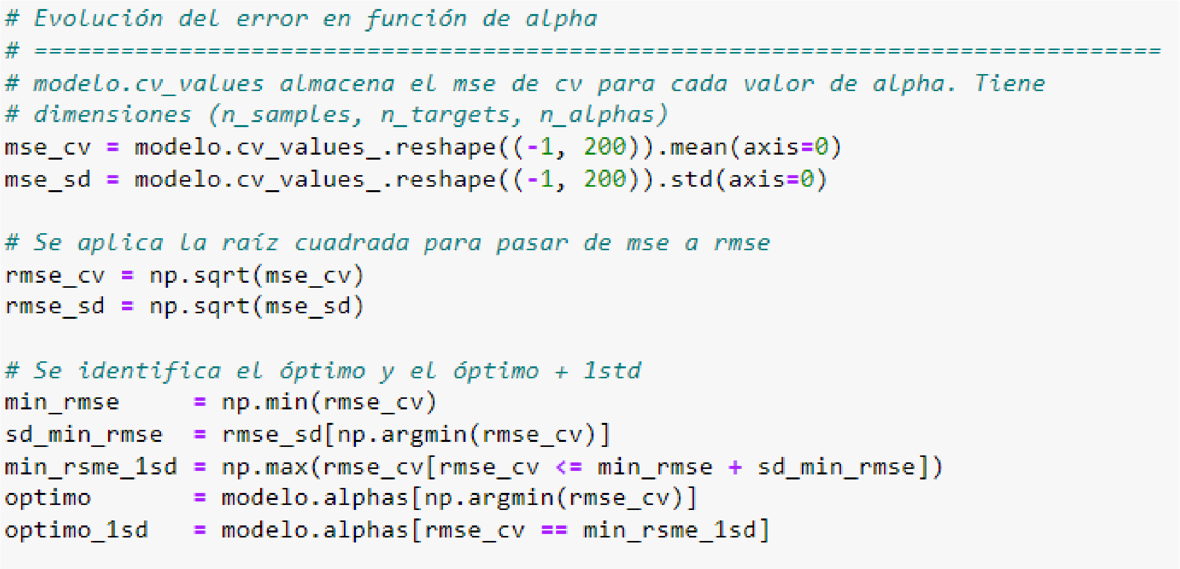
```
# Creación y entrenamiento del modelo (con búsqueda por CV del valor óptimo alpha)
# ==============================================================================
# Por defecto RidgeCV utiliza el mean squared error
modelo = RidgeCV(
            alphas          = np.logspace(-10, 2, 200),
            fit_intercept   = True,
            normalize       = True,
            store_cv_values = True
         )

_ = modelo.fit(X = X_train, y = y_train)
```

This is generated with the logspace so the different alphas can be tested



```
# Evolución de los coeficientes en función de alpha
# ==============================================================================
alphas = modelo.alphas
coefs = []

for alpha in alphas:
    modelo_temp = Ridge(alpha=alpha, fit_intercept=False, normalize=True)
    modelo_temp.fit(X_train, y_train)
    coefs.append(modelo_temp.coef_.flatten())

fig, ax = plt.subplots(figsize=(7, 3.84))
ax.plot(alphas, coefs)
ax.set_xscale('log')
ax.set_xlabel('alpha')
ax.set_ylabel('coeficientes')
ax.set_title('Coeficientes del modelo en función de la regularización');
plt.axis('tight')
plt.show()
```

In this you get all the alphas with. alphas and they are all tested in the for generating the model for each one and the coefficients are shown and then these are graphed to be able to see the evolution of the coefficients according to the evolution of the alphas



```
# Evolución del error en función de alpha
# ==============================================================================
# modelo.cv_values almacena el mse de cv para cada valor de alpha. Tiene
# dimensiones (n_samples, n_targets, n_alphas)
mse_cv = modelo.cv_values_.reshape((-1, 200)).mean(axis=0)
mse_sd = modelo.cv_values_.reshape((-1, 200)).std(axis=0)

# Se aplica la raíz cuadrada para pasar de mse a rmse
rmse_cv = np.sqrt(mse_cv)
rmse_sd = np.sqrt(mse_sd)

# Se identifica el óptimo y el óptimo + 1std
min_rmse     = np.min(rmse_cv)
sd_min_rmse  = rmse_sd[np.argmin(rmse_cv)]
min_rsme_1sd = np.max(rmse_cv[rmse_cv <= min_rmse + sd_min_rmse])
optimo       = modelo.alphas[np.argmin(rmse_cv)]
optimo_1sd   = modelo.alphas[rmse_cv == min_rsme_1sd]
```

Then the alphas are tested by comparing the errors of the mse and the rmse with the cv_value functions, obtaining the optimal alpha and then this is applied the same as in the normal one, that of creating the model, reviewing the coefs and the results of errors.

lasso

The same codes are repeated

(The codes are based on Joaquín Amat Rodrigo page of on the cienciadedatos.net)

And then later we do the interface for the user to make it easier for the user to know what their possible salary is, what is done is to have the 7 questions and the corresponding options for those questions as in skill a yes or no to know if the user has these skills to know what rating the company wants the type of data scientist the work sector Then within the quiz as such in the display result function different ifs are applied to fill out a list and then with this it predicts the salary.

Results

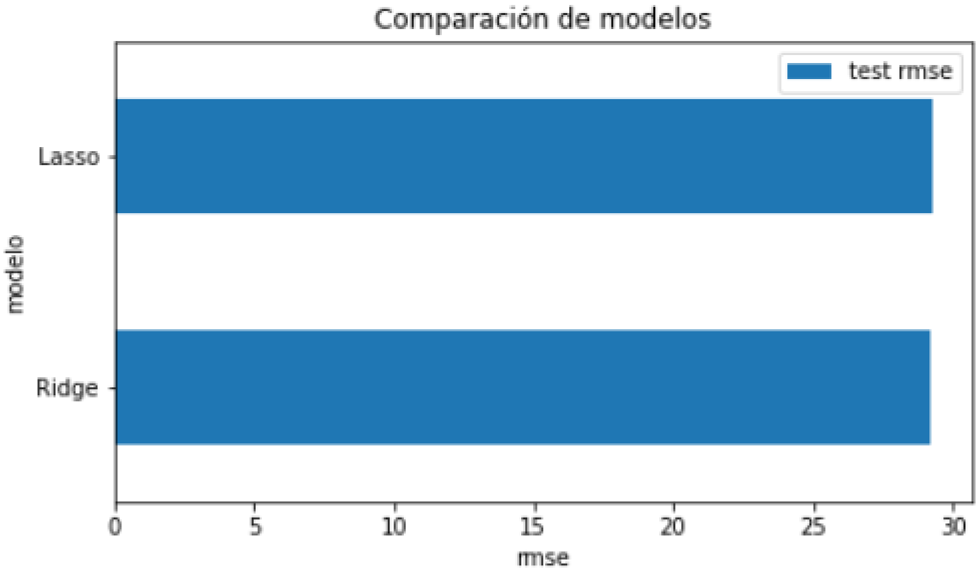
First, for the creation of the models, two assumptions were seen with the relationship of the variables, it is seen that there is no collinearity and then the linearity of the var dependent on the average salary, none of these Pearson relationships are greater than 0.5, therefore the model is not expected to be a good model

model results can be viewed for the

linear regression model

Determination coefficient R^2: 0.35825165237800705
The test error (rmse) is: 67704219917655.31
The Ridge Linear Regression Model
Determination coefficient R^2: 0.3799397683604361
The test error (rmse) is: 29.181519562785425

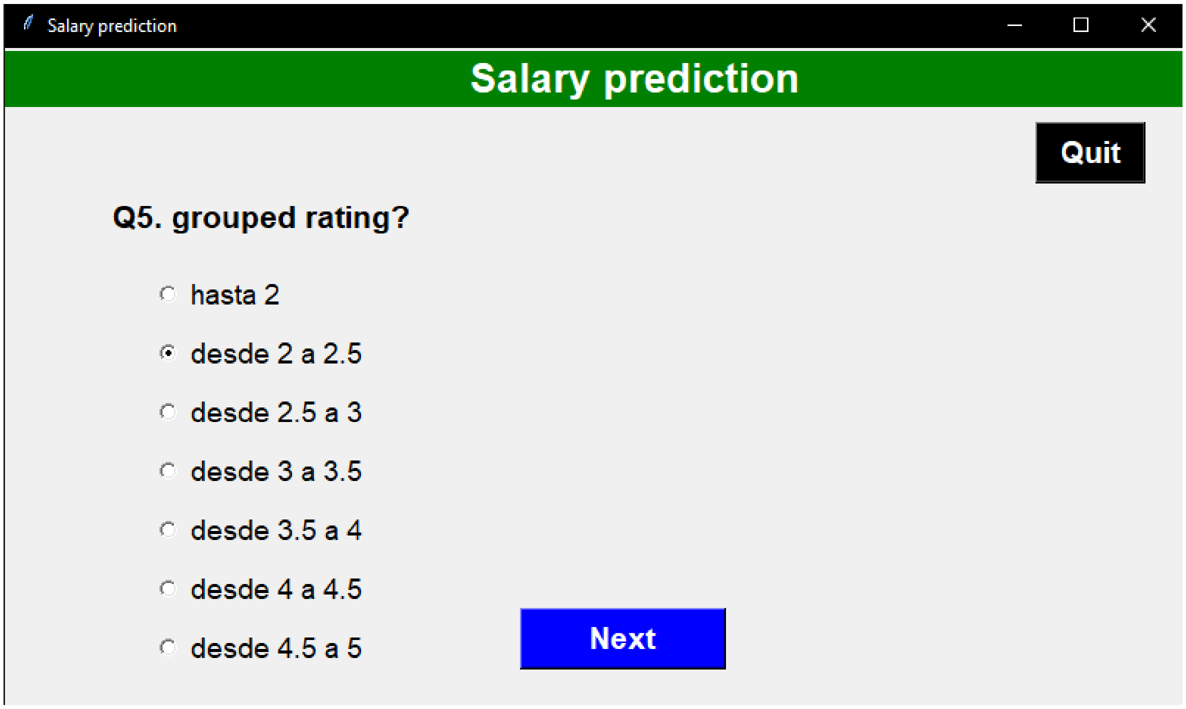
The Lasso Linear Regression Model
Determination coefficient R^2: 0.36689181218977196
The test error (rmse) is: 29.259519862616003



With these results, it can be analyzed that the r-squared fit is very low for the 3, none being in the range of 0.5 to 1, which is the most acceptable for the fit of the models so that they predict well and the sum of the mean errors is in the normal linear model is 67704219917655 which is a lot that can be generated by some problem in the data frame then in ridge and lasso are more understandable errors of 29.181519562785425 and 29.259519862616003 respectively very similar but still a bad number that means that in On average, the problem has an approximate error of 30,000 dollars per year, which would not serve to have it as a model to predict the salary of these people in a good way. This was already assumed as a hypothesis due to the little linear relationship that the independent variables had with salary.

Therefore, it would be expected to test with other types of machine learning models to find the best model and be able to give the best salary prediction

For the user interface, the lasso model is used because of the similar result of the rmse with ridge and itself, since it has fewer coefficients and only takes the important variables.



Q5. grouped rating?

○ hasta 2

● desde 2 a 2.5

○ desde 2.5 a 3

○ desde 3 a 3.5

○ desde 3.5 a 4

○ desde 4 a 4.5

○ desde 4.5 a 5

Bibliography

Foreign code:

https://www.kaggle.com/code/rucheiitr/data-scientist-salary-prediction

https://www.geeksforgeeks.org/python-mcq-quiz-game-using-tkinter/

Machine learning and sckit learning

https://www.cienciadedatos.net/documentos/py14-ridge-lasso-elastic-net-python.html

https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression-and-classification

https://www.cienciadedatos.net/documentos/py10-regresion-lineal-python.html

https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html