In [97]:

```
import numpy as np
import pandas as pd
df = pd.read_csv(r'speeddating.csv',encoding='ISO-8859-1')
df.head()
```

```
C:\Users\thanigaiharini.s\AppData\Local\Continuum\anaconda3\lib\site-package
s\IPython\core\interactiveshell.py:3326: DtypeWarning: Columns (4,11,12,16,1
7,18,19,20,40,41,42,43,44,45,52,53,54,55,56,74,75,76,77,78,79,80,81,82,83,8
4,85,86,87,88,89,90,108,110) have mixed types.Specify dtype option on import
or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Out[97]:

	id	has_null	wave	gender	age	age_o	d_age	d_d_age	race	race_o
0	1	0	1	female	21	27	6	[4-6]	Asian/Pacific Islander/Asian-American	European/Caucasian-Americar
1	2	0	1	female	21	22	1	[0-1]	Asian/Pacific Islander/Asian-American	European/Caucasian-Americar
2	3	1	1	female	21	22	1	[0-1]	Asian/Pacific Islander/Asian-American	Asian/Pacifi Islander/Asian Americar
3	4	0	1	female	21	23	2	[2-3]	Asian/Pacific Islander/Asian-American	European/Caucasian-Americar
4	5	0	1	female	21	24	3	[2-3]	Asian/Pacific Islander/Asian-American	Latino/Hispani Americar

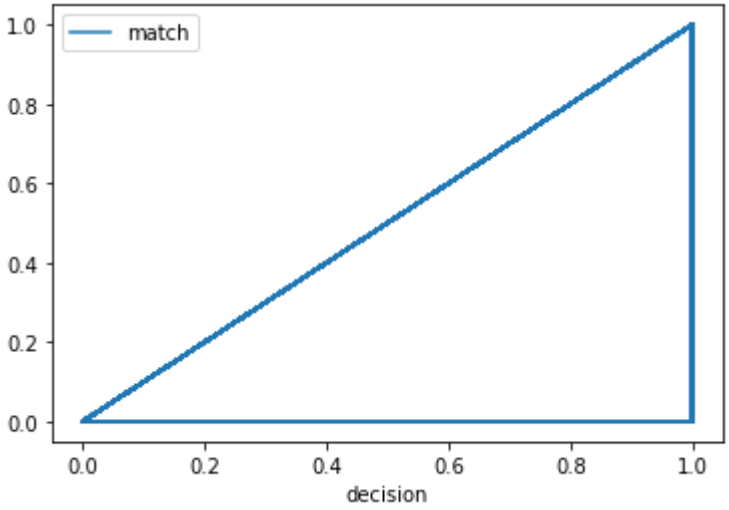
5 rows × 124 columns

In [98]:

```
df.plot(x="decision", y="match")
```

Out[98]:

<AxesSubplot:xlabel='decision'>

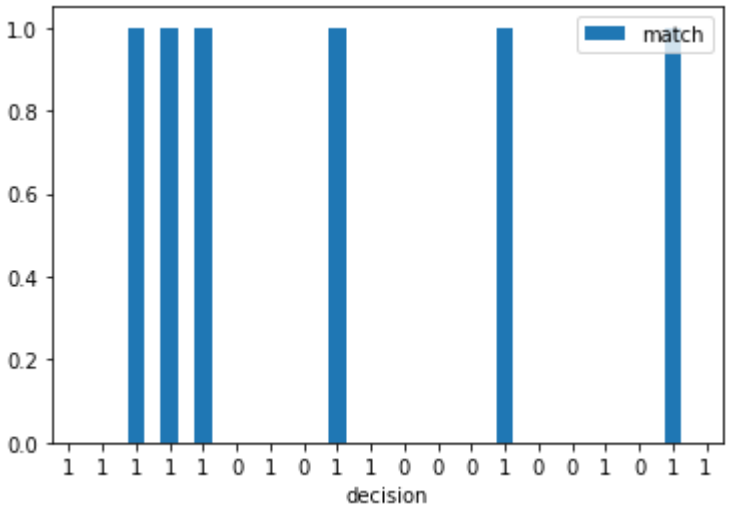


In [99]:

```
bg=df.head(20)
bg.plot.bar(x='decision', y='match', rot=0)
```

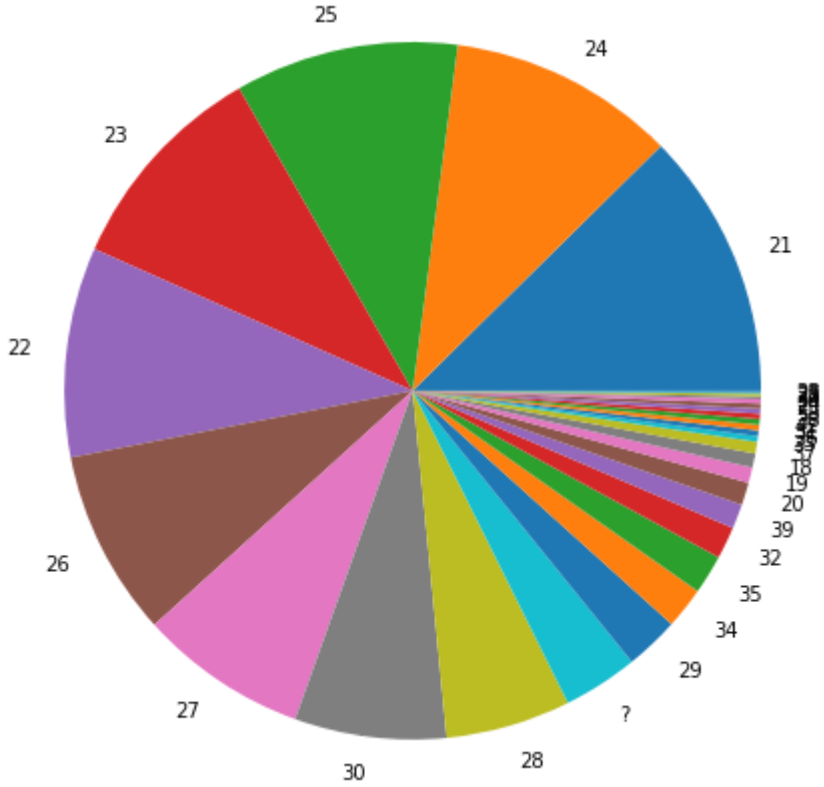
Out[99]:

<AxesSubplot:xlabel='decision'>



In [100]:

```
values =df['age'].value_counts()
labels= df['age'].unique().tolist()
plt.pie(values,labels =labels,radius =2)
plt.show()
print(values)
```




```
27    1037
23     884
26     869
24     841
25     815
28     724
22     655
29     589
30     486
21     291
32     210
33     161
34     152
31     125
?       95
30      88
35      60
```

```
20      55
36      45
24      22
28      22
27      22
25      22
19      20
42      20
38      19
39      18
18      10
23      10
55       6
37       5
Name: age, dtype: int64
```

In [138]:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
X = df.iloc[:,-1]
y = df.iloc[:, -3]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)
```

In [139]:

```
X_test=X_test.values.reshape(-1,1)
X_train=X_train.values.reshape(-1,1)
```

In [140]:

```
X_train
```

Out[140]:

```
array([[0],
       [0],
       [0],
       ...,
       [0],
       [0],
       [0]], dtype=int64)
```

In [141]:

```
y_train
```

Out[141]:

```
3545    0
3132    0
5432    1
6184    1
1949    1
       ..
4373    1
7891    0
4859    0
3264    0
2732    0
Name: decision, Length: 5585, dtype: int64
```

In [142]:

```
X_test
```

Out[142]:

```
array([[0],
       [0],
       [0],
       ...,
       [0],
       [1],
       [0]], dtype=int64)
```

In [143]:

```
y_test
```

Out[143]:

```
2265    0
2851    0
3655    0
196     0
3719    0
       ..
2696    0
2126    0
282     0
6512    1
2448    0
Name: decision, Length: 2793, dtype: int64
```

In [144]:

```
#Decision Tree
from sklearn import preprocessing
from sklearn import utils
lab = preprocessing.LabelEncoder()
y_transformed = lab.fit_transform(y_train)
print(y_transformed)
```

```
[0 0 1 ... 0 0 0]
```

In [145]:

```
from sklearn import preprocessing
from sklearn import utils
lab = preprocessing.LabelEncoder()
ytest_transformed = lab.fit_transform(y_test)
print(ytest_transformed)
```

```
[0 0 0 ... 0 1 0]
```

In [146]:

```
from sklearn import preprocessing
from sklearn import utils
#convert y values to categorical values
lab = preprocessing.LabelEncoder()
ytest_transformed = lab.fit_transform(y_test)
#view transformed values
print(ytest_transformed)
```

```
[0 0 0 ... 0 1 0]
```

In [147]:

```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(criterion = 'entropy')
```

In [148]:

```
clf.fit(X_train, y_transformed)
```

```
C:\Users\thanigaiharini.s\AppData\Local\Continuum\anaconda3\lib\site-package
s\sklearn\tree\tree.py:163: DeprecationWarning: `np.int` is a deprecated ali
as for the builtin `int`. To silence this warning, use `int` by itself. Doin
g this will not modify any behavior and is safe. When replacing `np.int`, yo
u may wish to use e.g. `np.int64` or `np.int32` to specify the precision. If
you wish to review your current use, check the release note link for additio
nal information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/d
evdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/rel
ease/1.20.0-notes.html#deprecations)
  y_encoded = np.zeros(y.shape, dtype=np.int)
```

Out[148]:

```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=Non
e,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

In [149]:

```
y_pred = clf.predict(X_test)
```

In [150]:

```
from sklearn.metrics import accuracy_score
print('Accuracy Score on train data: ', accuracy_score(y_true=y_transformed, y_pred=clf.pre
print('Accuracy Score on test data: ', accuracy_score(y_true=ytest_transformed, y_pred=y_pr
```

```
Accuracy Score on train data:  0.7416293643688451
Accuracy Score on test data:  0.7511636233440745
```

In [151]:

```
from sklearn import svm
from sklearn import metrics
```

In [152]:

```
clf = svm.SVC(kernel='rbf')
```

In [153]:

```
clf.fit(X_train,y_train)
```

```
C:\Users\thanigaiharini.s\AppData\Local\Continuum\anaconda3\lib\site-package
s\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will ch
ange from 'auto' to 'scale' in version 0.22 to account better for unscaled f
eatures. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

Out[153]:

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

In [154]:

```
y_pr = clf.predict(X_test)
print(y_pr)
```

```
[0 0 0 ... 0 1 0]
```

In [155]:

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pr))
```

```
Accuracy: 0.7511636233440745
```

In [156]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

In [157]:

```
final_model = LogisticRegression(C=1,solver='lbfgs',multi_class='auto')
final_model.fit(X_train, y_train)
print ("Final Accuracy: %s"
 % accuracy_score(y_train, final_model.predict(X_train)))
```

```
Final Accuracy: 0.7416293643688451

C:\Users\thanigaiharini.s\AppData\Local\Continuum\anaconda3\lib\site-package
s\sklearn\linear_model\base.py:291: DeprecationWarning: `np.int` is a deprec
ated alias for the builtin `int`. To silence this warning, use `int` by itse
lf. Doing this will not modify any behavior and is safe. When replacing `np.
int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the preci
sion. If you wish to review your current use, check the release note link fo
r additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/d
evdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/rel
ease/1.20.0-notes.html#deprecations)
  indices = (scores > 0).astype(np.int)
```

In [158]:

```
from sklearn import linear_model
from sklearn.metrics import mean_squared_error,confusion_matrix, precision_score, auc,roc_c
from sklearn import ensemble, linear_model, neighbors, svm, tree, neural_network
```

In [159]:

```
MLA = [
    linear_model.LogisticRegressionCV(),
    svm.SVC(probability=True),
    tree.DecisionTreeClassifier(),
]
```

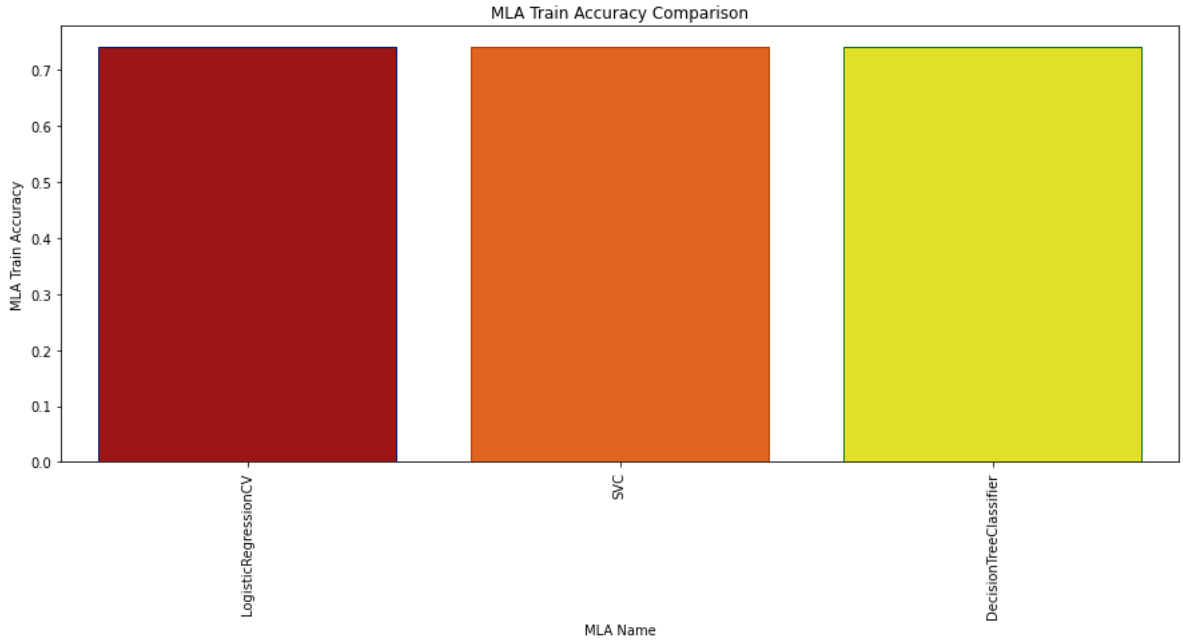
In [160]:

```
MLA_columns = []
MLA_compare = pd.DataFrame(columns = MLA_columns)
row_index = 0
for alg in MLA:
    predicted = alg.fit(X_train, y_train).predict(X_test)
    fp, tp, th = roc_curve(y_test, predicted)
    MLA_name = alg.__class__.__name__
    MLA_compare.loc[row_index,'MLA Name'] = MLA_name
    MLA_compare.loc[row_index, 'MLA Train Accuracy'] = round(alg.score(X_train, y_train), 4
    MLA_compare.loc[row_index, 'MLA Test Accuracy'] = round(alg.score(X_test, y_test), 4)
    MLA_compare.loc[row_index, 'MLA Precission'] = precision_score(y_test, predicted)
    MLA_compare.loc[row_index, 'MLA Recall'] = recall_score(y_test, predicted)
    MLA_compare.loc[row_index, 'MLA AUC'] = auc(fp, tp)
    row_index+=1
MLA_compare.sort_values(by = ['MLA Test Accuracy'], ascending = False, inplace = True)
MLA_compare
```

```
 deprecated alias for the builtin `int`. To silence this warning, use `in
t` by itself. Doing this will not modify any behavior and is safe. When r
eplacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to s
pecify the precision. If you wish to review your current use, check the r
elease note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdo
cs/release/1.20.0-notes.html#deprecations)
  indices = (scores > 0).astype(np.int)
C:\Users\thanigaiharini.s\AppData\Local\Continuum\anaconda3\lib\site-pack
ages\sklearn\linear_model\base.py:291: DeprecationWarning: `np.int` is a
 deprecated alias for the builtin `int`. To silence this warning, use `in
t` by itself. Doing this will not modify any behavior and is safe. When r
eplacing `np.int`, you may wish to use e.g. `np.int64` or `np.int32` to s
pecify the precision. If you wish to review your current use, check the r
elease note link for additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdo
cs/release/1.20.0-notes.html#deprecations)
  indices = (scores > 0).astype(np.int)
```

In [161]:

```
import seaborn as sns
plt.subplots(figsize=(15,6))
sns.barplot(x="MLA Name", y="MLA Train Accuracy",data=MLA_compare,palette='hot',edgecolor=s
plt.xticks(rotation=90)
plt.title('MLA Train Accuracy Comparison')
plt.show()
```



In [162]:

```
import pickle
```

In [163]:

```
with open('model_pkl', 'wb') as files:
    pickle.dump(final_model, files)
```

In [164]:

```
with open('model_pkl' , 'rb') as f:
    lr = pickle.load(f)
```

In [165]:

```
y_pr=lr.predict(X_test)
```

```
C:\Users\thanigaiharini.s\AppData\Local\Continuum\anaconda3\lib\site-package
s\sklearn\linear_model\base.py:291: DeprecationWarning: `np.int` is a deprec
ated alias for the builtin `int`. To silence this warning, use `int` by itse
lf. Doing this will not modify any behavior and is safe. When replacing `np.
int`, you may wish to use e.g. `np.int64` or `np.int32` to specify the preci
sion. If you wish to review your current use, check the release note link fo
r additional information.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/d
evdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdocs/rel
ease/1.20.0-notes.html#deprecations)
  indices = (scores > 0).astype(np.int)
```

In [166]:

```
y_pr
```

Out[166]:

```
array([0, 0, 0, ..., 0, 1, 0], dtype=int64)
```

In [167]:

```
print("Accuracy:",metrics.accuracy_score(y_test, y_pr))
```

```
Accuracy: 0.7511636233440745
```

In [ ]:

```

```

In [ ]:

```

```