396 1570	10944 Male 35 20000 0 68575 Female 26 43000 0 03246 Female 27 57000 0 04002 Male 19 76000 0 91863 Female 46 41000 1 06071 Male 51 23000 1 54296 Female 50 20000 1
: #gives sns.hea	
Estimat	User ID - 1 -0.00072 0.071 0.0071 - 0.8 Age0.00072 1 0.16 0.62 - 0.6 sedSalary - 0.071 0.16 1 0.36 - 0.4
P	redSalary - 0.071
1.58 1.57 1.57 1.56 1.56	75 - 70 - 75 - 75 - 75 - 75 - 75 - 75 -
Age	
15000 12500 10000 7500 5000	
- 2500 1 0 0	
<pre>X = data y = data : # Split</pre>	1.56 1.57 1.58 20 40 60 50000 100000 150000 0.0 0.5 EstimatedSalary Purchased aset.iloc[:, [2, 3]].values aset.iloc[:, 4].values ting the dataset into the Training set and Test set learn.model selection import train test split
<pre>X_train # Featu from sk sc = St X_train X_test :</pre>	<pre>, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0) re Scaling learn.preprocessing import StandardScaler andardScaler() = sc.fit_transform(X_train) = sc.transform(X_test)</pre>
# Fitti from sk classif classif	<pre>stic Regression ng Logistic Regression to the Training set learn.linear_model import LogisticRegression ier = LogisticRegression(random_state = 0) ier.fit(X_train, y_train) LogisticRegression .cRegression(random_state=0)</pre>
y_pred: y_pred: array([<pre>cting the Test set results = classifier.predict(X_test) 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1]) g the Confusion Matrix</pre>
<pre>cm = cor cm : array([: # Visua from mar X_set, y</pre>	<pre>learn.metrics import confusion_matrix nfusion_matrix(y_test, y_pred) [65, 3], [8, 24]]) lising the Training set results tplotlib.colors import ListedColormap y_set = X_train, y_train = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
plt.xlii plt.ylii for i, plt plt.tit plt.xla	<pre>np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.0 tourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel()]).T).reshape(X1.shape),</pre>
ve precent	ument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mappin edence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or py with a single row if you intend to specify the same RGB or RGBA value for all points. ument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mappin edence in case its length matches with *x* & *y*. Please use the *color* keyword-argument or py with a single row if you intend to specify the same RGB or RGBA value for all points. Logistic Regression (Training set)
Estimated Salary 0 1 0	
	lising the Test set results
X_set, y X1, X2 : plt.com plt.xlin plt.ylin for i, plt plt.tit	<pre>tplotlib.colors import ListedColormap y_set = X_test, y_test = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
plt.xlai plt.ylai plt.leg plt.show *c* arg ve prece 2D arra *c* arg ve prece 2D arra	bel('Age') bel('Estimated Salary') end()
Estimated Salary O 1 C	
- 1 -	-2 -1 0 1 2 3 Age
precision print(pr	lo de precisión del modelo learn.metrics import precision_score on = precision_score(y_test, y_pred) Precisión del modelo:") recision) ón del modelo: 888888888888888888888888888888888888
from skine exactition print ("I print (exactition of the control o	learn.metrics import accuracy_score ud = accuracy_score(y_test, y_pred) Exactitud del modelo:") xactitud) ud del modelo: lar la sensibilidad del modelo learn.metrics import recall_score
sensibi print("" print(so Sensibi 0.75 # Calcu from sk puntaje print("	<pre>lidad = recall_score(y_test, y_pred) Sensibilidad del modelo") ensibilidad) lidad del modelo lo el puntaje F1 del modelo learn.metrics import f1_score f1 = f1_score(y_test, y_pred) Puntaje F1 del modelo")</pre>
print("print(print	<pre>Puntaje F1 del modelo") untajef1) F1 del modelo 93220338982 lo la curva ROC -AUC del modelo learn.metrics import roc_auc_score = roc_auc_score(y_test, y_pred) Curva ROC - AUC del modelo:") oc_auc) OC - AUC del modelo:</pre>
WNN # Fitti from sk classif.	11764705883
Predi y_pred: y_pred array([hborsClassifier() cting the Test set results = classifier.predict(X_test) 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
<pre># Making from sk cm = coo cm array([</pre>	0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1]) g the Confusion Matrix learn.metrics import confusion_matrix nfusion_matrix(y_test, y_pred) [64, 4], [3, 29]])
from max X_set, X1, X2 plt.complt.xlinplt.ylinfor i, plt.	<pre>lising the Training set results tplotlib.colors import ListedColormap y_set = X_train, y_train = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
plt.xlal plt.ylal plt.leg plt.show *c* arg ve prece 2D arra *c* arg ve prece	
3 - 2 - 1 -	K-NN (Training set)
ed S.	
Estimated Salary	
# Visua from max X_set, x1, x2 plt.complt.xliiplt.ylii	<pre>lising the Test set results tplotlib.colors import ListedColormap y_set = X_test, y_test = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
# Visua from max X_set, X1, X2 = plt.complt.xliiplt.yliifor i, plt.vliiplt.ylaplt.legplt.show *c* arg ve precedure	<pre>lising the Test set results tplotlib.colors import ListedColormap y_set = X_test, y_test = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
# Visua from max X_set, X1, X2 = plt.complt.xlin plt.ylin for i, plt.ylin plt.ylaplt.leg plt.show *c* argive precedure array ve	<pre>lising the Test set results tplotlib.colors import ListedColormap y_set = X_test, y_test = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
# Visua from max X_set, X1, X2 plt.complt.xlin plt.ylin for i, plt plt.tit plt.xlal plt.ylal plt.leg plt.show *c* arg ve prece 2D array ve	<pre>lising the Test set results tplotlib.colors import ListedColormap y_set = X_test, y_test = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0</pre>
# Visua from max X set, X1, X2 plt.com plt.xlin plt.ylin for i, plt.ylin pl	Issing the Test set results tplotlib.colors import ListedColormap y_set = X_test, y_test enp.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0 np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.0 tourf(X1, X2, classifier.predict(np.array([X1.ravel()], X2.ravel()]).T).reshape(X1.shape), alpha = 0.75, cnap = ListedColormap(('red', 'green'))) in enumerate(np.unique(y_set)): scatter(X_set(y_set = i, 0], X_set(y_set = i, 1], c = ListedColormap(('red', 'green'))(i), label = j) bel('Age') bel('Age') bel('Astimated Salary') end() w(') ument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mappin edence in case its length matches with ** \$ *y*. Please use the *color* keyword-argument or p y with a single row if you intend to specify the same RGB or RGBA value for all points. when the color keyword argument or p y with a single row if you intend to specify the same RGB or RGBA value for all points. K-NN (Test set) **O ** **O
# Visua from max X_set, X1, X2 = plt.com plt.xlin plt.ylin for i, plt plt.tit plt.xlan plt.leg plt.show *c* arg ve prece 2D array *c* array	<pre>lising the Test set results tplotlib.colors import ListedColormap y_set = X_test _ y_test = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.0 np.arange(start = X_set[:, 1].min() - 1, stop = X_set[:, 1].max() + 1, step = 0.0 tourf(X1, X2, classifier.predict(np.array[X1.ravel(), X2.ravel()]).T).reshape(X1.shape),</pre>
# Visua from max X set, X1, X2 plt.com plt.xlin plt.ylin for i, plt plt.tit plt.xlal plt.leg plt.show *c* arg ve precedurate ar	Lising the Test set results rplotlib.colors import ListedColormap yet = X test, ytest - np.meshprid(np.arongelstart = X.set[:, 0].min() - 1, stop = X.set[:, 0].max() + 1, stop = 0.0 - np.meshprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.meshprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.meshprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.meshprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.mexhprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.mexhprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.mexhprid(np.arongelstart = X.set[:, 1].min() - 1, stop = X.set[:, 1].max() + 1, stop = 0.0 - np.mexhproperson - np.mexhprope
# Visua from ma X_set, X1, X2 plt.com plt.xli plt.yli for i, plt.yli plt.yla plt.leg plt.sho *c* arg ve prece 2D arra *c* arg ve prece 2D arra *c* arg ve prece 2D arra *c*	Age listing the Test set results liptofile.colors import listedColormap y set = X test, y test proportion armogistant = X set[: 0] win() = 1, stop = X set[. 0] mos() + 1, stop = 0.0 sphession armogistant = X set[: 0] win() = 1, stop = X set[. 1] mos() + 1, stop = 0.0 sphession armogistant = X set[: 0] win() = 1, stop = X set[. 1] mos() + 1, stop = 0.0 sphession armogistant = X set[: 0] win() = 1, stop = 0.0 sphession armogistant = X set[: 0] win() = 1, stop = 0.0 sphession armogistant = X set[: 0] win() = 1, stop = 0.0 sphession armogistant = 1, st
# Calcument of the service of the se	Age Its interpretation of another Listed Colormap Jest & Licat, yield representation of another listed Colormap Jest & Licat, yield representation of another listed Colormap Jest & Licat, yield representation of another listed Colormap Jest & Licat, yield representation of another listed Colormap Jest & Licat, yield Jest Age (Listed Colormap) Jest & Licat, X.J. (Lassifier prediction correy (RLI revel()), X.J. revel() 11, x.J. respace(XLI shape), Jest Age (Listed Colormap) Jest & Listed Colormap Jest & Listed Colormap Jest & Jest & Listed Colormap Jest & Jest & Listed Colormap Jest & Liste
# Visua from max X_set, X1, X2 plt.com plt.xlin plt.xlap plt.show *c* arguve preceded print(prin	Age sage the Test set results uptoclib.colors import ListedColormag **pureStrib.colors import ListedColormag **pureStrib.colors import ListedColormag **pureStrib.colors.colormag **pureStrib.colormag **pureStrib.co
# Visual from max X_set, X1, X2: plt.com plt.xlii plt.ylii for i, plt.ylii plt.show *c* arg ve prece 2D arra *c* arg ve prece 2D arra *c* arg ve prece 2D arra *c* arg ve print(p Precision 9.87878 # Calcu from sk exactit print(p Precision p	Age Using the Fest set results priorith-colors import instendiolomag representations separate instendiolomag representations presentations separate instendiolomag representations accommendates = X_sect[. 0].min() - 1, stop = X_sect[. 0].max() + 1, stop = 0.0 Local colors of the second separate instendiolomage (red = green i)) Local colors = 0.0, roop = intendiolomage(red = green i); Local colors = 0.0, roop = intendiolomage(red = green i); Local colors = intendiolomage(red = green
# Visua from max X_set, X1, X2 plt.com plt.xlin plt.yla plt.leg plt.show recisin print(" print	Age Interest that There exists: Interest the Terrest restarts: Interest the Terrest restarts: Interest the Terrest restarts: Interest the Terrest restarts: Interest restarts: Inte
# Calcument of the print of the	Target the sent of continues and continues property of the continues and
# Calcular for in in its plt. show the strength of the strengt	The process of the control of the co
# Calcular from sk precision of the state of	The property of the content of the property of the
# Visua from max X_1, X_2 in plt. xling plt.	The productions bear instead commands of the productions of the productions of the productions of the productions of the production of the pr
# Visua from max X_1, X2 in plt. xling plt.	Table of the part involved product involved transcriptors of selection and product involved product involved transcriptors of selection and product involved transcriptors of
# Visua from max X1, X2 plt. show sex arguments from sk precision print (print	Table of the part involved product involved transcriptors of selection and product involved product involved transcriptors of selection and product involved transcriptors of
# Calculum print (print	Design the case of control con
# Comparison of the print of th	Foreign day the control of the contr
# Calcular of the strength of	Service of the property of the
# Visua from sk composition of the composition of t	See the property of the proper
# Visual # Calcu #	Age The control of t
# Visua # V	
# Visua * V	Section 19 Comments of the control o
# Visua # Calcu # C	The control of the co
# Visua # V	Section 2012 And Sectio
# Visua # Fitti Form sk # Calcu # Fitti # Fitti # Calcu # Fitti # Fitti # Calcu # Fitti # Calcu # Fitti # Fitti # Calcu # Fitti # Fitti # Calcu # Fitti # Calcu # Fitti # Calcu # Fitti # Fitti # Calcu # Fitti # C	The second of the control of the con
Visua from sk, potential # Visua from sk, XI, X2 plt. Sti print(s) # Calcu from sk exactit print(s) # Calcu from sk classif # Visua from sk classif print(s) plt. Sti print(s) print(The second control of
# Calcuments #	
# Visua # V	
# Visua # V	

<pre>from matplotl X_set, y_set X1, X2 = np.m plt.contourf(plt.xlim(X1.m plt.ylim(X2.m for i, j in e plt.scatt plt.title('De plt.xlabel('A plt.ylabel('E plt.legend() plt.show() *c* argument ve precedence</pre>	looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping wie in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provi
c argument ve precedence 2D array with 3 - 2 -	a single row if you intend to specify the same RGB or RGBA value for all points. looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping with in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provious a single row if you intend to specify the same RGB or RGBA value for all points. Decision Tree Classification (Training set)
<pre>from matplotl X_set, y_set X1, X2 = np.m</pre>	-2 -1 0 1 2 3 Age **the Test set results** ib.colors import ListedColormap = X_test, y_test eshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),
<pre>plt.xlim(X1.m plt.ylim(X2.m for i, j in e plt.scatt plt.title('De plt.xlabel('A plt.ylabel('E plt.legend() plt.show() *c* argument ve precedence</pre>	<pre>alpha = 0.75, cmap = ListedColormap(('red', 'green'))) in(), X1.max()) in(), X2.max()) inumerate(np.unique(y_set)): ier(X_set[y_set == j, 0], X_set[y_set == j, 1],</pre>
c argument ve precedence 2D array with 3 -	looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping with in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provious a single row if you intend to specify the same RGB or RGBA value for all points. Decision Tree Classification (Test set)
Estimated Salary	-2 -1 0 1 2 3
<pre>from sklearn. precision = p print("Precisi print(precisi) Precisión del 0.82857142857 # calculo par from sklearn.</pre>	modelo: 14286 ra la exactitud del modelo metrics import accuracy_score
<pre>print("Exacti print(exactit Exactitud del 0.91 Randon # Fitting Ran from sklearn. classifier =</pre>	Forest Classification Indom Forest Classification to the Training set ensemble import RandomForestClassifier RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
classifier.fi RandomForest # Predicting y_pred = clas y_pred array([0, 0, 0, 1, 1, 0,	RandomForestClassifier Classifier(criterion='entropy', n_estimators=10, random_state=0) the Test set results sifier.predict(X_test) 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
<pre>1, 0, # Making the from sklearn. cm = confusio cm : array([[63,</pre>	0, 1, 0, 0, 0, 0, 0, 1, 1, 1]) Confusion Matrix metrics import confusion_matrix n_matrix(y_test, y_pred) 5],
<pre>plt.xlim(X1.m plt.ylim(X2.m for i, j in e plt.scatt plt.title('Ra plt.xlabel('A plt.ylabel('E plt.legend() plt.show() *c* argument ve precedence</pre>	looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping wie in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provi
c argument ve precedence 2D array with 3 -	a single row if you intend to specify the same RGB or RGBA value for all points. looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping wie in case its length matches with *x* & *y*. Please use the *color* keyword-argument or provious a single row if you intend to specify the same RGB or RGBA value for all points. Random Forest Classification (Training set)
Estimated Salary	
<pre>from matplotl X_set, y_set X1, X2 = np.m plt.contourf(plt.xlim(X1.m plt.ylim(X2.m for i, j in e plt.scatt</pre>	<pre>Age I the Test set results ib.colors import ListedColormap = X_test, y_test ibshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:, 0].max() + 1, step = 0.01),</pre>
plt.xlabel('A plt.ylabel('E plt.legend() plt.show() *c* argument ve precedence 2D array with *c* argument ve precedence	
Estimated Salary	
<pre>from sklearn. precision = p</pre>	precisión del modelo metrics import precision_score precision_score(y_test, y_pred) ión del modelo:")
<pre>print(precisi Precisión del 0.84848484848 # calculo par from sklearn. exactitud = a print("Exacti print(exactit Exactitud del</pre>	modelo: 48485 Ta la exactitud del modelo metrics import accuracy_score accuracy_score(y_test, y_pred) tud del modelo:") aud)
Exactitud del 0.91 Conclusión	