Used Car Prediction Analysis

Michael Miranda
Roger Ramirez
Andrew Pang

CIS 5560 Intro to Big Data Science | Professor Jongwook Woo

Cal State LA

Spring 2025

Github for Code Files

https://github.com/Mgratricz/CIS-5560-Used-Cars-Prediction-Analysis

Project Summary

In recent years, rising vehicle prices, limited availability of new cars, and ongoing supply chain issues have made buying a used car a more common and complex.

Our goal is to create models, using different types of algorithms learned in class , that can estimate the price of a used car based on its individual attributes and what primary attributes greatly affect a car with an accident.

About the Dataset

Data Source: US Used Cars Dataset

- https://www.kaggle.com/datasets/ananaymital/us-used-cars-dataset
- Data Size: ~2.5GB

Dataset Preview

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	vin	back_legroo	bed	bed_height	bed_length	body_type	cabin	city	city_fuel_ec	combine_fu	daysonmarket	dealer_zip	description	engine_cylir	engine_disp	engine_type	exterior_col	fleet
2	KL7CJKSB4LB0	35.7 in				SUV / Crossover		Raynham	26		183	2767	All specials inclu	I4	1400	I4	Summit White	
3	1C4PJMDX0LD(	40.3 in				SUV / Crossover		Rochester Hills	19		130	48307	[!@@Additional	V6	3200	V6	Diamond Black Crystal Pearlcoa	
4	3N1AB7AP5KL6	37.4 in				Sedan		Tulsa	29		44	74133	CarMax Curbsid	I4	1800	I4	Gray	TRUE
5	3N1AB8BV2LY2	37.4 in				Sedan		Atlanta	29		41	30360	Priced below KB	I4	2000	I4	Gun Metallic	
6	1GNKRHKD0HJ	36.8 in				SUV / Crossover		Beavercreek Tov	15		48	45434	Ask how your ne	V6	3600	V6	Mosaic Black Me	FALSE
7	3CZRU6H37LM	39.3 in				SUV / Crossover		Lancaster	27		71	1523	[!@@Additional	I4	1800	I4	MODERN STE	
8	5TFUY5F11LX9	34.7 in		--	78.7 in	Pickup Truck		Langhorne	13		17	19047	Team Toyota of I	V8	5700	V8	Super White	
9	1J4FA49S33P3(	35 in				SUV / Crossover		Austin	15		4	78757	5inch Suspensio	I6	4000	I6	Red	FALSE
10	1GCRYDED1LZ	35.2 in		--	79.4 in	Pickup Truck		Big Rapids	16		9	49307	Northsky Blue M	V8	5300	V8	Northsky Blue Metallic	
11	2C4RDGEG9HF	36.5 in				Minivan		Asheville	17		33	28805	Priced below KB	V6 Flex Fuel Vel	3600	V6 Flex Fuel Vel	Billet Clear Coat	TRUE
12	JA4MW51SX5J(	35.2 in				SUV / Crossover		Cumming	13		552	30028	470-533-5400. N	V6	3800	V6		FALSE
13	1FTEW1E53KF/	43.6 in		--	67.1 in	Pickup Truck		Pennsville	16		580	8070	[!@@Additional	V8 Flex Fuel Vel	5000	V8 Flex Fuel Vel	Oxford White/Ma	FALSE
14	2HGFA1F96AH5	34.6 in				Sedan		Maryville	25		110	37801	Visit Five Star Au	I4	1800	I4	Gray	FALSE
15	5FNYF5H50MB(	38.4 in				SUV / Crossover		Jefferson City	20		15	65109	2021 Honda Pilc	V6	3500	V6	Black	
16	3GKALPEXXLL2	39.7 in				SUV / Crossover		Plymouth	22		200	48170	[!@@Additional	I4	2000	I4	Summit White	
17	JTMZFREV2ED(	37.2 in				SUV / Crossover		Philadelphia	24		5	19111	Recent Arrival! L	I4	2500	I4	Shoreline Blue P	FALSE
18	1GKKVREDXBJ	36.9 in				SUV / Crossover		Sacramento	16		377	95823	THIS VEHICLE	V6	3600	V6	White	TRUE
19	1FMCU9G61LU	40.7 in				SUV / Crossover		Ukiah	26									
20	3MVDMBEM5LM	36.3 in				SUV / Crossover		Asheville	24									
21	1G1ZB5ST4LF1	38.1 in				Sedan		Andover	29									
22	1FMCU0G62LU	40.7 in				SUV / Crossover		San Bernardino	27									
23	1GYKPCRS5LZ	39.1 in				SUV / Crossover		Las Vegas	18									
24	3GCUYDED9LC	43.4 in		--	69.9 in	Pickup Truck		Clarksburg	16									
25	2G1115SL7E92	39.8 in				Sedan		Los Angeles	21									
26	2GNAXKEV1L6	39.9 in				SUV / Crossover		Brunswick	26									
27	19XFB2F59CE0	36.2 in				Sedan		Dallas	28									
28	3GTP1NEC7HG	40.9 in		--	69.3 in	Pickup Truck		Sheboygan										
29	1FTER1EH9LLA	30.4 in		--	72.8 in	Pickup Truck		Seaford	21									
30	1VWCN7A30EC	39.1 in				Sedan		Watertown	30									
31	2B3KA43G08H3	40.2 in				Sedan		Modesto										
32	3GTP9EEL4LG3	43.4 in		--	69.9 in	Pickup Truck		Austin	15									
33	19UUB2F42KA0	34.5 in				Sedan		Gilbert	20									
34	2CCUKREC8H(	40.9 in			69.3 in	Pickup Truck		Aurora	16									

Random Forest Regression

```
%pyspark

# 4. Assemble + scale
from pyspark.ml.feature import VectorAssembler, StandardScaler

feature_columns = [
    "city_fuel_economy","highway_fuel_economy","daysonmarket",
    "engine_displacement","horsepower","mileage","seller_rating",
    "year","make_indexed","model_indexed",
    "torque","engine_cylinders","power","wheelbase","width"
]

assembler = VectorAssembler(
    inputCols=feature_columns,
    outputCol="assembled_features"
)
```

The Goal

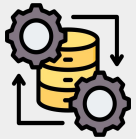
To create accurate models that can be used to predict the *price* and *accident* variables using Hadoop Distributed File System and PySpark.

Experimental Specifications

Hadoop Version	3.1.2
HDFS Configuration	390.17 gb Capacity
Total Nodes	3
Spark Version	3.0.2

Workflow Overview

01 Pre-Processing



02 Splitting the Data

03 Assembler & Pipeline

04 Model

05 Train Validation & Cross Validation

06 Evaluation

01 Pre-Processing

```
df.printSchema()

root
 |-- vin: string (nullable = true)
 |-- back_legroom: string (nullable = true)
 |-- bed: string (nullable = true)
 |-- bed_height: string (nullable = true)
 |-- bed_length: string (nullable = true)
 |-- body_type: string (nullable = true)
 |-- cabin: string (nullable = true)
 |-- city: string (nullable = true)
 |-- city_fuel_economy: double (nullable = true)
 |-- combine_fuel_economy: string (nullable = true)
 |-- daysonmarket: integer (nullable = true)
 |-- dealer_zip: string (nullable = true)
 |-- description: string (nullable = true)
 |-- engine_cylinders: string (nullable = true)
 |-- engine_displacement: double (nullable = true)
 |-- engine_type: string (nullable = true)
```

```
for name, dtype in [
    ("city_fuel_economy", "double"),
    ("highway_fuel_economy","double"),
    ("daysonmarket","int"),
    ("engine_displacement","double"),
    ("horsepower","double"),
    ("mileage","double"),
    ("seller_rating","double"),
    ("year","int"),
    ("price","double"),
    ("torque","double"),
    ("engine_cylinders","double"),
    ("power","double"),
    ("wheelbase","double"),
    ("width","double"),
]:
    data = data.withColumn(name, col(name).cast(dtype))
```

```
%pyspark
# 3. Index categorical features
from pyspark.ml.feature import StringIndexer

make_indexer  = StringIndexer(inputCol="make_name",  outputCol="make_indexed")
model_indexer = StringIndexer(inputCol="model_name", outputCol="model_indexed")

data = make_indexer.fit(data).transform(data)
data = model_indexer.fit(data).transform(data)
```

Took 3 sec. Last updated by anonymous at May 03 2025, 9:11:17 PM.

- Essential data cleaning processes, for example, converting data types, indexing categorical variables and handling missing values.

01 Feature Importance

```
%pyspark
# 9. Feature importance
importances = rf_model.featureImportances.toArray()
for feature, score in zip(feature_columns, importances):
    print(f"{feature}: {score:.4f}")
```

```
city_fuel_economy: 0.0658
highway_fuel_economy: 0.0716
daysonmarket: 0.0476
engine_displacement: 0.0757
horsepower: 0.2187
mileage: 0.1915
seller_rating: 0.0271
year: 0.1550
make_indexed: 0.0846
model_indexed: 0.0624
torque: 0.0000
engine_cylinders: 0.0000
power: 0.0000
wheelbase: 0.0000
width: 0.0000
```

Feature Importance from Random Forest

	feature	coefficient
0	city_fuel_economy	-14833.229917
4	engine_type_index	-14686.785567
7	maximum_seating_index	-11350.793752
6	fuel_type_index	-11280.454991
1	highway_fuel_economy	-9386.707043
3	body_type_index	-8176.955298
10	year_index	6893.039711
5	make_name_index	-3601.329178
8	model_name_index	-1646.974202
9	wheel_system_index	-483.822212
2	mileage	-468.488020

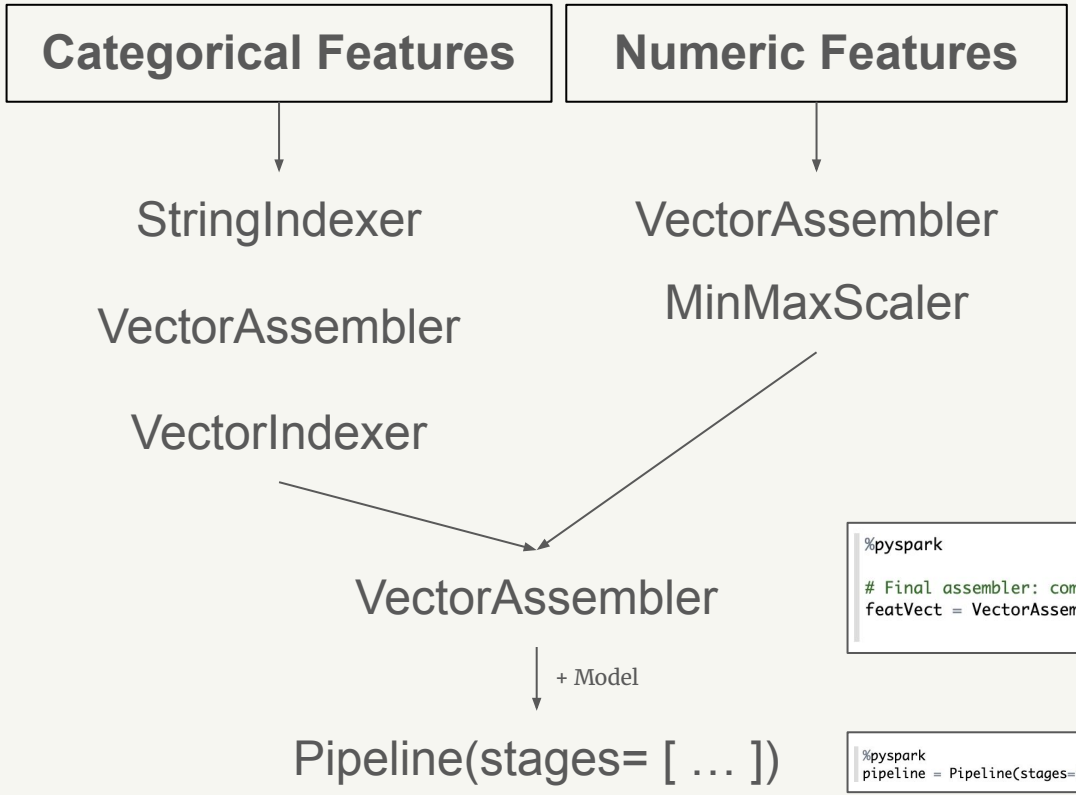
Coefficients from Linear Regression

02 Splitting the Data

```
%pyspark

splits = data.randomSplit([0.7, 0.3])
train = splits[0]
test = splits[1]
train_rows = train.count()
test_rows = test.count()
print("Training Rows:", train_rows, "\nTesting Rows:", test_rows)
```

03 Assembler & Pipeline




```pyspark
%pyspark

# Final assembler: combine scaled numeric + categorical
featVect = VectorAssembler(inputCols=["catFeatures", "scaled_features"], outputCol="features")
```

```pyspark
%pyspark
pipeline = Pipeline(stages=indexers + [encoder, categorical_VectorAssembler, numeric_assembler, scaler, featVect, lr])
```

04 Model

#1 Linear regression

```
from pyspark.ml.regression import LinearRegression
```

#2 Random forest regression

```
from pyspark.ml.regression import RandomForestRegressor
```

price

#3 Random forest classifier

```
from pyspark.ml.classification import RandomForestClassifier
```

#4 Binomial logistic regression

```
from pyspark.ml.classification import LogisticRegression
```

has_accidents

Cross-Validation for Random Forest Regression *(Price)*

```
# 6A. Cross-Validation
rf = RandomForestRegressor(featuresCol="features", labelCol="price")
paramGrid = ParamGridBuilder() \
    .addGrid(rf.numTrees, [50]) \
    .addGrid(rf.maxDepth, [5, 10]) \
    .build()

evaluator = RegressionEvaluator(labelCol="price", predictionCol="prediction", metricName="rmse")
crossval = CrossValidator(estimator=rf,
                          estimatorParamMaps=paramGrid,
                          evaluator=evaluator,
                          numFolds=3)
```

Train Validation Split for Linear Regression (*Price*)

```
%pyspark

tvs = TrainValidationSplit(estimator=pipeline,
                           estimatorParamMaps=paramGrid,
                           evaluator=RegressionEvaluator(labelCol="price", predictionCol="prediction", metricName="rmse"))

start = time.time()
tvsModel = tvs.fit(train)
end = time.time()

tvs_time

tvs_predictions = tvsModel.transform(test)

print(f"tvsModel fit time: {end - start:.2f} seconds")

# EVAL

evaluator_rmse = RegressionEvaluator(labelCol="price", predictionCol="prediction", metricName="rmse")
evaluator_r2 = RegressionEvaluator(labelCol="price", predictionCol="prediction", metricName="r2")

tvs_rmse = evaluator_rmse.evaluate(tvs_predictions)
tvs_r2 = evaluator_r2.evaluate(tvs_predictions)

print("TVS RMSE =", tvs_rmse)
print("TVS R² =", tvs_r2)
```

06 Evaluation

Results

	Algorithm	Predictor Variables	RMSE	R2	Computing Time (Training)
Regression	#1 Linear Regression	price	**CV:** 9537.86 **TVS:** 9546.95	**CV:** 0.7755 **TVS:** 0.7751	**CV:** 731.42 seconds **TVS:** 348.63 seconds
Regression	#2 Random Forest Regression	price	**CV:** 5978.83 **TVS:** 5978.84	**CV:** 0.8824 **TVS:** 0.8824	**CV:** 1131.87 seconds **TVS:** 763.25 seconds
		Predictor Variables	**Accuracy & AUC**		**Computing Time (Training)**
Classification	#3 Random Forest Classifier	has_accidents	**AUC:** 0.6600 **Accuracy:** 0.8842 **Precision:** 0.7818 **Recall:** 0.8842 **F1:** 0.8299		**CV:** 1084.35 seconds **TVS:** 630.73 seconds
Classification	#4 Binomial Logistic Regression	has_accidents	**AUC:** 0.6830 **Accuracy:** 0.8829 **Precision:** 0.8186 **Recall:** 0.8829 **F1:** 0.8281		**CV:** 400.45 seconds **TVS:** 251.54 seconds

Thank you!

https://github.com/Mgratricz/CIS-5560-Used-Cars-Prediction-Analysis

Github for Code Files

https://github.com/Mgratricz/CIS-5560-Used-Cars-Prediction-Analysis

```
%pyspark

# 4. Assemble + scale
from pyspark.ml.feature import VectorAssembler, StandardScaler

feature_columns = [
    "city_fuel_economy","highway_fuel_economy","daysonmarket",
    "engine_displacement","horsepower","mileage","seller_rating",
    "year","make_indexed","model_indexed",
    "torque","engine_cylinders","power","wheelbase","width"
]

assembler = VectorAssembler(
    inputCols=feature_columns,
    outputCol="assembled_features"
)
assembled_data = assembler.transform(data)

scaler = StandardScaler(
    inputCol="assembled_features",
    outputCol="features",
    withMean=True,
    withStd=True
)
scaler_model = scaler.fit(assembled_data)
final_data   = scaler_model.transform(assembled_data)


final_data.select("features").show(3, truncate=False)
```

```
+--------------------------------------------------------------------
|features
+--------------------------------------------------------------------
|[-0.5668006073478629,-0.4167300982404232,0.46520517013174334,0.387011
|[0.9316074980846417,1.1481263077387083,-0.29629995817986815,-0.849857
|[0.6319258769981408,0.2092124641512294,-0.05722276673319944,-0.849857
+--------------------------------------------------------------------
only showing top 3 rows
```

- Random Forest Regression (Price)
- Selecting the feature columns plus feature handling

Linear Regression (Regression)

```
25/05/05 00:49:37 INFO DAGScheduler: Job 971 is finished. Cancelling potential speculative or zombie tasks for this job
25/05/05 00:49:37 INFO YarnScheduler: Killing all running tasks in stage 1717: Stage finished
25/05/05 00:49:37 INFO DAGScheduler: Job 971 finished: treeAggregate at Statistics.scala:58, took 10.451307 s
CV RMSE = 9168.17633583364
CV R² = 0.7455060942866951
Best regParam: 0.1
Best elasticNetParam: 0.0
                 feature   coefficient
4        engine_type_index -16406.579791
2                  mileage -14508.299505
0        city_fuel_economy -13483.253628
6          fuel_type_index -11841.023882
7    maximum_seating_index  -9217.573999
1     highway_fuel_economy  -8859.655222
3          body_type_index  -7605.853971
10              year_index   3686.144165
8         model_name_index  -3469.757515
9       wheel_system_index  -1966.334445
5          make_name_index  -1140.343558

--- Summary of All Results ---
Training Rows: 525,123
Testing Rows: 224,887

--- Summary of Cross-Validation Results ---
Cross-Validated RMSE: 9168.1763
Cross-Validated R2:   0.7455
CV Time: 796.74 seconds

--- Summary of Train-Validation-Split Results ---
TrainValidationSplit RMSE: 9180.9726
TrainValidationSplit R2:   0.7448
TVS Time: 359.42 seconds
25/05/05 00:49:37 INFO SparkContext: Invoking stop() from shutdown hook
25/05/05 00:49:37 INFO AbstractConnector: Stopped Spark@611c7b5c{HTTP/1.1, (http/1.1)}{0.0.0.0:4042}
25/05/05 00:49:37 INFO SparkUI: Stopped Spark web UI at http://bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com:4042
25/05/05 00:49:37 INFO YarnClientSchedulerBackend: Interrupting monitor thread
```

Logistic Regression (Classification)

```
25/05/05 01:24:24 INFO DAGScheduler: Job 679 finished: collectAsMap at MulticlassMetrics.scala:61, took 12.460391 s
Train Validation Split AUC (Logistic Regression): 0.6830
Train Validation Split Accuracy (Logistic Regression): 0.8838
Train Validation Split Precision (Logistic Regression): 0.8336
Train Validation Split Recall (Logistic Regression): 0.8838
Train Validation Split F1 (Logistic Regression): 0.8301

--- Summary of All Results ---
Training Rows: 74,460
Testing Rows: 32,197

--- Summary of Cross-Validation Results ---
CV AUC: 0.6830
CV Accuracy: 0.8838
CV Precision: 0.8336
CV Recall: 0.8838
CV F1: 0.8301
CV Time: 400.45 seconds

--- Summary of Train-Validation-Split Results ---
TVS AUC: 0.6830
TVS Accuracy: 0.8838
TVS Precision: 0.8336
TVS Recall: 0.8838
TVS F1: 0.8301
TVS Time: 251.54 seconds
25/05/05 01:24:24 INFO SparkContext: Invoking stop() from shutdown hook
25/05/05 01:24:24 INFO AbstractConnector: Stopped Spark@7241c517{HTTP/1.1, (http/1.1)}{0.0.0.0:4042}
25/05/05 01:24:24 INFO SparkUI: Stopped Spark web UI at http://bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com:4042
25/05/05 01:24:24 INFO YarnClientSchedulerBackend: Interrupting monitor thread
25/05/05 01:24:24 INFO YarnClientSchedulerBackend: Shutting down all executors
25/05/05 01:24:24 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
25/05/05 01:24:24 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
25/05/05 01:24:24 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/05/05 01:24:24 INFO MemoryStore: MemoryStore cleared
25/05/05 01:24:24 INFO BlockManager: BlockManager stopped
25/05/05 01:24:24 INFO BlockManagerMaster: BlockManagerMaster stopped
```

Random Forest (Regression)

```
25/05/05 00:28:28 INFO DAGScheduler: Job 149 finished: treeAggregate at Statistics.scala:58, took 4.129032 s
Final RMSE: 6040.05
Final R2:   0.8800
city_fuel_economy: 0.0423
highway_fuel_economy: 0.0490
daysonmarket: 0.0384
engine_displacement: 0.0932
horsepower: 0.2167
mileage: 0.2171
seller_rating: 0.0406
year: 0.1681
make_indexed: 0.0792
model_indexed: 0.0555
torque: 0.0000
engine_cylinders: 0.0000
power: 0.0000
wheelbase: 0.0000
width: 0.0000
Cross-Validated RMSE: 5978.83
Cross-Validated R2:   0.8824
CV Time: 139.80 seconds
TrainValidationSplit RMSE: 5978.87
TrainValidationSplit R2:   0.8824
TVS Time: 61.21 seconds
Final RMSE: 6040.05
Final R2:   0.8800
25/05/05 00:28:28 INFO SparkContext: Invoking stop() from shutdown hook
25/05/05 00:28:28 INFO AbstractConnector: Stopped Spark@45c0f39d{HTTP/1.1, (http/1.1)}{0.0.0.0:4041}
25/05/05 00:28:28 INFO SparkUI: Stopped Spark web UI at http://bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com:4041
25/05/05 00:28:28 INFO YarnClientSchedulerBackend: Interrupting monitor thread
25/05/05 00:28:28 INFO YarnClientSchedulerBackend: Shutting down all executors
25/05/05 00:28:28 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
25/05/05 00:28:28 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
25/05/05 00:28:28 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
25/05/05 00:28:29 INFO MemoryStore: MemoryStore cleared
25/05/05 00:28:29 INFO BlockManager: BlockManager stopped
25/05/05 00:28:29 INFO BlockManagerMaster: BlockManagerMaster stopped
25/05/05 00:28:29 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
25/05/05 00:28:29 INFO SparkContext: Successfully stopped SparkContext
25/05/05 00:28:29 INFO ShutdownHookManager: Shutdown hook called
```

Random Forest (Classification)

```
5/05 01:09:51 INFO BlockManagerInfo: Added broadcast_1117_piece0 in memory on bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:29101 (size: 2.2 KiB, free: 365.4 MiB)
5/05 01:09:51 INFO MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 420 to 10.1.0.88:24762
5/05 01:09:51 INFO TaskSetManager: Finished task 0.0 in stage 981.0 (TID 1154) in 12 ms on bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com (executor 2) (1/1)
5/05 01:09:51 INFO YarnScheduler: Removed TaskSet 981.0, whose tasks have all completed, from pool
5/05 01:09:51 INFO DAGScheduler: ResultStage 981 (collectAsMap at MulticlassMetrics.scala:61) finished in 0.014 s
5/05 01:09:51 INFO DAGScheduler: Job 484 is finished. Cancelling potential speculative or zombie tasks for this job
5/05 01:09:51 INFO YarnScheduler: Killing all running tasks in stage 981: Stage finished
5/05 01:09:51 INFO DAGScheduler: Job 484 finished: collectAsMap at MulticlassMetrics.scala:61, took 16.606855 s
n Validation Split AUC (RandomForest Regression): 0.6600
n Validation Split Accuracy (RandomForest Regression): 0.8842
n Validation Split Precision (RandomForest Regression): 0.7818
n Validation Split Recall (RandomForest Regression): 0.8842
n Validation Split F1 (RandomForest Regression): 0.8299

Summary of All Results ---
ning Rows: 74,832
ing Rows: 31,825

Summary of Cross-Validation Results ---
UC: 0.6581
ccuracy: 0.8842
recision: 0.7818
ecall: 0.8842
1: 0.8299
ime: 1084.35 seconds

Summary of Train-Validation-Split Results ---
AUC: 0.6600
Accuracy: 0.8842
Precision: 0.7818
Recall: 0.8842
F1: 0.8299
Time: 630.73 seconds
10 Feature Importances (Readable Names):
: 0.0940
age: 0.0621
ue: 0.0209
r: 0.0201
h: 0.0175
_type = Pickup Truck: 0.0173
epower: 0.0139
ne_cylinders: 0.0130
lbase: 0.0127
l_name = 200: 0.0109
er_zip = 33483: 0.0100
er_zip = 49801: 0.0100
ne_displacement: 0.0085
rior_color = Jet Black/Medium Titanium: 0.0069
t_legroom: 0.0069
rior_color = jet black titanium: 0.0066
l_name = Pilot: 0.0060
er_zip = 35235: 0.0057
er_zip = 57401: 0.0055
l_name = ES 350: 0.0055
5/05 01:09:52 INFO SparkContext: Invoking stop() from shutdown hook
5/05 01:09:52 INFO AbstractConnector: Stopped Spark@1e1641cc{HTTP/1.1, (http/1.1)}{0.0.0.0:4040}
5/05 01:09:52 INFO SparkUI: Stopped Spark web UI at http://bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com:4040
5/05 01:09:52 INFO YarnClientSchedulerBackend: Interrupting monitor thread
5/05 01:09:52 INFO YarnClientSchedulerBackend: Shutting down all executors
5/05 01:09:52 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
5/05 01:09:52 INFO YarnClientSchedulerBackend: YARN client scheduler backend Stopped
5/05 01:09:52 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```