Deployment Guide for Housing Prediction Model

DATABRICKS IMPLEMENTATION

Sanika Katekar | Furniture.com Assignment | 28 Nov 2023

CONTENTS

EXECUTIVE SUMMARY

- **Overview:** Company X, a leading force in Real Estate, is embarking on a transformative journey with a cutting-edge Housing Prediction Model.
- **Key Focus Areas:** This executive summary highlights crucial facets of the deployment strategy, covering the deployment pipeline, infrastructure, data integration, model development, and monitoring.
- **Key Components:** The deployment plan integrates robust practices in model management, deployment, and security to ensure a seamless and trustworthy housing prediction process.
- **Outcome:** Through this strategic approach, Company X aims to instill confidence in users by delivering accurate and reliable housing predictions.

INTRODUCTION

Model Overview

Explored and implemented regression models like Linear Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM).

Best Models and Hyperparameter Tuning using GridSearchCV:

1. Linear Regression (No hyperparameters to tune).
2. DecisionTreeRegressor:
    - Max Depth: 5
    - Min Samples Split: 5
    - Min Samples Leaf: 1
3. RandomForestRegressor:
    - Max Depth: 30
    - Number of Estimators: 80
4. Support Vector Machine (SVM):
    - C: 1000
    - Gamma: 0.01

Model Evaluation on Validation Set:

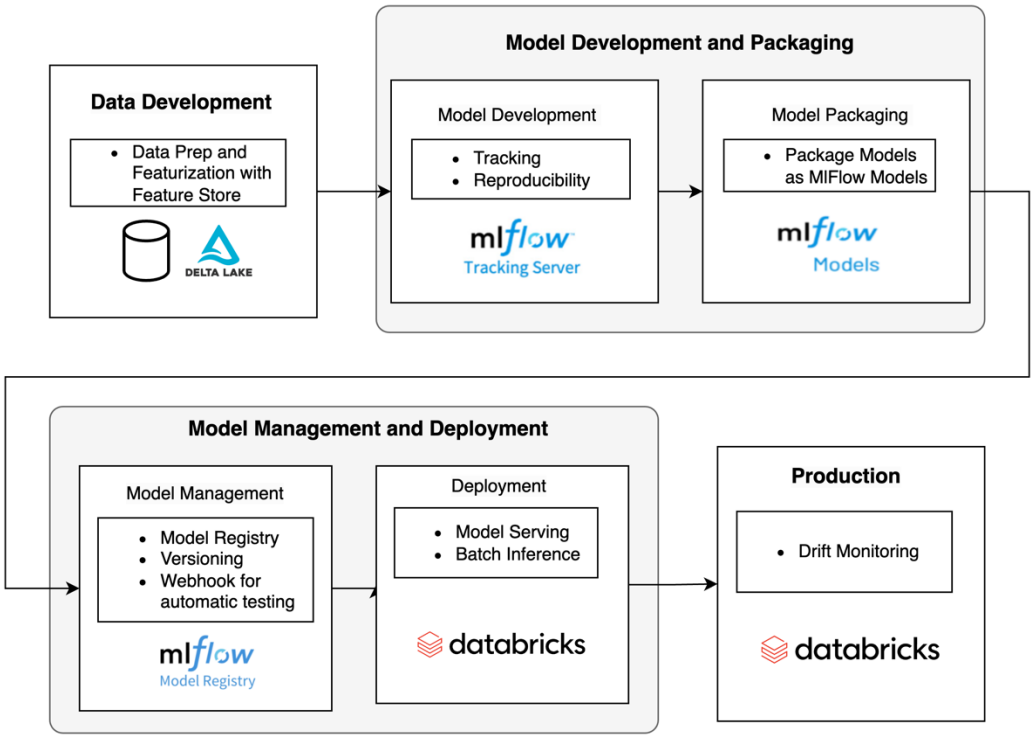
- Mean Squared Errors:
    1. Linear Regression: 0.0281
    2. Decision Tree: 0.0414

3. Random Forest: 0.0316
4. SVM: 0.0169

Final Model Selection:

- SVM was chosen as the final model for deployment based on its superior performance in Mean Squared Error on the validation set.

Deployment Pipeline



INFRASTRUCTURE AND DATA

Infrastructure Requirements

- **Databricks Workspace:** Create a dedicated workspace named "**Price_Discovery_Model**" with proper access controls for data scientists, engineers, and administrators.
- **Clusters Configuration:** Implement an optimized cluster configuration with considerations for the model's resource requirements. Enable autoscaling to handle varying workloads efficiently.

Data Integration and Storage

In the data preprocessing phase, columns were standardized, and irrelevant ones were dropped. Numeric columns were converted, and missing values were imputed using mean or mode. String columns underwent whitespace stripping and lowercase conversion. High-cardinality categorical columns were target-encoded, and binary categorical columns were one-hot encoded. The dataset was split into training, validation, and test sets, with 80%, 10%, and 10% of the data, respectively, and saved in CSV format. To handle outliers, a log transformation was applied to the entire dataset. Delta Lake and Feature store and be utilized to store and version the features created. This will ensure consistency between the features used for training and deployment.

MODEL DEVELOPMENT AND PACKING

Mlflow Tracking

Leverage MLflow for comprehensive experiment tracking during model development to enhance transparency and accountability.

- **Implementation Steps:** Log key information such as hyperparameters, metrics, and artifacts using Mlflow API commands during the training process. Leverage MLflow's capabilities to compare different model versions based on training runs.

MLflow Packaged Model

Package the trained model using MLflow creating an MLFlow model, encapsulating both the model and the associated pre- and post-processing steps.

- **Implementation Steps:** Implement a custom Python class extending mlflow.pyfunc.PythonModel to encapsulate the entire pipeline, handling loading, preprocessing, inference, and post-processing.

MODEL MANAGEMENT AND DEPLOYMENT

Model Registry and Webhooks

Register the packaged models in MLflow Model Registry to facilitate version control and organization.

- **Staging:** Register with mlflow.register_model and designate as "Staging."
- **Production:** Promote from staging to production via MLflow UI or API.
- **Archived:** Archive a model version when no longer deployed or used.

Configure webhooks in MLflow to trigger automated testing events.

- Develop a Databricks job to assess models in the registry.
- Automate through MLflow Webhook.
- When a new version enters:
    - Incorporate the model version.
    - Validate input/output schema.
    - Execute sample code.

Deployment And Serving

Deploy the packaged model using MLflow's deployment options. Configure score serving batch inference in the case of house prediction model.

Implementation Steps:

- On the MLflow Run page for your model, extract the code snippet generated for making predictions on pandas or Apache Spark DataFrames.
- For executing batch predictions systematically, develop a notebook incorporating the prediction code. Subsequently, initiate the notebook as a Databricks job, with options for immediate execution or scheduling based on our requirements.

MONITORING AND SECURITY

Model Production - Monitoring Model Drifts

Leverage MLflow for tracking and comparing model performance metrics over time. Implement specialized tools or statistical methods for advanced model drift detection.

- **Check Accuracy and Data Drift:** Ensure the model's accuracy and detect any data and concept drift. Implement statistical methods, such as Kolmogorov-Smirnov tests or Jensen-Shannon divergence, to quantify the similarity between feature distributions.
- Use Databricks jobs to compute statistical metrics on feature distributions. Log and track these metrics using MLflow to detect significant changes indicative of data drift.
- **Publish Metrics:** Store computed metrics in Lakehouse tables for analysis. Set up notifications if a metric surpasses a defined threshold.

Security And Compliance

Authentication, Authorization and Data Privacy Compliance:

- Implement proper authorization policies and robust authentication mechanisms for model access.
- Ensure the model adheres to data privacy regulations (e.g., GDPR) by design.

CONCLUSION

Company X is poised for a transformative leap in Real Estate with its advanced Housing Prediction Model. The deployment plan strategically covers key areas, from the deployment pipeline and infrastructure to data integration, model development, and monitoring. The deployment pipeline, guided by Databricks Workspace and Delta Lake, establishes a resilient infrastructure with optimized cluster configurations and version-controlled data management. Model development incorporates MLflow for comprehensive tracking and packaging, ensuring seamless deployment with webhooks automating testing events.