Pycaret Assignment

❖ Binary Classification

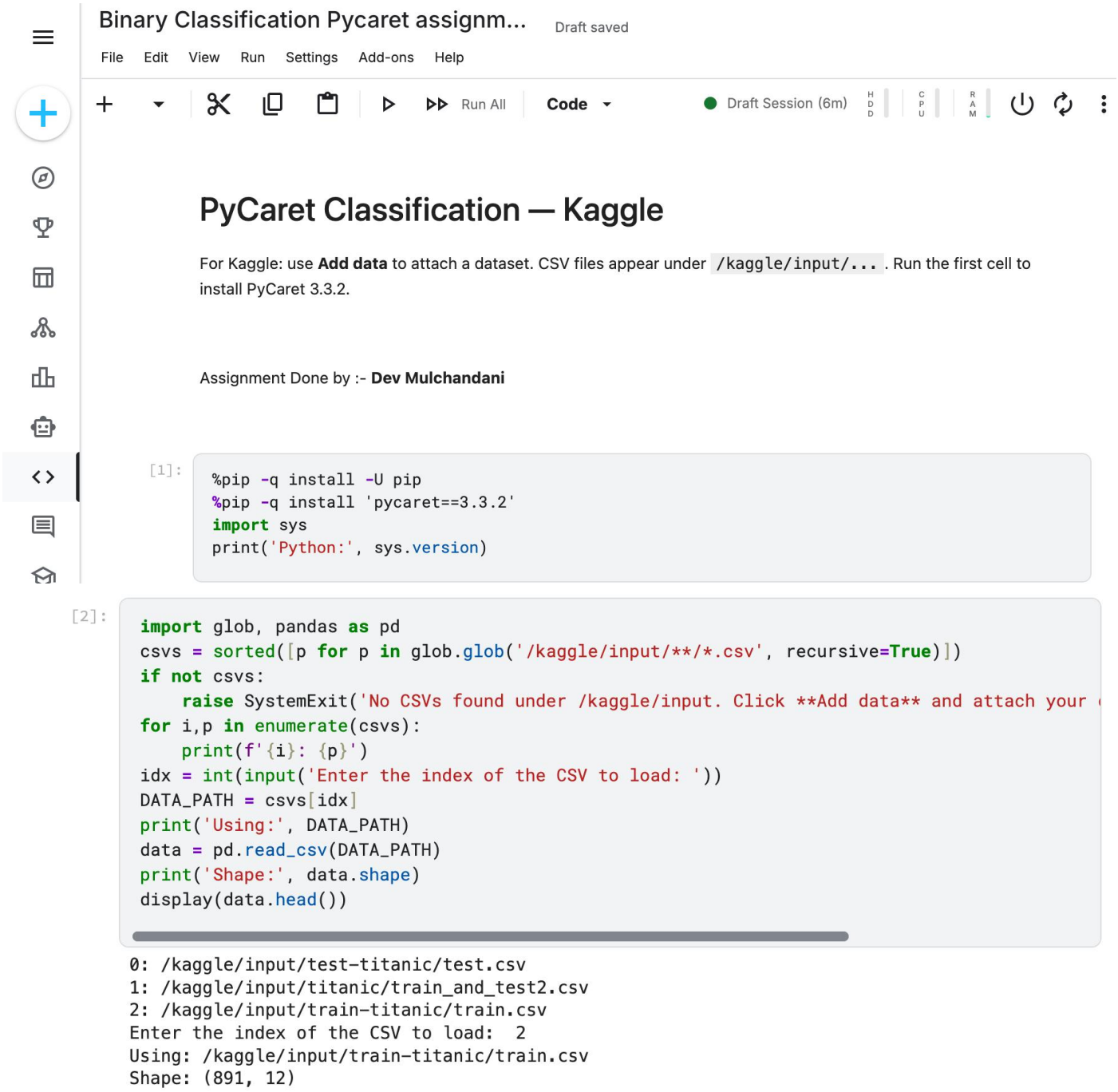
❖ Kaggle Notebook :- Link

❖ Submitted By :- Dev Mulchandani

❖ Overview :-

In my binary classification notebook, I used PyCaret's classification module to predict outcomes with two possible classes — for example, whether a passenger survived or not in the Titanic dataset. After loading and exploring the data, I initialized PyCaret using the setup() function, which automatically handled data preprocessing, encoding, and splitting. Then, I compared multiple classification models with compare_models() and selected the best one based on accuracy and F1-score. Finally, I evaluated the chosen model using visual plots, made predictions on the test data, and saved both the trained model and the results. This workflow demonstrated how binary classification can be done easily and efficiently using PyCaret with minimal code.

❖ Screenshots :-



	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
# Force CPU mode and avoid importing RAPIDS
import os
os.environ["CUDA_VISIBLE_DEVICES"] = ""  # hide GPUs from libraries

# (optional) if RAPIDS is present, uninstall so PyCaret won't try GPU backends
!pip -q uninstall -y cuml cudf cuml-cu12 cudf-cu12 cupy-cuda12x cupy-cuda11x cupy || true

# now import and continue
from pycaret.classification import *
TARGET = input('Enter target column name: ')
exp = setup(data, target=TARGET, session_id=123, use_gpu=False)
best = compare_models()
evaluate_model(best)
save_model(best, 'best_classification_model')
```

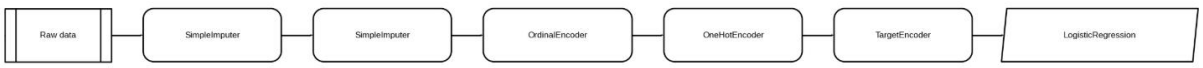
```
WARNING: Skipping cuml as it is not installed.
WARNING: Skipping cudf as it is not installed.
WARNING: Skipping cupy-cuda11x as it is not installed.
WARNING: Skipping cupy as it is not installed.
Enter target column name:  Survived
```

	Description	Value
0	Session id	123
1	Target	Survived
2	Target type	Binary
3	Original data shape	(891, 12)
4	Transformed data shape	(891, 14)
5	Transformed train set shape	(623, 14)
6	Transformed test set shape	(268, 14)
7	Numeric features	6
8	Categorical features	5
9	Rows with missing values	79.5%
10	Preprocess	True
11	Imputation type	simple
12	Numeric imputation	mean
13	Categorical imputation	mode
14	Maximum one-hot encoding	25
15	Encoding method	None
16	Fold Generator	StratifiedKFold
17	Fold Number	10
18	CPU Jobs	-1
19	Use GPU	False
20	Log Experiment	False

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.8024	0.8694	0.6697	0.7842	0.7175	0.5685	0.5761	0.8130
ridge	Ridge Classifier	0.7528	0.8647	0.4522	0.8257	0.5793	0.4273	0.4679	0.0650
et	Extra Trees Classifier	0.7400	0.7837	0.4774	0.7654	0.5815	0.4088	0.4356	0.1500
nb	Naive Bayes	0.6709	0.7925	0.1761	0.8582	0.2816	0.1808	0.2747	0.0580
knn	K Neighbors Classifier	0.6275	0.5906	0.3690	0.5175	0.4255	0.1654	0.1713	0.0700
lda	Linear Discriminant Analysis	0.6260	0.5382	0.0348	0.0800	0.0485	0.0335	0.0389	0.0590
dt	Decision Tree Classifier	0.6164	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0590
rf	Random Forest Classifier	0.6164	0.7890	0.0000	0.0000	0.0000	0.0000	0.0000	0.1770
qda	Quadratic Discriminant Analysis	0.6164	0.5386	0.0000	0.0000	0.0000	0.0000	0.0000	0.0620
ada	Ada Boost Classifier	0.6164	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0590
gbc	Gradient Boosting Classifier	0.6164	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1050
xgboost	Extreme Gradient Boosting	0.6164	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0910
lightgbm	Light Gradient Boosting Machine	0.6164	0.5183	0.0000	0.0000	0.0000	0.0000	0.0000	5.3210
catboost	CatBoost Classifier	0.6164	0.7567	0.0000	0.0000	0.0000	0.0000	0.0000	1.5940
dummy	Dummy Classifier	0.6164	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0560
svm	SVM - Linear Kernel	0.5651	0.6070	0.3915	0.5018	0.3121	0.0717	0.1066	0.0680

Plot Type:

Pipeline Plot | Hyperparameters | AUC | Confusion Matrix | Threshold
Precision Recall | Prediction Error | Class Report | Feature Selection | Learning Curve
Manifold Learning | Calibration Curve | Validation Curve | Dimensions | Feature Importance
Feature Importanc... | Decision Boundary | Lift Chart | Gain Chart | Decision Tree
KS Statistic Plot




```
Transformation Pipeline and Model Successfully Saved
```

[3]:
```
(Pipeline(memory=Memory(location=None),
          steps=[('numerical_imputer',
                  TransformerWrapper(exclude=None,
                                     include=['PassengerId', 'Pclass', 'Age',
                                              'SibSp', 'Parch', 'Fare'],
                                     transformer=SimpleImputer(add_indicator=False,
                                                               copy=True,
                                                               fill_value=None,
                                                               keep_empty_features=False,
                                                               missing_values=nan,
                                                               strategy='mean'))),
                 ('categorical_imputer',
                  TransformerWrapper(exclude=None...
                                                               handle_unknown='value',
                                                               hierarchy=None,
                                                               min_samples_leaf=20,
                                                               return_df=True,
                                                               smoothing=10,
                                                               verbose=0))),
```

[4]:

```
from pycaret.classification import load_model, predict_model
import pandas as pd

# Load your saved model
model = load_model('best_classification_model')

# Read the test dataset (from test-titanic folder)
test = pd.read_csv('/kaggle/input/test-titanic/test.csv')

# Generate predictions
preds = predict_model(model, data=test)

# Show first few rows
preds.head()
```

```
Transformation Pipeline and Model Successfully Loaded
```

[4]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	prediction_lab
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q	
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S	
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q	
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S	
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S	

[5]:

```python
# Create Kaggle submission file
submission = pd.DataFrame({
    'PassengerId': preds['PassengerId'],
    'Survived': preds['prediction_label']
})

# Save to CSV
submission.to_csv('submission.csv', index=False)

print("✅ Submission file saved as submission.csv")
submission.head()
```

```
✅ Submission file saved as submission.csv
```

[5]:

	PassengerId	Survived
0	892	0
1	893	0
2	894	0
3	895	0
4	896	1